



Robot Schedule  
13  
OPAL Reference  
Guide

## **Copyright Terms and Conditions**

---

Copyright © Fortra, LLC and its group of companies. All trademarks and registered trademarks are the property of their respective owners.

The content in this document is protected by the Copyright Laws of the United States of America and other countries worldwide. The unauthorized use and/or duplication of this material without express and written permission from Fortra is strictly prohibited. Excerpts and links may be used, provided that full and clear credit is given to Fortra with appropriate and specific direction to the original content.

202312271144

**■ Introduction**

Introduction ..... 7  
 OPAL Objects ..... 7  
 Data Area Objects ..... 8

**■ Robot OPAL Menu**

Accessing the Robot OPAL Object Menu ..... 9  
 Maintain OPAL Objects ..... 10  
   OPAL Object Panel ..... 12  
   Defining OPAL Code ..... 14  
   Prompting the Operation Values Field ..... 16  
   Creating the OPAL Program ..... 18  
   Finding Where an OPAL Object is Used ..... 19  
   Using the OPAL Simulator ..... 20  
   Displaying Trace Data ..... 23  
   Printing Your OPAL Code and Scan/Replace OPAL Values ..... 24  
 Display OPAL Source Code—RBTOPALSRC Command ..... 25  
 Maintain OPAL Tables ..... 26  
   OPAL Table Panel ..... 27  
   OPAL Table Elements Panel ..... 28  
 Maintain Data Area Definitions ..... 29  
   Data Area Definition Panel ..... 30  
   Data Area Definition Elements ..... 31

**■ OPAL Basics**

OPAL Statements ..... 32  
 Logic Operands ..... 33  
   IF—Perform If Conditions are Met ..... 34  
   Multiple Conditions ..... 34  
   AND—Add a Condition to a Condition Set ..... 36  
   OR—Start New Condition Set ..... 37  
   ELSE—If Conditions Not Met ..... 38  
   END—Ends Operation List ..... 39

**■ OPAL Operations**

Operation Statements Tell Robot Schedule What to Do ..... 40  
   EQ or NE: Equal or Not Equal ..... 41  
   GT, GE, LT, LE: Order Comparisons ..... 42  
   CT, DC: Contains or Doesn't Contain ..... 42  
   INTABLE, NOTINTABLE: Value Found or Not Found in a Table ..... 43  
 Operation Values Field ..... 44  
 OPAL Constants ..... 44  
 OPAL Variables ..... 44  
 Job Operations ..... 45  
   SKIP — Skip Job ..... 45  
   SKIPSTATUS — Will Job Be Skipped? ..... 46  
   ADDMIN — Add Minutes ..... 47  
 RTVMQDEPTH — Retrieve WebSphere MQ Message Queue Depths ..... 48

## ■ Check Current Status

## ■ Standard Operations

Reading or Changing Job Setup .....	51
RTVJOB — Retrieve Job Record .....	51
RTVJOB Example .....	52
UPDJOB — Update Job .....	53
Prerequisite Job Variables .....	54
THISJOB — The Current Job .....	55
RBTJOBNBR — Robot Schedule Job Number .....	55
RBTJOBNAME — Robot Schedule Job Name .....	55
Robot Schedule Job Variables Table .....	56
Reading or Changing Job Environment .....	57
RTVENV — Retrieve Environment .....	57
UPDENV — Update Environment .....	58
Environment Variables Table .....	59
Reading or Changing Data Areas .....	60
RTVDTADFN — Retrieve Data Area Definition .....	60
CHGDTADFN — Change Data Area Definition .....	60
Check Current Status .....	61
Check Object Status .....	61
RTVOBJSTS — Retrieve Object Status .....	61
OBJSTS — Object Status .....	62
Check Device Status .....	63
ACTCTL — Active Controller .....	63
ACTDEV — Active Device .....	63
ACTLIN — Active Line .....	64
ACTRCL — Active Robot Client .....	64
DEVSTS — Device Status .....	65
Check Work Management Status .....	66
ACTJOB — Active Job .....	66
ACTSBS — Active Subsystem .....	66
ACTUSR — Active User .....	67
RTVJOBQ — Retrieve Job Queue Status .....	67
JOBQCNT — Number of Jobs in Job Queue .....	68
JOBQHOLD — Is Job Queue on Hold? .....	69
Changing User Variable Values .....	70
CHGTO — Change the Value of a Variable .....	70
ADD — Add to a Numeric Value of a Variable .....	70
SUB — Subtract from a Numeric User Variable .....	71
Concatenate Information .....	72
CAT — Concatenate Values .....	72
BCAT — Insert a Blank and Concatenate .....	72
Send Messages .....	73
SNDMSG — Send an iSeries Message .....	73
SNDBRKMSG — Send an iSeries Break Message .....	73
SENDMC — Send a Message to a Message Center .....	74
PAGE — Send a Pager Message .....	75
QUIT — Stop All Processing .....	76

■ Variables

Changing the Next Statement Processed ..... 77  
 TAG — Tags a Program Location ..... 77  
 GOTO — Go to a Tagged Location ..... 77  
 \* — Adds a Comment ..... 77  
 SNDRBTDTA — Run a Reactive Job ..... 78

Variables ..... 79

Use the Robot Schedule Calendar ..... 81  
 WORKDAY—Is it a Working Day? ..... 81  
 DAY—Day of the Week ..... 81  
 DAYMTH—Day Number in the Month ..... 81  
 LASTDAY — Last Date in the Month ..... 82  
 WEEKNO — Week Number ..... 82

Assign Values to User Variables ..... 83  
 USRFLDn — Character Variables ..... 83  
 USRLONGn — Longer Character Variables ..... 84  
 USRFLGn — Flag Variables ..... 84  
 USRNBRn — Floating Point Variables ..... 85  
 USRCNTn — Numeric Variables ..... 85

Use System Information ..... 86  
 JOBNAME — Job Name ..... 86  
 JOBNBR — Job Number ..... 86  
 JOBUSER — Job User ..... 86  
 SBMTIME — Time Job was Submitted ..... 87  
 SYSTEM — System Name ..... 87  
 SYSTIME — System Time ..... 88  
 SYSDATE — System Date ..... 88

■ OPAL Examples

OPAL Examples ..... 89  
 Page Operator If Job Will Be Skipped ..... 89  
 Delay Job If a Certain User Is Active ..... 90  
 Prevent Job From Running Before a Certain Time ..... 90  
 Change Job Setup ..... 90  
 Change Job Environment ..... 91  
 Change Page Based on Time of Day ..... 91  
 Check Current Conditions ..... 91  
 Change Values in a Data Area ..... 92  
 Is Mainframe File Ready for Job? ..... 92  
 Delay Job If a Certain Job is Active ..... 92  
 Skip Job If IT Staff is Still Signed On ..... 93  
 Look for Less Busy Job Queue ..... 94  
 Run Restricted State Utilities Unless QSYSOPR is Active ..... 94  
 Choose a Job Queue Based on Conditions, Check for Active Devices ..... 95  
 Use Only Authorized Printers ..... 96  
 Use a Different Job Queue When Conditions are Right ..... 96  
 Skip or Delay Job Based on Completion Code ..... 97

- **Index**
- **OPAL Quick Reference**

Change Time Range ..... 97  
Restart Job If Not Running ..... 98  
Restart Subsystem If Not Running ..... 98  
Page Operator If Backups Not Started on Time ..... 99  
Check for Virtual Day ..... 99

Index ..... 100

Robot Schedule OPAL Quick Reference ..... 105

## ■ Introduction

Operator Assistance Language (OPAL), our powerful fourth-generation operations language (4GL) is now part of Robot Schedule by popular demand. The OPAL language lets you specify operations to be performed when certain conditions are met. The OPAL language is easy to learn. It is a fixed format language, like RPG, but its syntax is more like CL. To code OPAL statements you just fill in fields on a panel. Help is available for each field.

You enter an OPAL statement by filling one or more fields on a line. The statement fields are as follows:

<b>Logic Operand</b>	Used to set up powerful data conditions with IF, AND, and ELSE logic.
<b>Variable</b>	Specifies the variable to be tested or changed.
<b>Operation</b>	Specifies the operation to be performed or the comparison to be made.
<b>Operation Values</b>	Provides the value to be used by the comparison or operation.

Read through the first chapter of this Guide to understand the user interface for Robot Schedule OPAL. Refer to later chapters more detailed information about the operands, operations, variables, and operation values.

## ■ OPAL Objects

In Robot Schedule, you will generally use OPAL for six types of tasks:

- Checking for conditions that can cause a failure and fixing them
- Skipping or delaying a job
- Reading or changing a job setup
- Reading or changing a job environment
- Reading or changing a data area
- Checking the current status of objects or resources

OPAL lets you set up conditions so that it can make a decision about running a job at the actual run time—just like an operator would. OPAL can skip or delay a job if conditions are not right.

But OPAL can do much more. It can:

- Change any Robot Schedule job setup to reflect changed conditions.
- Change any job environment object to reflect changed conditions. For example, if one printer is down you can direct all the night's reports to another printer automatically.
- Change any data area including the LDA. (For more information, see the data area objects section later.)
- Check if an object exists on the IBM i, whether the object has any locks, and even if a file has any records in it.
- Check if printers, workstations, controllers, and communication lines are up and running.
- Check if jobs, subsystems, or users are active.
- Check if job queues are too busy or on hold.
- Read or update Robot Schedule Reserved Variables. This capability gives you two-way communication with other Robot Automated Operations Solution products. Using Robot Client, it also allows for two-way communications with PCs.
- Page an expert using Robot Alert.
- Start another Robot Schedule job—even if it is on a remote IBM i.
- Plus, do many other exciting things too numerous to mention.

All you have to do is come up with a name for the object and enter the language. Then your instructions can be used over and over again by other jobs—simply go to the Exception Scheduling panel for a job and select the OPAL object you want to use.

Now you can create your own very advanced scheduling options, job prerequisites, and event monitoring options. You can have exactly the options you want. You even have the option of running your OPAL code before or after the job is submitted.

Using a fill-in-the-blank panel, you can define user fields to segment any data area or LDA. These data area definitions can be assigned a name and be used over and over again. OPAL uses the data area objects to read and update data areas. This feature greatly enhances interprogram and interjob communications.

### ■ Data Area Objects



## ***Accessing the Robot OPAL Object Menu***

OPAL (OPerator Assistance Language) allows you to specify operations to be performed when certain conditions are met. For example, you can check for conditions that caused a failure and fix them; change a job setup, job environment, or data area; skip or delay a job, and much, much more. Select **option 2** on the Robot Main Menu to access the Scheduling Objects Menu. Then select **option 6** to access the Robot OPAL Object Menu.

```
RBT1010      Scheduling Objects Menu      13:37:15
SCHOBJ

Select one of the following:

Time-Based Objects
  1. Calendar Objects
  2. Date Objects

ROBOT Job Definition Objects
  3. Library List Objects
  4. Job Environment Objects
  5. Reserved Command Variable Obj
  6. ROBOT OPAL Object Menu

Query Objects
  7. Maintain ROBOT Queries
  8. Update ROBOT Jobs via Query

Selection: 6_

F3=Exit      F21=Command Line
```

Enter a **6** on the Scheduling Objects Menu to display the Robot OPAL Object Menu.

```
RBT1010      ROBOT OPAL Object Menu      12:59:17
OPALOBJ

Select one of the following:

  1. Maintain OPAL Objects

  2. Maintain OPAL Tables

  3. Maintain Data Area Definitions

Selection: __

F3=Exit      F21=Command Line
```

# Robot Schedule OPAL Menu

## Maintain OPAL Objects

The Maintain OPAL Objects panel lists the OPAL objects you have defined.

```
RBT1010          ROBOT OPAL Object Menu          12:59:17
OPALOBJ

Select one of the following:

1. Maintain OPAL Objects

2. Maintain OPAL Tables

3. Maintain Data Area Definitions

Selection: 1

F3=Exit      F21=Command Line
```

Enter **option 1** on the Robot OPAL Object Menu to display the Maintain OPAL Objects panel.

```
RBT225          Maintain OPAL Objects          13:31:06

Start list at Object: _____

Options
A=Add/Copy/Delete      I=Select      ?=More Options
OPAL

Opt  Objects      Description      OPAL Status  Run Status
---  ---
___  ACTPRINTER    Check for active printer
___  ACTUSER        Check for any active users before r  Changed
___  CHANGEENV      Change job environment
___  CHECKSYS       Check for status prior to SAUE
___  CHKACCTUSR     Check for Accounting User signed on
___  COUNTER        Count number of times this job has  New
___  DAN            Send a msg to Dan's msgq
___  DELAYJOB       Delay a job until after 11:30 pm
___  ENDOFMONTH     End of Month cycle                    *INACTIVE
___  GROUPMBRS     Check for users on system
___  LASTMONDAY     Skip job if dayno is 28, 29, 30, 31
___  PAGEONFRI     Page on a Friday job                  New
___  PAGEONMON     Page if this job runs on a Monday

More...

F3=Exit      F4=Prompt      F6=Add OPAL      F21=Command Line
```

**Maintain OPAL Objects**

Enter a **1** in the **Opt** field to see the OPAL Object panel. Press **function key 4** to display a pop-up menu of additional options.

The status field displays status information about your OPAL code.

**Blank** The object is ready to run.

**New** You've named and described the OPAL Object on the OPAL Object panel, but have not yet created any code.

**Changed** You've displayed the OPERator Assistance Language panel, but you have not yet created a new object.

**Generate** You've created a new object on the OPERator Assistance Language panel.

**Compile** The object is compiling.

**Abnormal** There is a serious problem with your OPAL; call Technical Support.

```

RBT225                               Maintain OPAL Objects                               13:31:06

Start list at Object: _____

Options
A=Add/Copy/Delete   1=Select   ?=More Options
OPAL

Opt  Objects      Description      OPAL Status  Run Status
___  ACTPRINTER   Check for active printer
1_   ACTUSER       Check for any active users before r   Changed
___  CHANGEENVU    Change job environment
___  CHECKSYS      Check for status prior to SAVE
___  CHKACCTUSR    Check for Accounting User signed on
___  COUNTER       Count number of times this job has   New
___  DAN           Send a msg to Dan's msgq
___  DELAYJOB      Delay a job until after 11:30 pm
___  ENDOFMONTH    End of Month cycle
___  GROUPEMBRS   Check for users on system
___  LASTMONDAY    Skip job if dayno is 28, 29, 30, 31
___  PAGEONFRI    Page on a Friday job
___  PAGEONMON     Page if this job runs on a Monday
                                         *INACTIVE
                                         More...

F3=Exit   F4=Prompt   F6=Add OPAL   F21=Command Line
    
```

The name and description of the OPAL object.

The Run Status field displays run status information about your OPAL code:

**Blank** Normal status.

**\*INACTIVE** The OPAL object is inactive. Any job to which it is attached ignores it.

## OPAL Object Panel

The OPAL Object panel allows you to name and describe your OPAL object and to select several other characteristics. You must complete this panel before you can create OPAL code.

```
RBT701      Options      13:43:36
1=Select Option
- > 1=Select OPAL Object
- > 2=Maintain OPAL Detail
- 3=OPAL Where Used
- 5=OPAL Simulator
- 6=Print OPAL
- 7=Scan/Replace OPAL

Bottom

F3=Exit  F11=More Info
```

Select **option 1** to display the OPAL Object panel.

```
RBT226      OPAL Object  13:31:45

OPAL Object Name . . . . : ACTUSER
Description . . . . . : Check for any active users before running_job
Logging Level . . . . . : 0 (0=Put messages in job log,
                          1=Build trace data,
                          2=Always send status message,
                          3=Send status message on skips)
When should OPAL run . . : 1 (1=Before Submission, 2=After Submission)
Make OPAL Inactive . . . : N (Y=Yes, N=No)

F3=Exit  F12=Previous  F21=Command Line
```

Enter a logical name for the OPAL object. The name can be up to 10 characters in length

Enter a description of the object.

**OPAL Object Panel**

Select a Logging Level for the object:

- 0** Sends two messages to the job log that indicate when the OPAL object starts and its effect on the job. For example, the job will run as scheduled, the job was skipped, or the job was delayed. This is the default value.
- 1** Create a trace that shows which OPAL statements executed the last time this OPAL object ran.
- 2** Send a status message to Robot Schedule's Job Completion History that indicates the effect of the OPAL object. For example, the job will run as scheduled, the job was skipped, or the job was delayed.
- 3** Sends a status message to Robot Schedule's Job Completion History **only** when job is skipped. After you have your OPAL set up and working well, we recommend you use this logging level.

```

RBT226                                OPAL Object                                13:31:45

OPAL Object Name . . . . . : ACTUSER
Description . . . . . : Check for any active users before running job
Logging Level . . . . . : 0 (0=Put messages in job log,
                          1=Build trace data,
                          2=Always send status message,
                          3=Send status message on skips)
When should OPAL run . . . : 1 (1=Before Submission, 2=After Submission)
Make OPAL Inactive . . . . : N (Y=Yes, N=No)

F3=Exit      F12=Previous    F21=Command Line
    
```

Specify when this OPAL object should run:

- 1** The OPAL runs before the job is submitted
- 2** The OPAL runs after the job is submitted

**Considerations:** Most of the time OPAL runs before the job is submitted. This avoids tying up system resources by submitting jobs that OPAL decides not to run. However, if you have certain mission-critical jobs that must have access to specific resources at the time they run, choose to run your OPAL after the job is submitted.

Specify whether you want to make the OPAL object inactive. An inactive OPAL object (**Y**) has the same effect as removing the OPAL object name from the job record of *all* the jobs that use this object. **N** (No) is the normal setting.

## Defining OPAL Code

After you've defined your OPAL object, you're ready to enter the OPAL code. From the Maintain OPAL Objects panel, select **option 2** from the options menu to display the OPERator Assistance Language panel.

OPAL statements let you specify operations to be performed when certain conditions are met. You enter OPAL statements by filling in one or more fields on a line, as shown below. You can find more detail about the logic operands, variables, operations, and operation values later in this Guide.

```

RBT781      Options      13:43:36

1=Select Option
_ > 1=Select OPAL Object
_ > 2=Maintain OPAL Detail
_ 3=OPAL Where Used
_ 5=OPAL Simulator
_ 6=Print OPAL
_ 7=Scan/Replace OPAL

Bottom

F3=Exit  F11=More Info
    
```

Select **option 2** for an OPAL object on the Maintain OPAL Objects panel to display the OPERator Assistance Language panel.

Press **function key 7** to display the Reserved Command Variables window. You can select a variable for the field.

```

RBT245      Operator Assistance Language      13:32:17

OPAL Object Name . . . : ACTUSER      Check for any active users before runnin
Start at Sequence . . . : ___

Logic
Operand  Variable      Operation      Operation Values      Seg
IF      ACTUSR      EQ      QSYSOPR      10
        USRLONG3     CHGTO     Someone is on the system!      20
        USRLONG3     BCAT      Job will wait 15 more minutes!      30
        SNOMSG      MSG(USRLONG3) TOUSR(JEFF)      40
        ADMIN      15      50
END      60
        QUIT      70
        *      If this user is on the system, send      80
        *      into a pending state for 15 minutes      90
        100
        110
        120
More...

F3=Exit      F4=Prompt      F7=ROBOT Variables
F12=Previous  F18=Resequence  F19=Logic Check
    
```

To resequence lines of code, change the sequence numbers and press Enter. Press **function key 18** to renumber lines by equal intervals.

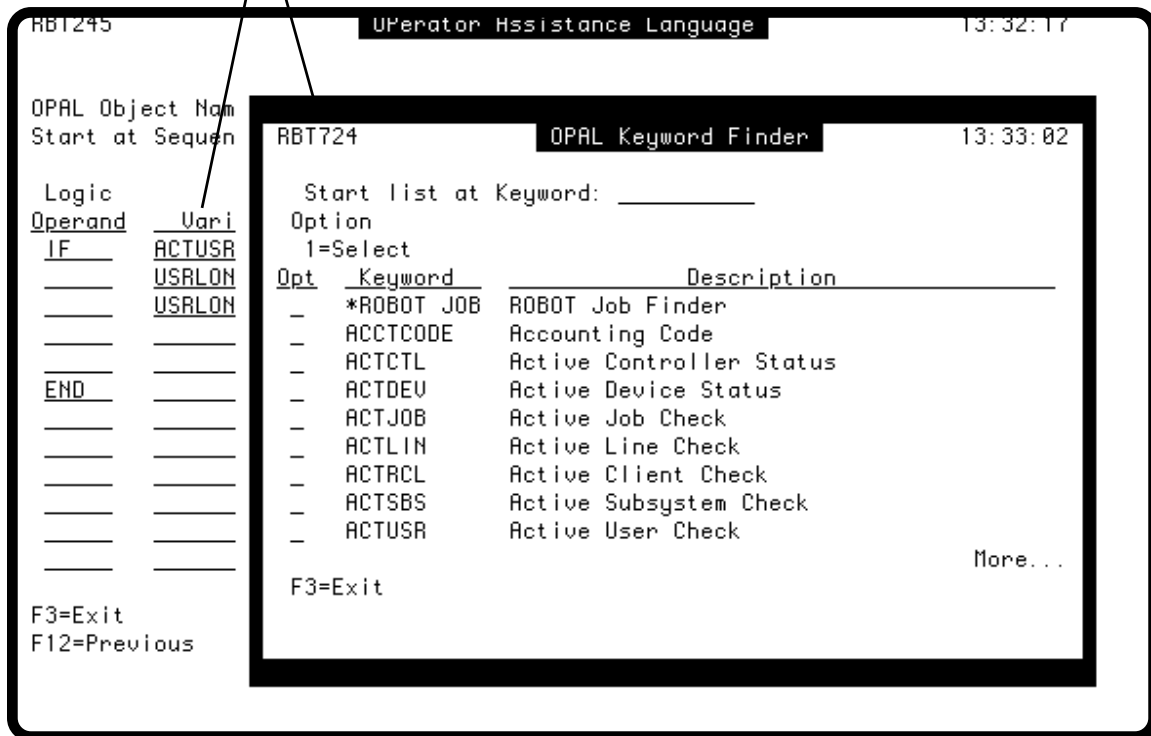
Press **function key 19** to check the logic of the OPAL code you've entered. It checks that logic operands are in the correct order and matches TAG and GOTO statements. In addition to using this logic check, you should test your OPAL code using the simulator provided before putting the code into production. See Using the OPAL Simulator, later in this section.

### Defining OPAL Code

You can edit your OPAL code at any time. Robot Schedule provides prompting for logic operands, variables, and operations. Some statements also support prompting in the Operation Values field (see next page). To display a prompt window, press **function key 4** in the field you want to prompt.

Each line of OPAL code is assigned a sequence number automatically. You can change the sequence numbers of the lines by typing new sequence numbers and pressing Enter. To renumber lines by equal intervals, press **function key 18**. You can delete a line by blanking out the entire line, including the sequence number.

Place your cursor in a field and press **function key 4** to select from a list of keywords for that field.







### Prompting the Operation Values Field

If you are working with an operation that accesses a command, press **function key 4** in the Operation Values field to display the command prompt panel. For example, the RBASNDMSG command prompt displays when prompting the Operation Values field after entering the PAGE operation. In this case, you see **\*\*Press F4 to see command\*\*** in the Operation Values field.

```

RBT245          Operator Assistance Language          13:37:13

OPAL Object Name . . . : SKIPSTATUS If job is skipped, page operator
Start at Sequence . . . : ____

Logic
Operand  Variable  Operation  Operation Values  Seq
----  -
IF      ACTDEV    NE        PRT01             10
AND     OUTQ      EQ        PRT01             20
        _____  SKIP      _____        30
END     _____  _____  _____        40
IF      ACTRCL   NE        DAVE              50
AND     USER     EQ        DAVE              60
END     _____  _____  _____        70
IF      SKIPSTATUS EQ        YES             80
        _____  PAGE     ** Prompt to see command **  90
        _____  _____  _____        100
        _____  _____  _____        110
        _____  _____  _____        120
                                           More...

F3=Exit      F4=Prompt      F7=ROBOT Variables
F12=Previous F18=Resequence F19=Logic Check
  
```

```

Send a Message to a Pager (RBASNDMSG)

Type choices, press Enter.

Message Text . . . . . Backups skipped
_____
_____
_____
_____

Pager ID or Broadcast List . . . jeff      Character value . . .
Response Required . . . . . *SETUP *SETUP, *YES, *NO
Truncate Message for Vendor . . *NO     *YES, *NO

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
  
```

## Creating the OPAL Program

After you've entered your OPAL code, press **function key 3** to exit the panel. Robot Schedule checks the syntax and logic of your OPAL statements automatically. A confirmation window displays asking if you are ready to submit the code to be compiled. Press Enter to create the OPAL program. If you do not want to create the OPAL program at this time, press **function key 3** to save the work you've done and return to the Maintain OPAL Objects panel.

If you have entered new statements or edited existing ones, press **function key 3** to exit the panel. A confirmation window asking if you want to create the OPAL program.

Enter a **1** to create or revise your OPAL object immediately.

```
RBT245          Operator Assistance Language          13:18:27

OPAL Object Name . . . : ACTUSER    Check for any active users before runnin
Start at Sequence . . . : _

Logic
Operand
IF
____
____
____
END
____
____
____
____

RBT770          Confirmation Window          13:18:29

Create the OPAL Program.

1=Select
 1 Yes, Continue

F3=Cancel Confirmation

F3=Exit
F12=Previous    F18=Resequenece    F19=Logic Check
Logic checking being processed. Please wait.
```

Press **function key 3** to save changes you made to your OPAL code and return to the Maintain OPAL Objects panel. Use this option if you want to make additional changes to the OPAL code or if you're not ready to have the OPAL code used.

### Finding Where an OPAL Object is Used

Robot Schedule does not allow you to delete an OPAL object that is attached to a job. However, you can easily find out which jobs are using an object by selecting **option 3** from the Options menu on the Maintain OPAL Objects panel. You must remove the OPAL object from each job listed on this panel before you can delete the object.

```
RBT701      Options      13:43:36
1=Select Option
- > 1=Select OPAL Object
- > 2=Maintain OPAL Detail
- 3=OPAL Where Used
- 5=OPAL Simulator
- 6=Print OPAL
- 7=Scan/Replace OPAL

Bottom

F3=Exit  F11=More Info
```

Select **option 3** to display the Where is an OPAL Object Used panel.

```
RBT227      Where is an OPAL Object Used      13:43:02
OPAL Object Name . . . : ACTUSER      Check for any active users before r
Start list at ROBOT Job : _____

  Type  Job Name  Group Name  Description
  ---  -
  C     BACKUP1               Evening Backup
  C     PWRDOWNSYS             System Powerdown
  C     RSUBBACKUP             Restricted State Backup

Bottom

F3=Exit  F21=Command Line
```

The jobs that use the selected OPAL object.

## Using the OPAL Simulator

The OPAL Simulator lets you test your OPAL code before putting it into production. The top part of the panel consists of several input fields. Enter your OPAL object and Robot Schedule job information and press **Enter**. The Simulator tells you if the job will be skipped or delayed as a result of your OPAL code.

```
RBT701      Options      13:43:36
1=Select Option
- > 1=Select OPAL Object
- > 2=Maintain OPAL Detail
- 3=OPAL Where Used
- 5=OPAL Simulator
- 6=Print OPAL
- 7=Scan/Replace OPAL

Bottom

F3=Exit  F11=More Info
```

Select **option 5** on the Maintain OPAL Objects panel to display the OPAL Simulator panel.

```
RBT235      OPAL Simulator      16:44:26

Input Fields:
OPAL Object Name . . . . . : ACTUSER      (F4=Prompt)
ROBOT Job Name . . . . . :  BACKUP1      (F4=Prompt)
ROBOT Group Name . . . . . :
Run Date . . . . . : 101900
Run Time . . . . . : 1644
Should the Files be updated . . . : N      (Y=Yes, N=No)

Output Fields:
Will this job be skipped . . . . :
Will this job be delayed . . . . :

NOTE: Selecting "Y" to update files may effect your Job Schedule List.
Use with caution.

F3=Exit  F4=Prompt  F7=OPAL Trace  F13=Robot Job
F15=Robot Temp. Job  F21=Command Line
```

The OPAL Object name is the name of the object you selected on the Maintain OPAL Objects panel. You can type in a different name (or press **function key 4** to select one from a finder) if you want to test a different object.

Enter a Robot Schedule Job Name or press **function key 4** to select a job from the Robot Job Finder panel.

If you want to test this job as a member of a group, enter the group name.

*Using the OPAL Simulator*

```
RBT235                                OPAL Simulator                                16:42:02

Input Fields:
OPAL Object Name . . . . . : ACTUSER      (F4=Prompt)
ROBOT Job Name . . . . . : BACKUP1      (F4=Prompt)
ROBOT Group Name . . . . . :
Run Date . . . . . : 101900
Run Time . . . . . : 1641
Should the Files be updated . . . : N      (Y=Yes, N=No)

Output Fields:
Will this job be skipped . . . . : NO
Will this job be delayed . . . . : NO

NOTE: Selecting "Y" to update files may effect your Job Schedule List.
      Use with caution.

F3=Exit      F4=Prompt      F7=OPAL Trace      F13=Robot Job
F15=Robot Temp. Job      F21=Command Line
```

Enter the run date you want to test.

Enter the run time you want to test.

If the job name specified on this panel has an OPAL temporary job on the Job Schedule List, press **function key 15** to display the Initial Job Setup window for the temporary job. You then can view or update the job information.

The results of the simulator run. The fields indicate if Robot Schedule will skip or delay the job.

If you have the Logging Level on the OPAL Object panel set to Build Trace Data, press **function key 7** to display the trace listing.

If you want to edit the job setup based on the results on this test, press **function key 13** to display the Initial Job Setup panel for the job.

## Using the OPAL Simulator

```

RBT235                OPAL Simulator                13:45:52

Input Fields:
OPAL Object Name      : ACTUSER      (F4=Prompt)
ROBOT Job Name       : BACKUP1      (F4=Prompt)
ROBOT Group Name     :
Run Date            : 101600
Run Time           : 1345
Should the Files be updated : N      (Y=Yes, N=No)

Output Fields:
Will this job be skipped . . . :
Will this job be delayed . . . :

NOTE: Selecting "Y" to update files may effect your Job Schedule List.
      Use with caution.

F3=Exit      F4=Prompt      F7=OPAL Trace      F13=Robot Job
F15=Robot Temp. Job      F21=Command Line
  
```

If you do **NOT** want to change the Robot Schedule job record as a result of this test, enter an **N**. This is the default. If you do want to change the Robot Schedule job record as a result of this test, enter a **Y**.

**CAUTION:** If your OPAL code contains the operations that update the job (UPDJOB) or add minutes to the scheduled run time (ADDMIN), entering a Y can result in unintended changes to your job schedule. The table below shows the simulator action for each job operation, based on what you enter in this field.

Operation	Simulator Action Update=Y	Simulator Action Update=N
UPDJOB	Temporary job created	No temporary job, no changes to Robot Schedule Job Record
ADDMIN	Temporary job created, Job Delayed flag set to YES	No temporary job created, Job Delayed flag set to YES
UPDENV	Environment values changed in environment specified on Robot Schedule job record	Environment values NOT changed in environment specified on Robot Schedule job record
CHGDTADFN	Data area value changed	Data area NOT changed
SKIP	No temporary job created, Job Skipped flag set to YES	No temporary job created, Job Skipped flag set to YES
SNDRBTDTA	Sent	NOT sent
SNDMSG	Message sent	No message sent
SNDBRKMSG	Message sent	No message sent
SENDMC	Message sent	No message sent
PAGE	Page initiated	No page initiated

### Displaying Trace Data

The OPAL Object panel includes a logging level field. If you set the Logging Level to Build Trace Data, you can display your trace results. After you have tested your code using the OPAL Simulator, press **function key 7** to display the OPAL Trace Statements panel. It shows you which OPAL statements actually executed.

```
RBT228          ROBOT OPAL Trace Statements          14:06:20

ROBOT Group . . . . . : *NONE
ROBOT Job . . . . . : BACKUP1      Evening Backup
ROBOT OPAL Object . . : ACTUSER    Check for any active users before running job

Logic
Operand  Variable  Operation  Operation Values  Sequence
-----  -
          QUIT                               70

F3=Exit      F12=Previous      F19>Delete Records      F21=Command Line

Bottom
```

The OPAL Trace Statements panel displays the statements that executed in the OPAL simulator test of this OPAL object.

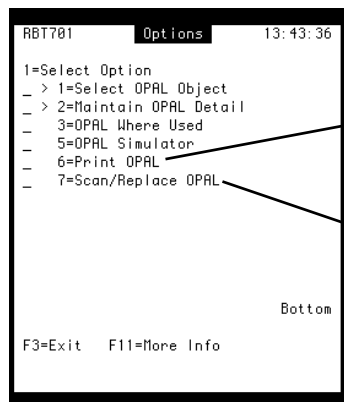
Press **function key 19** to remove the trace records for the simulated job run.

**Printing Your OPAL Code and Scan/Replace OPAL Values**

You can print out your OPAL code if you wish. Select **option 6** (Print OPAL) on the Maintain OPAL Objects option menu to print your OPAL object.

Select **option 7** (Scan/Replace OPAL Values) to display the Scan/Replace OPAL Values panel. This panel allows you to look for values in all your OPAL code and change them globally. Scan and replace works similarly to search and replace in a word processing program. Enter the information Robot Schedule should look for in the Scan Information fields. Robot Schedule tries to match all the items entered in these fields. Therefore, the more fields in which you make an entry the more restrictive the search becomes. When Robot Schedule finds a match, it replaces the Scan Information values with the Replace Values.

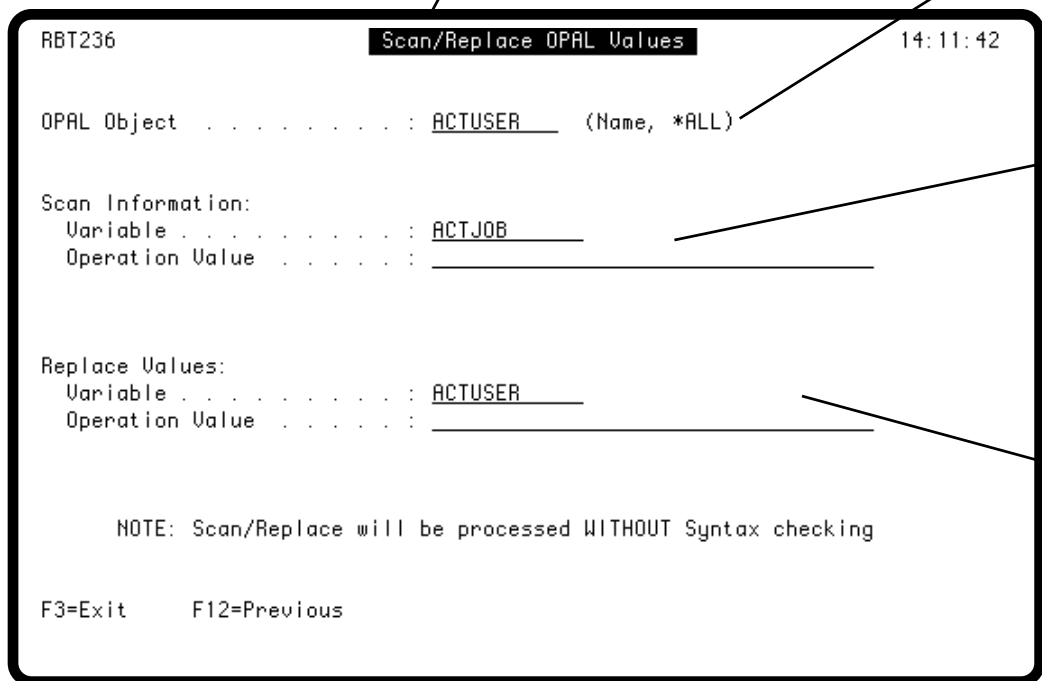
**Note:** The replacement **Operation Value** must have the same number of characters as the scan **Operation Value**.



Select **option 6** to print your OPAL object. The Select Output Queue window displays so you can specify an output queue.

Select **option 7** to display the Scan/Replace OPAL Values panel.

The Object name is the one you selected on the Maintain OPAL Objects window. You can change the object name or enter \*ALL to scan all OPAL objects.



Enter the Variable and/or Operation Value that Robot Schedule should search for.

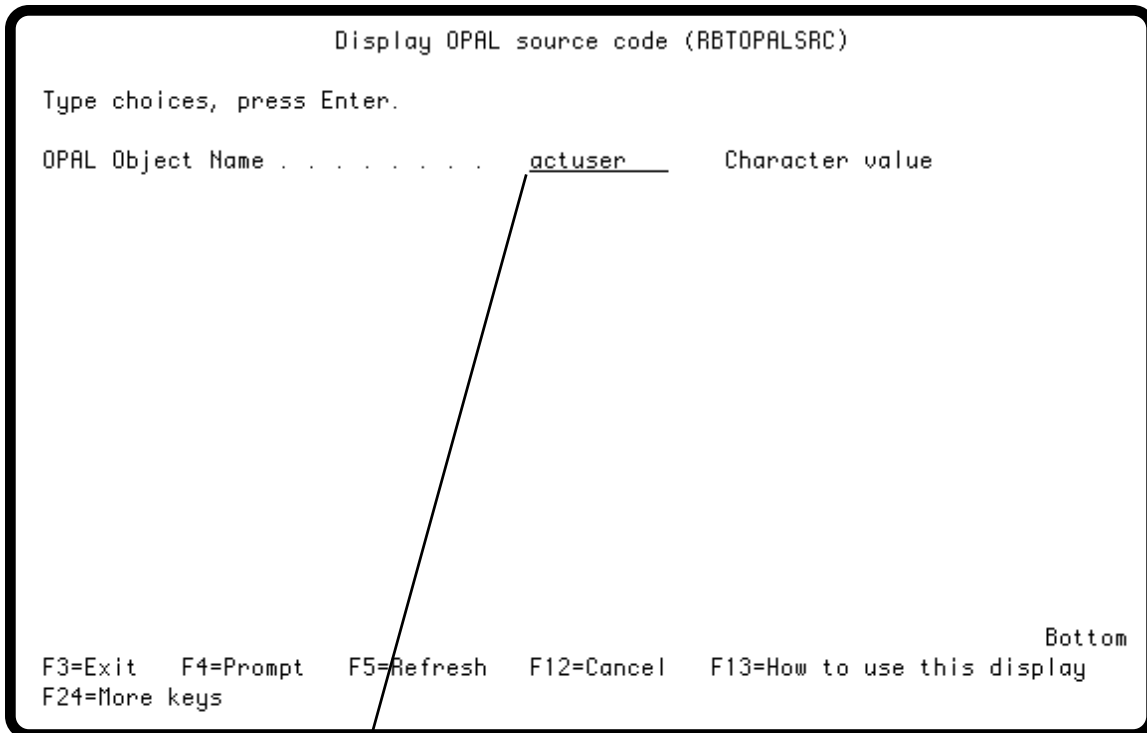
Enter the replacement Variable and/or Operation Value.



### Display OPAL Source Code—**RBTOPALSRC** Command

Robot Schedule provides a command that allows you to display the source code of your OPAL objects. Robot Schedule automatically checks the logic and syntax of the OPAL code when you exit the OPAL entry panel. The **RBTOPALSRC** command lets you go one step further in checking your code. It displays the source code the OPAL object generates via IBM's Source Edit Utility (SEU). Once you are in the utility, you can browse through your code or print it out.

The source code can be helpful in diagnosing problems with your OPAL object. If you are having trouble with an OPAL object, technical support may ask you to print out your source code and send it to us for analysis. Enter the command **ROBOTLIB/RBTOPALSRC** on a command line and press **function key 4** to see the prompt panel, shown below.



Enter the name of the OPAL object for which you want to display the OPAL source code.

## Maintain OPAL Tables

The Maintain OPAL Tables panel lists the names and descriptions of the tables you have defined. OPAL tables are used to store lists of values used with the OPAL keywords **INTABLE** and **NOTINTABLE**.

```
RBT1010          ROBOT OPAL Object Menu          15:19:00
OPALOBJ

Select one of the following:

1. Maintain OPAL Objects
2. Maintain OPAL Tables
3. Maintain Data Area Definitions

Selection: 2

F3=Exit      F21=Command Line
```

Select **option 2** on the Robot OPAL Object Menu to display the Maintain OPAL Tables panel. The panel lists all the OPAL tables you have defined to Robot Schedule.

```
RBT229          Maintain OPAL Tables          14:23:56

Start at Table . . . . . : _____

Options
A=Add/Copy/Delete      1=Select Table      2=Maintain Elements

Opt  OPAL Table      Description
--  -
1   EXECOUTQ      Executive OUTQs
   PRINTERS       Printer Table
   TAPEDEVICE     Tape devices

Bottom

F3=Exit      F6=Add Table      F21=Command Line
```

Enter a **1** in the **Opt** field to review or edit information on the OPAL Table panel.

Press **function key 6** to create a new table.

Enter a **2** in the **Opt** field to display the OPAL Table Elements panel.

### OPAL Table Panel

Enter a **1** in the **Opt** field on the Maintain OPAL Tables panel to display the OPAL Table panel. The OPAL Table panel allows you to enter a name and description for the table. Enter the table name in the Operation Values field when you use the OPAL operations **INTABLE** and **NOTINTABLE**.

Enter a name for the table. The table name can be up to 10 characters long. Use a logical name that describes how the table is used.

Enter a description of the table.

The screenshot shows a terminal window titled "OPAL Table" with the ID "RBT230" and a timestamp of "15:17:59". The prompt "Enter OPAL Table Information:" is displayed. The user has entered "PRINTERS" for the "OPAL Table" field and "Printer Table" for the "Description" field. At the bottom of the screen, the following function key instructions are visible: "F3=Exit", "F12=Previous", and "F21=Command Line".

```
RBT230                                     OPAL Table                                     15:17:59

Enter OPAL Table Information:
OPAL Table . . . . . : PRINTERS
Description . . . . . : Printer Table

F3=Exit    F12=Previous    F21=Command Line
```



## Maintain Data Area Definitions

The Maintain Data Area Definitions panel lists the definition names and descriptions of the data areas you have defined. Data Area Definitions are used with the OPAL keywords RTVDTADFN and CHGDTADFN. RTVDTADFN fills user fields with values from the defined data area; CHGDTADFN updates the defined data area with values from the OPAL user fields.

```

RBT1818          ROBOT OPAL Object Menu          15:19:08
OPALOBJ

Select one of the following:

1. Maintain OPAL Objects

2. Maintain OPAL Tables

3. Maintain Data Area Definitions

Selection: 3

F3=Exit      F21=Command Line
    
```

Select **option 3** on the Robot OPAL Object Menu to display the Maintain Data Area Definitions panel. The panel lists all the data area definitions you have defined to Robot Schedule.

```

RBT232          Maintain Data Area Definitions    15:29:10

Start at Data Object . . . . : _____

Options
A=Add/Copy/Delete      1=Select      2=Maintain Elements

Opt  Definition  Description          Data Area
      Name       Library
1   ACCOUNTING  Accounts Payables DTAARA Definition  ACTPAY01  QGPL
      APREPORT    Data area definition for AP Reports  *LDA
      ITOPER      IT Operations flag data area        ITOPER    QGPL

Bottom

F3=Exit      F6=Add Definition      F21=Command Line
    
```

Enter a **1** in the **Opt** field to review or edit information on the Data Area Definition panel.

Press **function key 6** to create a new data area definition.

Enter a **2** in the **Opt** field to display the Data Area Definition Elements panel.

**Data Area Definition Panel**

Enter a **1** in the **Opt** field on the Maintain Data Area Definitions panel to display the Data Area Definition panel. Use this panel to give a Robot Schedule name to a data area. You then can use the data area name in your OPAL code with the RTVDTADFN and CHGDTADFN keywords.

Enter a name for the data area definition. The data area definition name can be up to 10 characters in length. Use a logical name that describes how the data area definition is used.

Enter a description of the data area definition.

```
RBT233                               Data Area Definition                               15:35:08

Enter Data Area Information:
Data Area Definition . . . : ACCOUNTING
Description . . . . . : Accounts Payables DTAARA Definition
Data Area . . . . . : ACTPAY01
Library . . . . . : QGPL

F3=Exit   F12=Previous   F21=Command Line
```

Enter the name of a data area and its library. Or, enter \*LDA. When Robot Schedule executes the OPAL that references the \*LDA, it first checks to see if there is an LDA defined for the job in the Robot Schedule job record. If there isn't, it uses the actual LDA for the iSeries job.

**Data Area Definition Elements**

Enter a **2** in the Opt field on the Maintain Data Area Definitions panel to display the Data Area Definition Elements panel. Use this panel to assign user variables to portions of the data area you want to work with. When you use the RTVDTADFN keyword, the elements shown would retrieve values from the data area and put them in OPAL user variables for use in your OPAL code. When you use the CHGDTADFN keyword, the values in the user variables would update the values in the data area. If you need more information on the Robot Schedule OPAL user variables, see the Variables section of this User Guide.

Enter the beginning and ending positions within the data area to access.

Enter an OPAL user variable name. Valid names include USRFLG*n*, USRFLD*n*, and USRLONG*n*, where *n* is a value between 1 and 5.

Enter a description of what you are retrieving from the data area or what value you are updating.

RBT234 15:41:53

**Data Area Definition Elements**

Data Area Definition . . . : ACCOUNTING Accounts Payables DTAAAA Definition  
 Data Area . . . . . : ACTPAY01  
 Library . . . . . : QGPL  
 Start List at Sequence . . . : —

Positions		OPAL	Description	Sequence
Beg	End	Variable		Number
<u>10</u>	<u>16</u>	USRFLD1	Beginning Date	<u>10</u>
<u>24</u>	<u>30</u>	USRFLD2	Ending Date	<u>20</u>
<u>21</u>	<u>21</u>	USRFLG1	Flag For Active Users	<u>30</u>
<u>22</u>	<u>22</u>	USRFLG2	Flag For Active Tape Dev	<u>40</u>
<u>23</u>	<u>23</u>	USRFLG3	Flag To Run Monthly Rpt	<u>50</u>
<u>170</u>	<u>190</u>	USRLONG1	Transaction Field #	<u>60</u>
<u>65</u>	<u>75</u>	USRFLD3	Transaction Field Dept	<u>70</u>
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—

More . . .

F3=Exit                      F12=Previous                      F18=Resequence  
 F21=Command Line

To change the sequence of elements, type the new sequence numbers in the Sequence Number column and press Enter. To renumber the statements by equal intervals, press **function key 18**.

## ■ OPAL Statements

The OPAL language is easy to learn. It is a fixed format language, like RPG, but its syntax is like CL. To code OPAL statements, you just fill in fields on a panel. Help is available for each field.

OPAL code consists of a sequence of OPAL statements. The OPAL statements are performed in order unless an IF or GOTO statement changes the processing order.

Each OPAL statement is entered on a single line. (Like RPG, OPAL does not use continuation lines.) Each line has multiple fields as shown below.

The content of each field is indicated by its column heading.

RBT245      **Operator Assistance Language**      11:21:31

OPAL Object Name . . . . . PR\_ACTIVE    Check to see if Payroll is active  
 Start at Sequence . . . . . :    —

Logic Operand	Variable	Operation	Operation Values	Seq
IF	ACTJOB	EQ	PR140	10
		SNMSG	'Payroll is still ative' JIM	20
		SKIP		30
ELSE				40
IF	ACTLIN	NE	ACH_LINE	50
		SNMSG	'Check the ACH Line' JIM	60
		ADMIN	15	70
END				80
END				90
				100
				110
				120

F3=Exit                      F4=Prompt                      F7=ROBOT Variables  
 F12=Previous                F18=Resequenece                F19=Logic Check

More...

The **Seq** field is not part of the OPAL statement; it contains the sequence number of the statement.

## ■ OPAL Fields

The following fields in a line constitute an OPAL statement. Each OPAL example in this manual uses a graphic similar to the one below to show you the fields used by the statements.

Logic Operand	Variable	Operation	Operation Values
---------------	----------	-----------	------------------

The pages that follow describe what you can enter in the OPAL statement fields.



## ■ Logic Operands

The first field in an OPAL statement is the Logic Operand field. Logic operands are used to define logic control for the OPAL code, that is, which OPAL statement is performed next.

## ■ Logic Control

Seven logic control structures are available in OPAL for Robot ScheduleE: IF, AND, OR, ELSE, END, GOTO and TAG. The IF structure specifies conditions that must be met before a set of operations is performed. A GOTO operation changes the next statement processed to the specified TAG statement.

You do not need logic control in your OPAL code if all operations in the code are to be performed for every job processed by that OPAL code. For example, the following OPAL code sends a message each time a job that uses this code runs.

Logic Operand	Variable	Operation	Operation Values
		SNDMSG	'Check ACH line.' SYSOPR

## ■ Conditional Operations

To make operations conditional, that is, to specify conditions that must be met before other operations are performed, use Logic Operands in your OPAL code.

The following is an example of an IF-ELSE structure. The operation in the IF structure is performed only if the conditions are met. For example, suppose you want to make sure the job PR140 is NOT active when you run another payroll job. The following OPAL code will check if PR140 is the active job. If it is, it sends a message to Jim and skips this job. If PR140 is NOT the active job, the code then checks to make sure a certain communications line this job needs (the ACH line) is active. If it is not, it sends a message to Jim and reschedules its run time to 15 minutes later.

Logic Operand	Variable	Operation	Operation Values
IF	ACTJOB	EQ	PR140
		SNDMSG	'Payroll is still active' JIM
		SKIP	
ELSE			
IF	ACTLIN	NE	ACH_LINE
		SNDMSG	'Check the ACH Line' JIM
		ADDMIN	15
END			
END			

## ■ IF—Perform If Conditions are Met

Use an IF statement to test for one or more conditions. If the conditions are true, the operations following the condition list are performed. If the conditions are not true, the program continues at the statement following the operation list.

If the operations should be performed just once if the conditions are true, use an IF statement.

## ■ Syntax

Logic Operand	Variable	Operation	Operation Values
IF	required	required	required

## ■ Examples

Logic Operand	Variable	Operation	Operation Values
IF	CALENDAR	EQ	FISCAL
IF	STARTRANGE	LT	SYSTIME

## ■ Multiple Conditions

You can add more conditions to the IF using AND and OR statements after the IF.

For example, if PRT01 is not active and the output queue for this job is PRT01, the OPAL code changes the output queue to PRT02.

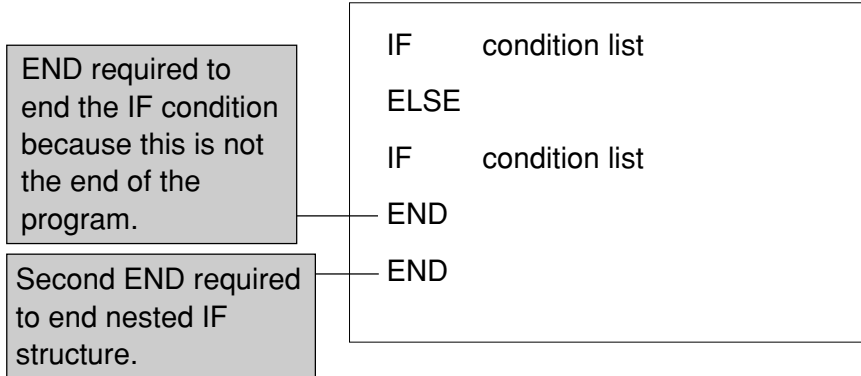
Logic Operand	Variable	Operation	Operation Values
IF	ACTDEV	NE	PRT01
AND	OUTQ	EQ	PRT01
	OUTQ	CHGTO	PRT02
		UPDJOB	
END			

## ■ Nested IFs

IF structures can be nested, that is, an IF structure can be enclosed in another IF structure.

■ **END Statements Required**

For every IF statement, there must be a corresponding END statement. However, END statements can be omitted at the end of the OPAL program. Consider this program structure:



## ■ AND—Add a Condition to a Condition Set

Use the AND statement to add a condition to an IF condition list. The AND connects the condition to the preceding condition in the list. Both conditions must be true for a true result; if either condition is false, the result is false.

Use an AND statement to add another condition to a set of conditions. (In contrast, an OR statement starts a new condition set.) All conditions in a set must be true for a true result.

## ■ Syntax

Logic Operand	Variable	Operation	Operation Values
<b>AND</b>	<b>required</b>	<b>required</b>	<b>required</b>

## ■ Examples

Logic Operand	Variable	Operation	Operation Values
AND	USRNBR1	EQ	USRNBR2
AND	DATEOBJ	NE	FISCALEX

## ■ Any Number of Conditions in a Set

A condition set can contain any number of conditions. For example, the following IF statement has three conditions that must all be true for the operation to be performed:

Logic Operand	Variable	Operation	Operation Values
IF	ACCTCODE	EQ	PAYROLL
AND	SCHEDULE	NE	H
AND	OUTQ	EQ	PRT01
		SNDMSG	'Load checks' SYSOPR
END			

■ **OR—Start New Condition Set**

■ **Syntax**

Use the OR statement to start a new set of conditions in an IF condition list. (To add a condition to an existing set of conditions, use the AND statement.)

Logic Operand	Variable	Operation	Operation Values
<b>OR</b>	<b>required</b>	<b>required</b>	<b>required</b>

■ **Examples**

Logic Operand	Variable	Operation	Operation Values
OR	ENDRANGE	GT	SYSTIME
OR	GROUP	NE	PAYROLL

■ **OR Starts a New Condition Set**

Each OR statement in an IF condition list starts a new set of conditions. For example, the OR statement in the following IF condition list starts a second condition set. Each set contains one condition; if either condition is true, the result is true.

The following OPAL code retrieves information about DKJFILE in DKJLIB. If the file does not exist, or does not contain records, or is locked, the job is skipped.

Logic Operand	Variable	Operation	Operation Values
		RTVOBJSTS	DKJLIB/DKJFILE *FILE
IF	OBJSTS	NE	EXISTS
OR	OBJSTS	NE	RECORDS
OR	OBJSTS	EQ	LOCKED
		SKIP	
END			

■ **Each Set Tested Separately**

Each condition set is tested separately. Therefore, for the condition set to be true, any statement in the set can be true.

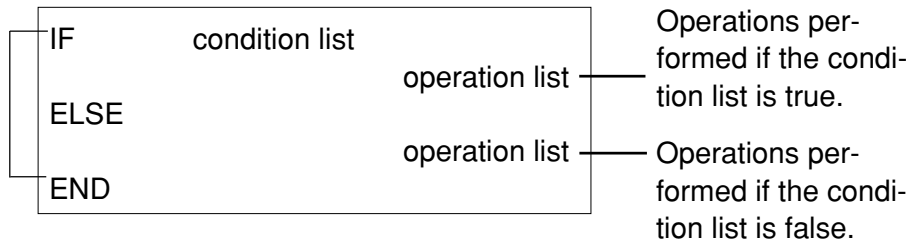
## ■ ELSE—If Conditions Not Met

## ■ Syntax

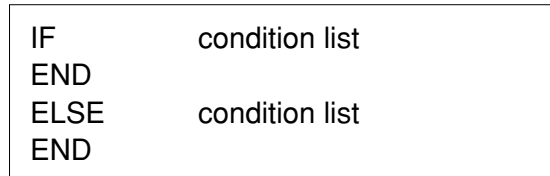
Use the ELSE condition to extend an IF structure so it includes processing to be done only if the preceding IF condition lists are not true.

Logic Operand	Variable	Operation	Operation Values
ELSE	blank	blank	blank

The simplest IF-ELSE structure is as follows:

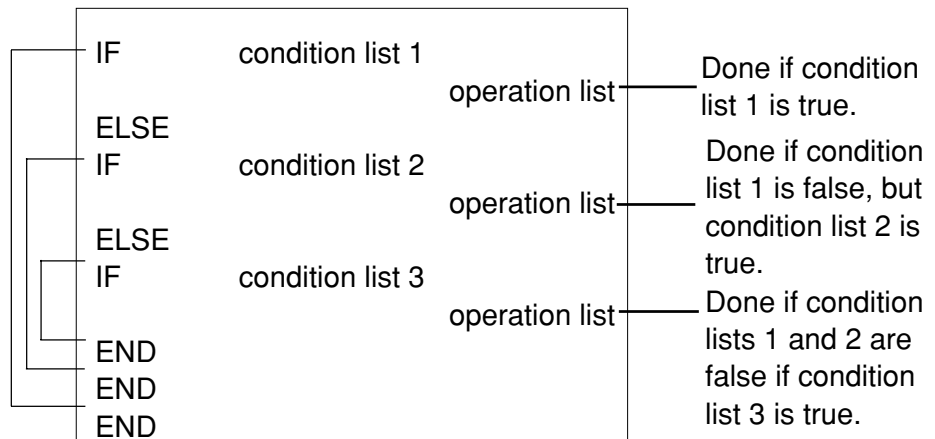


Each IF statement must be followed by a corresponding END statement. However, an END is not allowed before an ELSE. The following IF-THEN-ELSE structure is invalid:



## ■ Extends IF for Additional Conditions

Additional ELSE clauses can be added to process other conditions. Only one operation list in the IF structure is executed. For example:



■ **END—Ends Operation List**

■ **Syntax**

■ **When END is Optional**

Every IF statement must be followed by a corresponding END statement. The END statement marks the end of the processing for the IF.

However, an END statement is optional if it is at the end of the program.

Logic Operand	Variable	Operation	Operation Values
<b>END</b>	<b>blank</b>	<b>blank</b>	<b>blank</b>

The following outline shows where END statements are optional:

IF	condition list	operation list
IF	condition list	operation list
ELSE		operation list
END		operation list
ELSE		operation list
END		operation list

Required to end nested IF statement.

Optional at the end of the OPAL program.

So, if you remove the optional END, the remaining code is as follows.

IF	condition list	operation list
IF	condition list	operation list
ELSE		operation list
END		operation list
ELSE		operation list
		operation list

## ■ Operation Statements Tell Robot Schedule What to Do

OPAL code exists in Robot Schedule to allow the user to change or skip the scheduling of a Robot Schedule job based on the conditions that exist at the time the OPAL code is processed. The OPAL statements that tell Robot Schedule what to do are called operation statements. OPAL code must include at least one operation statement. Any other statements in the OPAL code just determine which operation statements are performed.

## ■ Syntax

The general syntax of an operation statement is as follows:

Logic Operand	Variable	Operation	Operation Values
optional	optional	required	optional

When you code an operation statement, you always fill in the Operation field; you provide an Operation Value if one is needed by the operation. The CHGTO, ADD, and SUB operations require an OPAL variable name in the Variable field. Usually, the Logic Operand field is left blank.

## ■ Examples

For example, the following statement uses only the Operation field:

Logic Operand	Variable	Operation	Operation Values
		SKIP	

Many operation statements use the Operation and Operation Value fields, as follows:

Logic Operand	Variable	Operation	Operation Values
		SNDBRKMSG	'Get off the system' *ALLWS

The following CHGTO operation also uses the Variable field:

Logic Operand	Variable	Operation	Operation Values
	USRFLD1	CHGTO	JOBNBR



■ **A Condition  
Must Specify a  
Comparison**

When you specify a condition, you must specify a comparison in the Operation field. The comparison specifies a relationship between the value of the OPAL variable in the Variable field and the value in the Operation Value field. If the relationship exists, the condition is true.

For example, the following condition specifies the greater than comparison (GT) in the Operation field. For the condition to be true, the system time (specified by the OPAL variable SYSTIME) must be greater than the operation value (1900).

Logic Operand	Variable	Operation	Operation Values
IF	SYSTIME	GT	1900

The comparisons you can specify in the Operation field are as follows:

- Equal (EQ)
- Not equal (NE)
- Greater than (GT)
- Greater than or equal to (GE)
- Less than (LT)
- Less than or equal to (LE)
- Contains (CT)
- Doesn't contain (DC)
- Matches table value (INTABLE)
- Doesn't match table value (NOTINTABLE)

■ **EQ or NE:  
Equal or  
Not Equal**

**EQ** Equal; the values match exactly.

**NE** Not equal; the values do not match exactly.

### ■ GT, GE, LT, LE: Order Comparisons

The next four comparisons are for comparing the order of the values. For numeric values, a numeric comparison is done. For alphanumeric values, an alphanumeric comparison is done, character by character, from left to right. For example, the numeric value 12 is greater than both 1 and 2. But, the alphanumeric value 12 is greater than 1, but less than 2.

**GT** Greater than; the OPAL variable value precedes the operation value.

**GE** Greater than or equal to; the OPAL variable value precedes or is equal to the operation value.

**LT** Less than; the OPAL variable value follows the operation value.

**LE** Less than or equal to; the OPAL variable value follows or is equal to the operation value.

### ■ CT, DC: Contains or Doesn't Contain

The next two comparisons compare the characters in the variable in the Variable field with the characters in the Operation Value field of the OPAL statement. These comparisons are case-sensitive. Character strings in upper and lower case must be enclosed in single quotation marks. For example, the character sequence SALE is contained in SALES, NEWSALE, and OLDSALES, but it is not contained in SALUPD or Sale.

**CT** Contains; the character sequence from the operation value is also in the OPAL variable value.

**DC** Doesn't contain; the character sequence from the operation value is not in the OPAL variable value.

■ **INTABLE,  
NOTINTABLE:  
Value Found or  
Not Found in a  
Table**

The next two comparisons compare the value of the OPAL variable to the values listed in a table. You must create an OPAL table and enter table elements for this comparison. The name of the table can be up to 10 characters long. The table name is placed in the Operation Values field. (See “Maintain OPAL Tables” earlier in this User Guide for more information on creating tables.)

**INTABLE** The OPAL variable value precisely matches a value in the referenced table.

**NOTINTABLE** The OPAL variable value does not match any value in the referenced table.

**Example:** Suppose you have had some trouble with jobs that have output queues that are not attached to printers. You could create an OPAL table called PRINTER that lists all the output queues that have printers. The OPAL code will check if the output queue is attached to a printer and make sure it is active. Otherwise, it will default the output queue to PRT01.

Logic Operand	Variable	Operation	Operation Values
IF	OUTQ	INTABLE	PRINTER
AND	ACTDEV	NE	OUTQ
	OUTQ	CHGTO	PRT01
		UPDJOB	
END			

## Operation Value Field

### ■ Operation Values Field

Many OPAL conditions and operations require a value in the Operation Values field. The value can be a constant or a variable. Valid values for each operation are described in the Job Operations section.

**Note:** Most Operation fields do not require use of quotes with the operation value keyword, regardless of whether the operation value is character or numeric.

**Example:** The first statement below compares the value of the variable CALENDAR to the constant FISCAL. The second statement compares the values of two OPAL variables — ACTUSR and USER:

Logic Operand	Variable	Operation	Operation Values
IF	CALENDAR	EQ	FISCAL
IF	ACTUSR	NE	USER

### ■ OPAL Constants

The following are special OPAL constants:

**BLANK** Blank characters to fill the field.  
**YES** True logical value.  
**NO** False logical value.

### ■ OPAL Variables

You can use any Robot Schedule OPAL variable to represent a value. The following is a list of variables available in Robot Schedule OPAL:

<b>ACCTCODE</b>	<b>EJOB</b>	<b>JOB</b>	<b>RBTJOBNAME</b>
<b>ACTCTL</b>	<b>EJOB</b>	<b>JOB</b>	<b>RBTJOB</b>
<b>ACTDEV</b>	<b>EJOB</b>	<b>JOB</b>	<b>SB</b>
<b>ACTJOB</b>	<b>EJOB</b>	<b>JOB</b>	<b>SCH</b>
<b>ACTLIN</b>	<b>ELIB</b>	<b>JOB</b>	<b>SKIP</b>
<b>ACTRCL</b>	<b>EMSG</b>	<b>JOB</b>	<b>STAR</b>
<b>ACTSBS</b>	<b>EMSG</b>	<b>LAST</b>	<b>SYSTIME</b>
<b>ACTUSR</b>	<b>EMSG</b>	<b>LIB</b>	<b>SYSDATE</b>
<b>ATPOOL</b>	<b>ENDR</b>	<b>MAX</b>	<b>SYSTEM</b>
<b>CALENDAR</b>	<b>EOUT</b>	<b>MAX</b>	<b>THIS</b>
<b>CURLIB</b>	<b>EOUT</b>	<b>MSG</b>	<b>TIME</b>
<b>DATEOBJ</b>	<b>EPAG</b>	<b>MSG</b>	<b>USER</b>
<b>DAY</b>	<b>EUSE</b>	<b>MSG</b>	<b>USRCNT</b>
<b>DAYMTH</b>	<b>GR</b>	<b>NON</b>	<b>USRFLD</b>
<b>EACCTCODE</b>	<b>JOB</b>	<b>OUT</b>	<b>USRFLG</b>
<b>EATPOOL</b>	<b>JOB</b>	<b>OUT</b>	<b>USRLONG</b>
<b>ECALENDAR</b>	<b>JOB</b>	<b>OPAL</b>	<b>USRNBR</b>
<b>ECURLIB</b>	<b>JOB</b>	<b>PAG</b>	<b>WEEKNO</b>
			<b>WORKDAY</b>

**Note:** OPAL treats a special value keyword (\*USRPRF, \*JOB, and so forth) as the keyword. It does not resolve the keyword to a value.

## ■ Job Operations

OPAL's special job operations allow you to skip a job or add minutes to a job's scheduled run time. You can also use the SKIPSTATUS variable to find out if the skip flag has been set by the various conditions you choose to test. Job operations make temporary changes to the job.

## ■ SKIP — Skip Job

The **SKIP** operation causes the current job to be skipped, meaning it will not run. The completion code for a skipped job is "K" on the Job Schedule List and Job Completion History panel.

**Operation value:** None.

**Example:** The following OPAL code retrieves information about DKJFILE in DKJLIB. If the file does not exist, or does not contain records, or is locked, the job is skipped.

Logic Operand	Variable	Operation	Operation Values
		RTVOBJSTS	DKJLIB/DKJFILE *FILE
IF	OBJSTS	NE	EXISTS
OR	OBJSTS	NE	RECORDS
OR	OBJSTS	EQ	LOCKED
		SKIP	
END			

## ■ SKIPSTATUS — Will Job Be Skipped?

The SKIPSTATUS variable lets you check whether or not the SKIP flag has been set earlier in the OPAL code. It is especially useful when you have many different IF conditions, and you need to know whether or not the job will be skipped.

**Operation:** The only valid comparisons are EQ and NE.

**Operation Values:** YES (or Y), NO (or N). YES means the job will be skipped; NO means it will not be skipped.

**Note:** This keyword cannot be used with the CHGTO operation.

**Example:** The following code looks at two conditions. Condition 1: PRT01 is not active, but it is the OUTQ for the job. Condition 2: DOUG's Robot/CLIENT is not active, but he is the user profile listed on the Robot Schedule job record. SKIPSTATUS will equal YES (true) if either condition is true.

Logic Operand	Variable	Operation	Operation Values
IF	ACTDEV	NE	PRT01
AND	OUTQ	EQ	PRT01
		SKIP	
END			
IF	ACTRCL	NE	DOUG
AND	USER	EQ	DOUG
		SKIP	
END			
IF	SKIPSTATUS	EQ	YES
		SNDMSG	"Backups skipped." OPR
		PAGE	Press F4 to see command
END			

■ **ADDMIN — Add Minutes**

The **ADDMIN** operation suspends the current job and creates a temporary job with the Schedule Override Code of NEXT. The temporary job will be processed after the number of minutes specified in the Operation Values field have elapsed. OPAL temporary jobs start with OP in the description field on the Job Schedule List.

**Note:** If more than one ADDMIN operation statement is processed by your OPAL code, they are not cumulative. You should add the QUIT operation statement after each ADDMIN statement to avoid this issue, and improve the performance.

So what happened to the current job? The current job is placed in a suspended state and cannot be selected on the Job Schedule List until the temporary job is processed. When the temporary job completes (either normally or abnormally), the job completion history is recorded under the current job and the temporary job disappears from the schedule.

**Operation value:** Number of minutes to add to the run schedule (1-1440).

**Example:** If a certain user currently is signed on, wait 5 minutes, and try again.

Logic Operand	Variable	Operation	Operation Values
IF	ACTUSR	EQ	QSYSOPR
		ADDMIN	5
END			

**Tip:** It is possible to create an OPAL object that just keeps adding minutes and the job never gets to run. How can you get out of this kind of loop? You have a few choices:

- Go to the Exception Scheduling panel (**option 10** from the Job Schedule List) for the **temporary** job and blank out the name of the OPAL object
- Delete the **temporary** job from the Job Schedule List
- Go to the OPAL Object Maintenance panel and make the OPAL object inactive. (If you use this choice, it will affect all Robot Schedule jobs that use this OPAL object.)

Then edit your object and try again.

**Tip:** If you are using the ADDMIN operation with RTVJOB, see the section on RTVJOB to see how the job is affected.

### ■ RTVMQDEPTH — Retrieve WebSphere MQ Message Queue Depths

Robot Schedule can monitor the message queue depths of IBM's MQ Series (5.2 and higher) messaging software on the iSeries or AS/400. Monitoring queue depths prevents critical WebSphere MQ situations or failures from occurring and lets you know, before running critical jobs, if messages on the WebSphere MQ are backed up.

Robot Schedule uses an Operator Assistance Language (OPAL) operation to check message queue depths. Using Robot Schedule, you can check queue depths as often as is necessary. If a message queue depth reaches a maximum that you specify, Robot Schedule can start any job you choose, start a specific job to correct the problem, execute a warning message, page an operator, and more.

To use Robot Schedule to monitor WebSphere MQ queue depths, start the message queue manager with the STRMQM command and start the message queue manager command server with the STRMQMCSVR command. In addition, RBTUSER needs \*USE authority to the QMQMADM user profile.

The following pages contain a sample scenario that outlines how Robot Schedule can be used to monitor WebSphere MQ message queue depths.



## RTVMQDEPTH — Retrieve WebSphere MQ Message Queue Depths

The OPERator Assistance Language panel below contains OPAL code for a sample scenario that specifies when the message queue depth is greater than 50 messages, a pager message is sent to notify an operator of a potential problem.

RTVMQDEPTH is the OPAL operation that retrieves the actual number of queued messages.

Set the maximum number of queued messages allowed here.

Press **function key 4** to display the Extended Command Entry panel. This is where you enter the message queue manager and message queue names. See the example on the following page.

```

RBT245                               Operator Assistance Language                               15:15:10

OPAL Object Name . . . . : MQSERIES      Check WebSphere MQ message queue depth
Start at Sequence . . . . : _____

Logic
Operand  Variable      Operation      Operation Values      Seq
-----  -
      USRCNT1          CHGTO         50
      RTVMQDEPTH
IF      MQDEPTH        GT            USRCNT1              30
      SNDMSG          MSG('MQDEPTH > 50') TOUSR(MQUSER)  40
      PAGE            ** Prompt to see command **        60
END
      _____      _____      _____      _____      70
      _____      _____      _____      _____      80
      _____      _____      _____      _____      90
      _____      _____      _____      _____     100
      _____      _____      _____      _____     110
      _____      _____      _____      _____     120
      _____      _____      _____      _____     130
More...

F3=Exit      F4=Prompt      F7=ROBOT Variables
F12=Previous F18=Resequence F19=Logic Check
    
```

Compares the actual number of messages (MQDEPTH value) to a numeric variable (the set value, in this case, 50).

If the message queue is greater than 50, Robot Schedule sends a pager message to an operator.

The OPAL keyword, RTVMQDEPTH, returns the value MQDEPTH, which is the actual number of queued messages.

## ■ RTVMQDEPTH — Retrieve WebSphere MQ Message Queue Depths

Enter the message queue manager and message queue names here, separated by one space. Press **function key 3** to return to the OPerator Assistance Language panel.

```

RBT246                               Extended Command Entry                               15:09:57

OPAL Object . . : MQSERIES      Sequence: 20

Extended Value  CYBRKING.SYSTEM.ADMIN.QMGR.EVENT
_____
_____
_____

F3=Exit      F4=Prompt      F7=ROBOT Variables      F12=Previous
    
```

The following is the completed OPerator Assistance Language panel for the sample scenario. We recommend that you call Technical Support at 952-933-0609 for specific instructions on setting up your system to monitor WebSphere MQ queue depths.

```

RBT245                               Operator Assistance Language                               15:15:10

OPAL Object Name . . . : MQSERIES      Check WebSphere MQ message queue depth
Start at Sequence . . . : _____

Logic
Operand  Variable      Operation      Operation Values      Seq
_____  _____      _____      _____              ____
_____  USRCNT1          CHGTO         50                      10
_____  _____      RTVMQDEPTH   CYBRKING.SYSTEM.ADMIN.QMGR.EVENT  20
IF       MQDEPTH        GT           USRCNT1                 30
_____  _____      SNDMSG       MSG('MQDEPTH > 50') TOUSR(MQUSER)  40
_____  _____      PAGE         ** Prompt to see command **      60
END      _____      _____      _____              70
_____  _____      _____      _____              80
_____  _____      _____      _____              90
_____  _____      _____      _____             100
_____  _____      _____      _____             110
_____  _____      _____      _____             120
_____  _____      _____      _____             130
More...

F3=Exit      F4=Prompt      F7=ROBOT Variables      F19=Logic Check
F12=Previous F18=Resequence
    
```

### ■ Reading or Changing Job Setup

Robot Schedule OPAL also allows you to read and temporarily change job setup information by using the RTVJOB and UPDJOB operations.

When you attach OPAL code to a job, an implied RTVJOB operation is performed for the current job. As long as you are working with the current job, you do not need to code an explicit RTVJOB operation. However, if you retrieve information about a different job and then want to read or change information about the current job, you need a RTVJOB statement to get the values for the current job again (see example).

Many of the values you can retrieve come from the Control Options panel. If \*RBTDFE is entered in a field on the Control Options panel, Robot Schedule automatically retrieves that value from the job's environment.

To change a value, you need to use the UPDJOB operation. The UPDJOB operation creates a temporary job with the changes to the job setup made by your OPAL code.

We provide several special variables that are especially useful with the RTVJOB and UPDJOB operations. These are THISJOB, RBTJOBNBR, and RBTJOBNAME. These are explained after the RTVJOB and UPDJOB operations. In addition, we provide a wide variety of Robot Schedule job variables to work with.

The **RTVJOB** operation retrieves a Robot Schedule job record.

**Operation value:** The 12-digit Robot Schedule job number of the job record to be retrieved or THISJOB to retrieve the current job's job record. If you want to retrieve the record of a group member job, specify the group name, followed by a slash, and the job number (see example, next page).

#### Notes:

- If you are using the RTVJOB operation with ADDMIN and the job being retrieved has:
  - No times, the time is added to the current system time.
  - One run time, the time is added to the retrieved job's time.
  - Multiple times, the time is added to the next run time. For example, if the job has run times of 10:00, 11:30, and 14:15, and the current time is 10:45, the time will be added to the 11:30 time; if the current time is 15:00, the time will be added to the 10:00 run for the next day.
- If you are using the RTVJOB operation with the THISJOB variable, read the section on THISJOB first.

### ■ RTVJOB — Retrieve Job Record

## ■ RTVJOB Example

**Example:** If today is Wednesday, the OPAL code retrieves a Robot Schedule job record. The record is for the instance of job number 000000006126 when it runs as a member of the accounts payable group (APGROUP). If the last run of this job completed normally, the current job is skipped. Otherwise, if the other job is still running, it retrieves the record for the current job, and reschedules the current job to run in 15 minutes.

Logic Operand	Variable	Operation	Operation Values
	USRFLD1	CHGTO	GROUP
IF	DAY	EQ	3
		RTVJOB	APGROUP/000000006126
IF	RBTJOBNBR	EQ	C
		SKIP	
ELSE			
IF	RBTJOBNBR	EQ	R
		RTVJOB	USRFLD1/THISJOB
		ADMIN	15
END			
END			
END			

**■ UPDJOB —  
Update Job**

The first time the **UPDJOB** operation is processed for a job, it creates a temporary job that includes the changes that resulted from processing the OPAL code. Even if your code has multiple UPDJOB statements, only one temporary job is created. The temporary job has a schedule override code of NEXT, which means the next run time. The original job is placed in a suspended state and cannot be selected on the Job Schedule List until the temporary job is processed. If your OPAL code is likely to have produced changes in the job, perform UPDJOB operation before performing a RTVJOB operation.

When the temporary job is processed, the OPAL code may produce additional changes. In this case, the temporary job is simply updated. When the temporary job completes (either normally or abnormally), the job completion history is recorded under the regular Robot Schedule job and the temporary job disappears from the schedule.

**Note:** It is easy to spot OPAL temporary jobs on the Job Schedule List because they have a prefix of OP in the Description field. You may also see them on the Group Membership panel. OPAL temporary jobs are NOT listed on the Robot Schedule Job Finder.

**Note:** if you are using the UPDJOB operation with RTVJOB and the retrieved job has:

- No times, the current system time is used and the job is put on hold.
- One run time, the temporary job is created with a schedule override code of NEXT and the run time for the original job.
- Multiple times, the run time closest to the current system time is used.

**Operation Value:** None.

**Example:** The following OPAL code creates a temporary job that changes the job's output queue from DKJQ to PRT01. It then retrieves job completion information about another job. If the retrieved job completed normally, the temporary job is skipped.

Logic Operand	Variable	Operation	Operation Values
IF	OUTQ	EQ	DKJQ
	OUTQ	CHGTO	PRT01
		UPDJOB	
END			
		RTVJOB	APGROUP/000000006126
IF	RBTJOBNBR	EQ	C
		SKIP	
END			

### ■ Prerequisite Job Variables

Robot Schedule provides several prerequisite job variables for you to use. These variables are especially useful for doing complex tests of job prerequisites that cannot be accomplished by using the regular Reactive Job Setup panel.

OPAL in Robot Schedule lets you use the actual Robot Schedule job number. However, to make your code useful to multiple jobs, it also allows you to use the variable THISJOB or the variable RBTJOBNBR to refer to the Robot Schedule job number. You can also use the variable RBTJOBNAME to refer to the Robot Schedule job name. None of these special variables (THISJOB, RBTJOBNBR, or RBTJOBNAME) can be changed using the CHGTO keyword.

When these variables are used in the Variable field, the value of these variables depends what is entered in the Operation Values field, as follows:

**Completion Code:** If the value in the Operation Values field is a single character, it is assumed to be the last job completion code. Valid completion codes are C, T, E, R, S, K, D, P, and U. These codes have the following meanings:

- C** Completed normally
- T** Terminated abnormally
- E** Completed either normally or abnormally
- R** Running
- S** Submitted, but not yet running
- K** Skipped by OPAL
- D** Delayed by OPAL
- P** Pending
- U** Delete user job messages

**Completion Time:** If the value in the Operation Values field is four digits in the range 0000 to 2400, it is assumed to be the last job completion time.

**Completion Date:** If the value in the Operation Values field is six digits in system date format, it is assumed to be the last job completion date.

**Robot Schedule Job Number:** If the value in the Operations Values field is 12 digits, it is assumed to be a Robot Schedule job number.

■ **THISJOB — The Current Job**

The **THISJOB** variable always refers to the current job and can be used anywhere the Robot Schedule job number could be used.

**Value:** Depends on the corresponding entry in the Operation Values field as explained on the previous page.

**Example:** The following condition is true if the current job last completed after 12:15 p.m.

Logic Operand	Variable	Operation	Operation Values
IF	THISJOB	GT	1215

**Notes:**

- This keyword cannot be used with the **CHGTO** operation.
- If using a **RTVJOB** operation and **THISJOB** variable to access the group member part of a job, you must qualify the **THISJOB** variable with the group name.

Logic Operand	Variable	Operation	Operation Values
	USRFLD1	CHGTO RTVJOB	GROUP USRFLD1/THISJOB

■ **RBTJOBNBR — Robot Schedule Job Number**

The **RBTJOBNBR** variable can be used anywhere the Robot Schedule job number could be used. When your **OPAL** code begins executing, it refers to the current job and thus has the same value as **THISJOB**. However, if you retrieve information about a different job using the **RTVJOB** operation, the **RBTJOBNBR** variable will have a value from the retrieved job.

**Value:** Depends on the corresponding entry in the Operation Values field as explained on the previous page.

**Note:** This variable cannot be changed with the **CHGTO** keyword.

■ **RBTJOBNAME — Robot Schedule Job Name**

The **RBTJOBNAME** variable contains the name of the Robot Schedule job.

**Value:** Robot Schedule job name (up to 10 characters).

**Note:** This variable cannot be changed with the **CHGTO** keyword.

## ■ Robot Schedule Job Variables Table

Robot Schedule lets you use many different aspects of the Robot Schedule job as variables in your OPAL code. Here is a quick reference chart; the variables are explained in detail in the Robot Schedule User Guide.

Job Variable Name	Description	Size and Type or Special Values	Use with CHGTO Operation?
ACCTCODE	Accounting code	Up to 15 characters	Yes
ATPOOL	Auto Tune pool size for job	Up to 6 digits	Yes
CALENDAR	Calendar name	Up to 10 characters	No
CURLIB	Current library name	Up to 10 characters	Yes
DATEOBJ	Date object name	Up to 10 characters	No
ENDRANGE	End time of time range	4 digits in 24-hour clock notation	Yes
GROUP	Group name	Up to 10 characters	No
JOBDESC	Job description	Up to 10 characters	Yes
JOBDLIB	Job description library name	Up to 10 characters	Yes
JOBENV	Job environment name	Up to 10 characters	Yes
JOBQ	Job queue name	Up to 10 characters	Yes
JOBQLIB	Job queue library name	Up to 10 characters	Yes
LIBLNAME	Library list name	Up to 10 characters	Yes
MAXRUN	Maximum run duration in minutes	Up to 4 digits. 0 = no time limit. Max = 1439.	Yes
MAXRUNACT	Action to take at maximum run time	*ENDJOB, *WARN, or *PAGE	Yes
MSGQ	User message queue for job completion messages	Up to 10 characters	Yes
MSGQLIB	User message queue library name	Up to 10 characters	Yes
MSGRPY	How should Robot respond to messages?	*DFT, *SYSRPYL, *JOBDESC, or *RQD	Yes
NONWORK	What should Robot do if the job is scheduled to run on a non-working day?	Y, N, F, or B	Yes
OPALNAME	OPAL object name	Up to 10 characters	No
OUTQ	Output queue name	Up to 10 characters	Yes
OUTQLIB	Output queue library name	Up to 10 characters	Yes
PAGER	Pager name	Up to 10 characters	Yes
SCHEDULE	Schedule override code	D, H, N, or O	No
STARTRANGE	Start time of time range	4 digits in 24-hour clock notation	Yes
TIME	Time the job is scheduled to run	4 digits in 24-hour clock notation	No
USER	User profile	Up to 10 characters	Not to QSECOFR



■ **Reading or Changing Job Environment**

OPAL’s environment operations allow you to retrieve information about any environment and to update the environment. Changes to an environment are **permanent**, unlike job record changes, which are temporary.

When you attach OPAL code to a job, an implied RTVENV operation is performed for the current job. As long as you are working with the current job, you do not need to code an explicit RTVENV operation. However, if you retrieve information about a different environment and then want to read or change information about the current environment, you need a RTVENV statement to get the values for the current environment again.

To change a value, you need to use the UPDENV operation. The UPDENV operation changes the environment permanently.

We provide several special variables that are especially useful with the RTVENV and UPDENV operations. These are THISJOB, RBTJOBNBR, and RBTJOBNAME. These are explained earlier under Reading or Changing Job Setup. In addition, we provide a wide variety of Environment variables to work with.

■ **RTVENV — Retrieve Environment**

The **RTVENV** operation retrieves a Robot Schedule job’s environment and fills the job environment variables.

**Operation value:** The name of the environment to be retrieved (up to 10 characters) or THISJOB to retrieve the current job’s environment.

**Example:** The following OPAL code retrieves the PAYROLL environment and makes sure that the current job’s job queue is the same as the job queue specified in the PAYROLL environment.

Logic Operand	Variable	Operation	Operation Values
		RTVENV	PAYROLL
IF	EJOBQ	NE	JOBQ
	JOBQ	CHGTO	EJOBQ
		UPDJOB	
END			

## ■ UPDENV — Update Environment

The **UPDENV** operation updates the environment permanently to reflect the changes made by the OPAL program. If your OPAL code is likely to have changed the environment for the current job, you should use the UPDENV operation before doing a RTVENV operation.

**Operation Value:** None.

**Example:** The following OPAL code changes the current job's environment output queue from PRT02 to PRT01 and updates the environment.

Logic Operand	Variable	Operation	Operation Values
IF	EOUTQ	EQ	PRT02
AND	ACTDEV	NE	EOUTQ
	EOUTQ	CHGTO	PRT01
		UPDENV	
END			

**Tip:** You can use the UPDENV operation with the CHGTO operation to make permanent changes to an environment.

**■ Environment  
Variables Table**

Robot Schedule provides a set of variables that allow your OPAL code to retrieve information about the job's environment. All the environment variables except for ECALENDAR can also be updated. You can permanently change an environment using OPAL.

All environment variables begin with the letter E. The descriptions of the environment variables are the same as those of the Robot Schedule job variables with the same name.

Environment Variable Name	Description	Size and Type or Special Values	Use with CHGTO Operation?
EACCTCODE	Accounting code	Up to 15 characters	Yes
EATPOOL	Auto Tune pool size for job	Up to 6 digits	Yes
ECALENDAR	Calendar name	Up to 10 characters	No
ECURLIB	Current library name	Up to 10 characters	Yes
EJOB	Job description	Up to 10 characters	Yes
EJOBDLIB	Job description library name	Up to 10 characters	Yes
EJOBQ	Job queue name	Up to 10 characters	Yes
EJOBQLIB	Job queue library name	Up to 10 characters	Yes
ELIBLNAME	Library list name	Up to 10 characters	Yes
EMSGQ	User message queue for job completion messages	Up to 10 characters	Yes
EMSGQLIB	User message queue library name	Up to 10 characters	Yes
EMSGRPY	How should Robot respond to messages?	*DFT, *SYSRPYL, *JOB, or *RQD	Yes
EOUTQ	Output queue name	Up to 10 characters	Yes
EOUTQLIB	Output queue library name	Up to 10 characters	Yes
EPAGER	Pager name	Up to 10 characters	Yes
EUSER	User profile	Up to 10 characters	Not to QSECOFR

## ■ Reading or Changing Data Areas

You can assign user variables to portions of a data area and have OPAL read and update them. The data area operations allow you to retrieve and update data area values. The retrieve data area definition (RTVDTADFN) operation allows you to get values from a data area and put them in your user variables. The change data area definition (CHGDTADFN) operation allows you to update the data area with values from your user variables. For more information on Data Area Definitions, go to the OPAL Menu chapter earlier in this Guide.

## ■ RTVDTADFN — Retrieve Data Area Definition

The **RTVDTADFN** operation allows you retrieve values from your data area to establish initial values for user variables. You must assign user variables to portions of the data area before you can refer to them in your OPAL code. See the Data Area Definition Elements panel.

**Operation Values:** Name of the Data Area Definition (up to 10 characters) to be retrieved.

**Example:** See below.

## ■ CHGDTADFN — Change Data Area Definition

The **CHGDTADFN** operation allows you to update your data area with values from your user variables. You must assign user variables to portions of the data area before you can refer to them in your OPAL code. See the Data Area Definition Elements panel.

**Operation Values:** Name of the Data Area (up to 10 characters) to be changed.

**Example:** You have created a data area definition named PAYROLL. Based on information in the company number field, you want to change the company name field. As the last step, you want to update the values in the data area PAYROLL.

Logic Operand	Variable	Operation	Operation Values
		RTVDTADFN	PAYROLL
IF	USRFLD1	NE	00536
AND	USRFLD1	NE	12984
	USRLONG1	CHGTO	ABC Widgets Inc.
ELSE			
	USRLONG1	CHGTO	XYZ Widgets Mfg.
		CHGDTADFN	PAYROLL
END			

**Tip:** You can use CHGDTADFN with the CHGTO operation to make permanent changes to the data area defined in the data area definition.

### ■ Check Current Status

OPAL lets you check on the current status of iSeries objects, devices, and work management issues. For example, OPAL can check if an object exists on the iSeries, whether the object has any locks, and even if a file has any records in it. It can check if printers, workstations, controllers, and communication lines are up and running. Furthermore, it can check on active jobs, users, and subsystems and retrieve important information about job queues.

### ■ Check Object Status

The Robot Schedule OPAL object status operations allow you to retrieve certain information about any iSeries object. By using these operations, you can prevent your applications from terminating abnormally because transactions have not arrived from another iSeries or an interactive program has a lock on an object.

**Note:** If you are not authorized to the object, or the variable you are checking for is not on the system, the OPAL is skipped and the Robot Schedule job runs without checking the object.

### ■ RTVOBJSTS — Retrieve Object Status

The **RTVOBJSTS** operation allows you to find out if any iSeries object exists on the system. It also allows you to find out if the object is locked (not all objects can be locked) and if there are records in any file. This operation is used with the OBJSTS variable which allows you to check if an object exists, is locked, or contains records (see OBJSTS variable).

**Operation Value:** Lists the object to be retrieved. The general notation for an object is LIBRARY/OBJECT \*TYPE. If you do not specify the library, this value defaults to \*LIBL. If the object is a file, you can also specify the member like this: LIBRARY/FILE(MEMBER) \*TYPE. If you are retrieving a file and do not specify a member, the member value defaults to \*FIRST.

**Example:** The following OPAL code verifies that the output queue PRT01 exists and is active, and then changes the output queue for the job. Then it retrieves information about a file. If the file is empty, the job is skipped.

Logic Operand	Variable	Operation	Operation Values
		RTVOBJSTS	PRT01 *OUTQ
IF	OBJSTS	EQ	EXISTS
AND	ACTDEV	EQ	PRT01
	OUTQ	CHGTO	PRT01
		UPDJOB	
		RTVOBJSTS	DKJLIB/DKJFILE *FILE
IF	OBJSTS	NE	RECORDS
		SKIP	
END			
END			

## ■ OBJSTS — Object Status

The **OBJSTS** variable checks whether or not the specified object exists, is locked, or contains records. It is generally used after performing a retrieve object status (RTVOBJSTS) operation.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if the status in the Operation Values field matches the status of the object; NE is true if the status in the Operation Values field does not match the status of the object.

### Notes:

- This variable cannot be used with the CHGTO operation.
- If you do not have authority to the object, it will be treated as if it does not exist.

**Operation Values:** The following three values—EXISTS, LOCKED, RECORDS.

**Example:** The following condition is true if DKJFILE in DKJLIB exists, contains records, and is not locked.

Logic Operand	Variable	Operation	Operation Values
		RTVOBJSTS	DKJLIB/DKJFILE *FILE
IF	OBJSTS	EQ	EXISTS
AND	OBJSTS	EQ	RECORDS
AND	OBJSTS	NE	LOCKED

## ■ Check Device Status

Use the following OPAL variables to check the status of various devices.

### Notes:

- When using ACTCTL, ACTDEV, or ACTLIN, the following statuses are considered active:

VARY OFF PENDING	ACTIVE READER
VARIED ON	ACTIVE WRITER
SIGNON DISPLAY	HELD
ACTIVE	SYSTEM REQUEST

- Unlike other OPAL variables, these do not represent values and so cannot be used in the Operation Values field.

## ■ ACTCTL — Active Controller

The **ACTCTL** variable checks whether or not the specified controller is active. A controller is a processor that controls one or more I/O devices, such as display stations or tape units.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if the controller is active; NE is true if the controller is not active.

**Operation Values:** The name of the controller to be checked.

**Example:** The following condition is true if the controller CTL01 is not active.

Logic Operand	Variable	Operation	Operation Values
IF	ACTCTL	NE	CTL01

## ■ ACTDEV — Active Device

The **ACTDEV** variable checks whether or not the specified device is active. An iSeries device can be a display station, printer, diskette unit, tape unit, or remote system.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if the device is active; NE is true if the device is not active.

**Operation Values:** The name of the device to be checked.

**Example:** The following condition is true if the printer PRT01 is not active.

Logic Operand	Variable	Operation	Operation Values
IF	ACTDEV	NE	PRT01

## ■ ACTLIN — Active Line

The **ACTLIN** variable checks whether or not the specified communications line is active.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if the line is active; NE is true if the line is not active.

**Operation Values:** The name of the line to be checked.

**Example:** The following condition is true if the line ACH\_LINE is not active.

Logic Operand	Variable	Operation	Operation Values
IF	ACTLIN	NE	ACH_LINE

## ■ ACTRCL — Active Robot/CLIENT

The **ACTRCL** variable checks whether or not Robot/CLIENT is active on the specified PC.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if Robot/CLIENT is active; NE is true if Robot/CLIENT is not active.

**Operation Values:** The Robot/CLIENT name of the PC to be checked.

**Example:** The following condition is true if the Robot/CLIENT PC Dave Johnson is active.

Logic Operand	Variable	Operation	Operation Values
IF	ACTRCL	EQ	Dave Johnson



## ■ DEVSTS — Device Status

The **DEVSTS** variable is used in conjunction with the ACTCTL, ACTDEV, or ACTLIN variables to check specific status of the controller, device, or line.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if the status in the Operation Values field matches the status of the controller, device, or line; NE is true if the status in the Operation Values field does not match the status of the controller, device, or line.

**Operation Values:** Any of the following statuses:

VARIED OFF	RCYPND
VARY OFF PENDING	RCYCNL
VARY ON PENDING	SYSTEM REQUEST
VARIED ON	FAILED
CONNECT PENDING	FAILED READER
SIGNON DISPLAY	FAILED WRITER
ACTIVE	DIAGNOSTIC MODE
ACTIVE READER	DAMAGED
ACTIVE WRITER	LOCKED
HELD	UNKNOWN
POWERED OFF	

**Example:** The following example skips the job if TAP01 is not varied on.

Logic Operand	Variable	Operation	Operation Values
IF	ACTDEV	EQ	TAP01
END			
IF	DEVSTS	NE	VARIED ON
		SNDMSG	SNDMSG 'TAP01 is not ready' QSYSOPR
		SKIP	
		*	Job skipped—no resources available
END			

## ■ Check Work Management Status

### ■ ACTJOB — Active Job

Robot Schedule OPAL also lets you check on the status of various work management objects including jobs, subsystems, users, and job queues. It lets you find out whether or not a job queue is on hold and how many jobs are in a queue.

The **ACTJOB** variable checks whether or not the specified job is active. It first looks at Robot Schedule jobs. If the specified job is a Robot Schedule job with a status of R (running), it comes back as active. If it is not a Robot Schedule job, it checks OS/400 to see if the job has an active status.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if the job is active; NE is true if the job is not active.

**Operation Values:** The name (and optionally the user) of the job to be checked.

**Example:** The following condition is true if the job name and user RBACCTPAY/QPGMR is not active.

Logic Operand	Variable	Operation	Operation Values
IF	ACTJOB	NE	RBACCTPAY/QPGMR

### ■ ACTSBS — Active Subsystem

The **ACTSBS** variable checks whether or not the specified subsystem is active.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if the subsystem is active; NE is true if the subsystem is not active.

**Operation Values:** The name of the subsystem to be checked.

**Example:** The following condition is true if the subsystem is active.

Logic Operand	Variable	Operation	Operation Values
IF	ACTSBS	EQ	QBATCH

## ■ ACTUSR — Active User

The ACTUSR variable checks whether or not the specified user is active.

**Operation:** The only valid comparisons are EQ and NE. EQ is true if the user is active; NE is true if the user is not active.

**Operation Values:** The name of the user profile to be checked.

**Example:** The following condition is true if DAVEJ is not signed on.

Logic Operand	Variable	Operation	Operation Values
IF	ACTUSR	NE	DAVEJ

## ■ RTVJOBQ — Retrieve Job Queue Status

The **RTVJOBQ** operation is used to retrieve the status of a job queue. You can check if a job queue is on hold and also find out how many jobs are in the queue. It is used in conjunction with the job queue variables JOBQCNT and JOBQHOLD.

**Operation Values:** The library and job queue to be retrieved or THISJOB. If no library is specified, the library value defaults to \*LIBL.

**Example:** The following OPAL code checks if the job queue QBATCH is on hold. If it is, it checks QPGMR. If both are on hold, it uses the job queue listed for the job on its Control Options panel. If \*RBTDFE is listed on the Control Options panel, it uses the queue listed in the job's Environment.

Logic Operand	Variable	Operation	Operation Values
		RTVJOBQ	QBATCH
IF	JOBQHOLD	EQ	NO
	JOBQ	CHGTO	QBATCH
ELSE			
		RTVJOBQ	QPGMR
IF	JOBQHOLD	EQ	NO
	JOBQ	CHGTO	QPGMR
END			
END			
		UPDJOB	

## ■ JOBQCNT — Number of Jobs in Job Queue

The **JOBQCNT** variable allows you to check how busy a job queue is and then change the queue if it is too busy.

**Operation Value:** A number that the number of jobs in the queue can be compared to.

**Example:** The following code looks at the QBATCH job queue. If there are fewer than three jobs in the queue, it changes the current job's job queue to QBATCH. If there are three or more jobs in the QBATCH job queue, it checks the QPGMR queue. If there are fewer than five jobs in the QPGMR queue, it changes the job queue to QPGMR. If neither of these conditions is true, it will use the job queue listed on the job's Control Options panel (or if the value on the Control Options panel is \*RBTDFT it uses the job queue listed in the job's Environment).

Logic Operand	Variable	Operation	Operation Values
		RTVJOBQ	QBATCH
IF	JOBQCNT	LT	3
	JOBQ	CHGTO	QBATCH
ELSE			
		RTVJOBQ	QPGMR
IF	JOBQCNT	LT	5
	JOBQ	CHGTO	QPGMR
END			
END			
		UPDJOB	

## ■ JOBQHOLD — Is Job Queue on Hold?

The **JOBQHOLD** variable lets you find out if a job queue is on hold and avoid sending jobs to queues that are on hold.

**Value:** YES means the job queue is on hold; NO means it is not on hold.

**Example:** The following OPAL code checks if the job queue QBATCH is on hold. If it is, it checks QPGMR. If both are on hold, it uses the job queue listed for the job on its Control Options panel. If \*RBTDFE is listed on the Control Options panel, it uses the queue listed in the job's Environment.

Logic Operand	Variable	Operation	Operation Values
		RTVJOBQ	QBATCH
IF	JOBQHOLD	EQ	NO
	JOBQ	CHGTO	QBATCH
ELSE			
		RTVJOBQ	QPGMR
IF	JOBQHOLD	EQ	NO
	JOBQ	CHGTO	QPGMR
END			
END			
		UPDJOB	

## ■ Changing User Variable Values

### ■ CHGTO — Change the Value of a Variable

Use these operations to change the values of OPAL variables, especially the user variables. The user variables can be used to hold operation values to be used later.

Use the **CHGTO** operation to assign a value to a user variable. Enter the name of the user variable (USRFLDn, USRFLGn, USRNBRn, USRLONGn or USRCNTn, where n is a value from 1 to 5) in the variable field. The CHGTO operation can also be used with certain other variables. For example, to use the CHGTO operation to change a job, you must use it with the UPDJOB operation.

**Operation value:** Value to be assigned to the user variable. Specify the following: for USRFLDn, up to ten characters; for USRFLGn, one character; for USRNBRn, a numeric value (15, 5); for USRLONGn, up to 45 characters; for USRCNTn, a numeric value (5, 0).

**Notes:**

- The CHGTO operation has many restrictions when used with variables other than user variables in Robot Schedule OPAL. If you enter the CHGTO operation and a value in the Operations Value field that does not exist on the system where you are entering the OPAL code, you will get a warning message. We allow you to enter “invalid” values so you can create code on one system and use it on another. These values are re-checked at run time for validity on the system where the code is running and for proper authorization (security). These checks assure that your Robot Schedule job records will NOT be updated with invalid values. A warning message is sent to the Job Completion History indicating the value was not changed. See the Robot Schedule Job Variables Table and the Environment Variables Table to see the restrictions.
- If the OPAL header is set up to use ‘After’ for when OPAL should run and the OPAL code has a CHGTO operation after an ADDMIN operation, changes to environments occur, but the UPDJOB does not pick them up.

**Example:** The following statement changes the value of USRFLD1 to Total.

Logic Operand	Variable	Operation	Operation Values
	USRFLD1	CHGTO	Total

### ■ ADD — Add to a Numeric User Variable

Use the **ADD** operation to add to the value of one of the numeric user variables specified in the variable field (USRNBRn or USRCNTn, where n is a value from 1 to 5).

**Operation value:** Number to be added to the numeric value already in the variable. For USRNBRn, specify a numeric value in the format (15, 5); for USRCNTn specify a numeric value in the format (5, 0). The number to be added can also be specified by another numeric user variable.

**Example:** The following statement adds 999.99 to the value in USRNBR1.

Logic Operand	Variable	Operation	Operation Values
	USRNBR1	ADD	999.99

**Example:** The following statement adds the value in the USRCNT1 variable to USRCNT2.

Logic Operand	Variable	Operation	Operation Values
	USRCNT2	ADD	USRCNT1

**Example:** The following statement retrieves the value found in a Robot Schedule reserved variable and adds it to the value in USRNBR1. This works only when the Robot Schedule reserved variable returns a numeric value.

Logic Operand	Variable	Operation	Operation Values
	USRNBR1	ADD	@@NUMVAL

■ **SUB — Subtract from a Numeric User Variable**

Use the **SUB** operation to subtract from the numeric user variable specified in the variable field.

**Operation value:** Number to be subtracted from the numeric value already in the USRNBRn or USRCNTn variable. The number to be subtracted can also be specified by another numeric user variable.

**Example:** The following statement subtracts 1 from the value in the USRNBR1 variable.

Logic Operand	Variable	Operation	Operation Values
	USRNBR1	SUB	1

## ■ Concatenate Information

## ■ CAT — Concatenate Values

OPAL gives you the ability to concatenate information from several fields into a single variable.

The **CAT** operation concatenates user variable values.

**Operation value:** USRFLDn or USRLONGn.

**Example:** The following statement concatenates the value in USRLONG5 to the value (if any) already in USRLONG1. The CAT operation does not add a space between the concatenated values.

Logic Operand	Variable	Operation	Operation Values
	USRLONG1	CAT	USRLONG5

## ■ BCAT — Insert a Blank and Concatenate

The **BCAT** operation works the same way as CAT but inserts a blank between the values it is concatenating.

**Operation value:** USRFLDn or USRLONGn.

**Example:** The following code puts the name of the Robot Schedule job into the user variable USRLONG1 and then adds a blank and the words ‘was changed.’ It then sends USRLONG1 as a message to QSYSOPR.

Logic Operand	Variable	Operation	Operation Values
	USRLONG1	CHGTO	RBTJOBNAME
	USRLONG1	BCAT	was changed
		SNDMSG	USRLONG1 QSYSOPR



■ **Send Messages**

■ **SENDMSG —  
Send an iSeries  
Message**

Use the following operations to send a message to an iSeries message queue, a Robot Console message center, or a Robot Alert pager.

The **SENDMSG** operation allows you to send a standard iSeries message to a user.

**Operation value:** If your message is brief, you can enter it in single quotes, followed by a space and the user profile name to which the message should be sent (see example, below). You can also position your cursor in the Operation Values field and press **function key 4** to get the standard IBM SENDMSG prompt screen.

**Example:** The following statement sends the message ‘Load checks’ to the user profile SYSOPR.

Logic Operand	Variable	Operation	Operation Values
		SENDMSG	‘Load checks’ SYSOPR

■ **SNDBRKMSG —  
Send an iSeries  
Break Message**

The **SNDBRKMSG** operation allows you to send a iSeries break message to one or more devices.

**Operation value:** If your message is brief, you can enter it in single quotes, followed by a space and the device name to which the message should be sent (see example, below). You can also position your cursor in the Operation Values field and press **function key 4** to get the standard IBM SNDBRKMSG prompt screen.

**Example:** The following statement sends a break message to all workstations.

Logic Operand	Variable	Operation	Operation Values
		SNDBRKMSG	‘Get off the system’ *ALLWS

## ■ SENDMC — Send a Message to a Message Center

If you have Robot Console installed, use the **SENDMC** operation to send a message to a Robot Console message center.

**Operation value:** Position the cursor in the Operation Values field and press **function key 4**. A prompt screen appears, as shown below. Enter the message center name (up to 10 characters) and the message text to be sent. You can also require a response to the message.

**Example:** The following statement sends a message to a Robot Console message center.

Logic Operand	Variable	Operation	Operation Values
		SENDMC	Prompt to see command

Enter the name of the message center that should receive the message.

```

Send Message to Message Center (SENDMC)

Type choices, press Enter.

Message Center Name . . . . . JEFF      Name
Message Text . . . . . Robot/SCHEDULE job waiting for tape mount on
TAP01

Response Required? . . . . . *NO      *NO, *YES

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

Enter the message text.

Enter \*YES to require a response to the message; enter \*NO if no response is required.

■ **PAGE — Send a Pager Message**

If you have Robot Alert installed, the **PAGE** operation executes the Robot Alert command that sends a pager message. For more information, see the Robot Alert User Guide.

**Operation value:** Position the cursor in the Operation Value field and press **function key 4**. If you have Robot Alert installed, the prompt screen for the RBASNDMSG command appears. Enter a pager message to be sent by your Robot Schedule job to the specified pager or broadcast list.

**Example:** The following statement sends a pager message.

Logic Operand	Variable	Operation	Operation Values
---------------	----------	-----------	------------------

PAGE

Prompt to see command

Enter the pager message you want to send. The message length is limited to the number of characters your pager vendor will allow.

```

Send a Message to a Pager (RBASNDMSG)

Type choices, press Enter.

Message Text . . . . . > 'ACH Line not active. Vary it on.'
_____
_____
_____
_____

Pager ID or Broadcast List . . . . . > OPERATIONS      Character value
Response Required . . . . . *YES_             *SETUP, *YES, *NO
Truncate Message for Vendor . . . . . *NO_           *YES, *NO
Available Responses . . . . . _____
+ for more values
Restrict Responses . . . . . *NO_             *YES, *NO
Response ID . . . . . _____      Character value
Response Program . . . . . _____      Name
Library . . . . . _____          Name, *LIBL
More...

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
    
```

Enter the pager ID or broadcast list name to which you want to send the pager message. The pager ID or broadcast list must be in the pager directory.

## ■ QUIT — Stop All Processing

The **QUIT** operation stops all processing of the OPAL code. If you have a long, complicated OPAL program, you may want to place a QUIT operation strategically. The QUIT operation functions like a go to end of program statement.

**Operation value:** None

**Example:** The following statement stops all processing of the OPAL Code.

Logic Operand	Variable	Operation	Operation Values
------------------	----------	-----------	---------------------

QUIT

■ **Changing the Next Statement Processed**

■ **TAG — Tags a Program Location**

Use the TAG operation to assign a name to a location in the OPAL program. Use a GOTO operation to continue processing at the tagged location. The \* (asterisk) operation lets you add comments to the program.

The **TAG** operation assigns a name to a location in the OPAL program. A GOTO operation then specifies the tag name to go to the tagged location.

**Operation value:** Name for the location.

**Example:** The following statement assigns the name BEGIN to this location in the OPAL code.

Logic Operand	Variable	Operation	Operation Values
		TAG	BEGIN

■ **GOTO — Go to a Tagged Location**

The **GOTO** operation transfers processing to the location specified by the tag name. When a GOTO is performed, processing immediately jumps to the tagged location. Thus, the next statement processed is the statement that follows the TAG operation.

**Operation value:** Name specified on a TAG operation in the code.

**Example:** The following statement continues processing at the TAG statement that contains the name BEGIN.

Logic Operand	Variable	Operation	Operation Values
		GOTO	BEGIN

\* — **Adds a Comment**

The \* (asterisk) operation indicates that the statement is a comment used to document the processing performed by the OPAL program. You can enter text in the Operation Value field to describe what the code does, or you can leave the field blank to improve the readability of the code.

**Note:** A comment statement cannot appear between IF and END statements.

**Operation value:** Any characters.

**Example:** The following statement inserts a blank line in the code for readability.

Logic Operand	Variable	Operation	Operation Values
		*	

## ■ SNDRBTDTA — Run a Reactive Job

The **SNDRBTDTA** operation sends a job completion status to a reactive job. When all prerequisites for the job have been met, the reactive job can run. For more information about reactive jobs, go to the reactive jobs section of the Robot Schedule User Guide.

**Operation value:** Position the cursor in the Operation Values field and press **function key 4**. The prompt screen for the SNDRBTDTA command appears. Enter the job name, completion status, and system name for a user job that has already been entered in the prerequisite list of a reactive job.

**Example:** The following statement sends a completion code of C for the specified user job (SUPPORTREQ) to the reactive job's prerequisite list.

Logic Operand	Variable	Operation	Operation Values
---------------	----------	-----------	------------------

SNDRBTDTA

Prompt to see command

Enter the name of a user job that has already been entered in a Robot Schedule prerequisite list.

Send ROBOT Reactive Data (SNDRBTDTA)

Type choices, press Enter.

Prerequisite user job name . . . .	SUPPORTREQ	Character value
OR Prereq ROBOT job number . . .		Character value
Completion status code . . . . .	C	B, C, D, K, P, R, S, T
System Name . . . . .	DALLAS	Character value

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys

Enter the job completion status that Robot Schedule should enter in the prerequisite list for the user job.

Enter the name of the system on which the prerequisite job resides. If the job resides on the local system, you can leave the field blank.

■ **Variables**

In an OPAL statement you use the variable field for either of the following:

- To specify the variable to be tested by an IF, OR, or AND statement.
- To specify the variable to be changed by a CHGTO, ADD, SUB, INTABLE, or NOTINTABLE operation.

For all other statements, the field is left blank. The CHGTO, ADD, SUB, INTABLE, or NOTINTABLE operations are described earlier under Operations.

■ **Conditions**

To specify a condition on an IF, AND, or OR statement, you fill in three fields:

<b><u>Field</u></b>	<b><u>Value</u></b>
Variable	The variable whose value is to be tested
Operation	A comparison operation
Operation Values	An operation value (either the actual value or a variable)

When the condition is tested, the values of the Variable and Operation Values fields are compared. If the two values have the relationship specified by the comparison operation, the condition is true. Otherwise, the condition is false.

Example: Consider the following IF condition.

Logic Operand	Variable	Operation	Operation Values
IF	SYSTIME	GT	1700

SYSTIME is a variable containing the current system time; the comparison operator is GT (greater than); and the operation value is 1700 (5 p.m. expressed in the hours-minutes format using a 24-hour clock). When the condition is tested, the current system time is compared with the value 1700. If the time is greater than 1700, the condition is true; otherwise, it is false.

## ■ Variable Field

Variables contain values that are obtained from the Robot Schedule job being processed or from the system where the job is being run. In most cases, the variable receives its value when Robot Schedule begins processing the job. You can use a named variable in the Variable field, in the Operation Values field, or as a command parameter in the Operation Values field.

For example, all of the following statements use the variable USER:

Logic Operand	Variable	Operation	Operation Values
IF	USER	EQ	BOB
	USRFLD1	CHGTO	USER
		SENDMSG	'Your job was skipped'
			TOUSR(USER)



## ■ Use the Robot Schedule Calendar

You can reference the characteristics of a date as defined by the Robot Schedule calendar for a job using OPAL. These variables always refer to the calendar for the current job, even if you retrieve information about another job. In addition, these variables are based on the job's start date. This can be important if you are working with a group of jobs that use a common start date, regardless of when they actually run.

## ■ WORKDAY— Is it a Working Day?

The **WORKDAY** variable contains a true value (YES) if today is a working day as defined by the Robot Schedule calendar currently in effect. Otherwise, it contains a NO value.

**Value:** YES or NO (or Y or N).

**Example:** The following condition is true if today is not a working day.

Logic Operand	Variable	Operation	Operation Values
IF	WORKDAY	EQ	NO

## ■ DAY—Day of the Week

The **DAY** variable gets the number (1–7) of the current day of the week (Monday, Tuesday, and so on).

**Value:** Number where Monday is 1, Tuesday is 2, and so forth up to 7 for Sunday.

**Example:** The following condition is true if today is a Friday.

Logic Operand	Variable	Operation	Operation Values
IF	DAY	EQ	5

## ■ DAYMTH— Day Number in the Month

The **DAYMTH** variable contains the day number of today within the current month as defined by the Robot Schedule calendar currently in effect.

**Value:** One- or two-digit day number counting from the beginning of the month. The month-ends are defined in the Robot Schedule calendar used. For example, if a fiscal month ends August 28, the day number for August 29 is 1.

## ■ LASTDAY — Last Date in the Month

**Example:** The following condition is true if today is after the tenth day of the month.

Logic Operand	Variable	Operation	Operation Values
IF	DAYMTH	GT	10

The **LASTDAY** variable contains the date of the last day of the current month as defined by the Robot Schedule calendar currently in effect.

**Value:** Six-digit date.

**Note:** If you compare the **LASTDAY** variable with a date in a Robot Schedule reserved command variable, make sure that the date in the Robot Schedule variable is in yymmdd (year, month, day) format.

**Example:** The following condition is true if today is the last day of the month.

Logic Operand	Variable	Operation	Operation Values
IF	LASTDAY	EQ	SYSDATE

## ■ WEEKNO — Week Number

The **WEEKNO** variable contains the week number of the job's start date within the current month. The first seven days of the month are week 1, the next seven days are week 2, and so forth. The previous month end is defined by the calendar. For example, if a fiscal month ends August 28, the week number for August 29 is 1.

**Value:** One-digit number where 1 is the first week in the month, 2 is the second, and so forth.

**Example:** The following condition is true if today is after the first week of the month.

Logic Operand	Variable	Operation	Operation Values
IF	WEEKNO	GT	1

■ **Assign Values to User Variables**

OPAL also provides user variables that do not contain a value until you assign them one. You can assign a value to a user variable using the CHGTO operation and later reference the variable in a condition or operation. Once a user variable has a value assigned to it, it can be compared to constants or to other variables.

**Note:** When you use a user variable in the Operation Values field, make sure you enter the variable name in upper-case format. In addition, you should make sure that you compare variables of the same type. For example, compare a numeric variable with another numeric variable, or a character variable with another character variable.

■ **USRFLD $n$  — Character Variables**

The **USRFLD $n$**  variables are five user variables that can contain up to ten characters each. The first variable is referenced as USRFLD1, the second as USRFLD2, and so forth, up to USRFLD5.

**Value:** Up to ten characters.

**Example:** The following condition is true if USRFLD1 contains the characters TOTAL.

Logic Operand	Variable	Operation	Operation Values
IF	USRFLD1	EQ	TOTAL

### ■ USRLONG $n$ — Longer Character Variables

The **USRLONG $n$**  variables are five user variables that are used like the **USRFLD $n$**  variables, except they can accommodate longer character strings. They can contain up to 45 characters each, although only 35 characters can be entered in the Operation Values field. The first variable is referenced as **USRLONG1**, the second as **USRLONG2**, and so forth, up to **USRLONG5**. These variables may be especially useful with data area definition elements and in conjunction with the concatenation operations **CAT** and **BCAT**.

**Value:** Up to 45 characters.

**Example:** The following condition is true if **USRLONG1** equals the characters **Department 23**.

Logic Operand	Variable	Operation	Operation Values
IF	USRLONG1	EQ	Department 23

### ■ USRFLG $n$ — Flag Variables

The **USRFLG $n$**  variables are five user variables that can contain one character each. The first field is referenced as **USRFLG1**, the second as **USRFLG2**, and so forth, up to **USRFLG5**.

**Value:** One character.

**Example:** The following condition is true if **USRFLG1** contains the character **Y**.

Logic Operand	Variable	Operation	Operation Values
IF	USRFLG1	EQ	Y

■ **USRNBR $n$  — Floating Point Variables**

The **USRNBR $n$**  variables are five user variables that can contain a number in the format (15,5). The first field is referenced as USRNBR1, the second as USRNBR2, and so forth, up to USRNBR5.

**Value:** Number in format (15,5).

**Example:** The following is true if USRNBR1 is greater than 9999.99.

Logic Operand	Variable	Operation	Operation Values
IF	USRNBR1	GT	9999.99

■ **USRCNT $n$  — Numeric Variables**

The **USRCNT $n$**  variables are five user variables that can contain a number in the format (5,0). The first field is referenced as USRCNT1, the second as USRCNT2, and so forth, up to USRCNT5.

**Value:** Number in format (5,0).

**Example:** The following condition is true if USRCNT1 is less than 7.

Logic Operand	Variable	Operation	Operation Values
IF	USRCNT1	LT	7

**Example:** The following condition is true if USRCNT1 is greater than USRCNT2.

Logic Operand	Variable	Operation	Operation Values
IF	USRCNT1	GT	USRCNT2

## ■ Use System Information

Use the following OPAL variables to refer to the iSeries job name, job number, job user, system name, system time, or system date.

**Note:** Because this information is not available until the job is executed, these variables are not useful for OPAL that is run before the job is submitted (with the exception of SYSTEM).

## ■ **JOBNAME** — Job Name

The **JOBNAME** variable contains the name of the current iSeries job.

**Value:** Job name (up to 10 characters).

**Example:** The following condition is true if the job name is RBACCTPAY (Accounts payable).

Logic Operand	Variable	Operation	Operation Values
IF	JOBNAME	EQ	RBACCTPAY

## ■ **JOBNBR** — Job Number

The **JOBNBR** variable contains the iSeries job number.

**Value:** Job number (6 numeric characters).

**Example:** The following statement concatenates the iSeries job number with the information already stored in the user variable USRLONG1.

Logic Operand	Variable	Operation	Operation Values
	USRLONG1	CAT	JOBNBR

## ■ **JOBUSER** — Job User

The **JOBUSER** variable contains the name of the current iSeries job user.

**Value:** User profile name (up to 10 characters).

**Example:** The following condition is true if the job user is Johnson.

Logic Operand	Variable	Operation	Operation Values
IF	JOBUSER	EQ	JOHNSON

## ■ SBMTIME — Time Job was Submitted

The **SBMTIME** variable contains the time this job was submitted.

**Value:** Hours and minutes as they would be displayed on a 24-hour clock (four digits).

**Note:** This keyword cannot be used with the **CHGTO** operation.

**Example:** The following condition is true if the time this job was submitted was 6 p.m.

Logic Operand	Variable	Operation	Operation Values
IF	SBMTIME	EQ	1800

## ■ SYSTEM — System Name

The **SYSTEM** variable gets the system name when the statement is processed.

**Value:** System name (up to 8 characters).

**Note:** This keyword cannot be used with the **CHGTO** operation.

**Example:** The following condition is true if the system name is D10.

Logic Operand	Variable	Operation	Operation Values
IF	SYSTEM	EQ	D10

## ■ SYSTIME — System Time

The **SYSTIME** variable gets the current time from the system clock when the statement is processed.

**Value:** Four-digit number representing the time as hours and minutes on a 24-hour clock. For example, 1230 is exactly one half-hour after noon.

**Note:** This keyword cannot be used with the **CHGTO** operation.

**Example:** The first condition is true if the current time is after 5 p.m., but before midnight. The second condition is true if the current time is after midnight, but before 8 a.m.

Logic Operand	Variable	Operation	Operation Values
IF	SYSTIME	GT	1700
OR	SYSTIME	LT	0800
		PAGE	OPERATOR
END			

## ■ SYSDATE — System Date

The **SYSDATE** variable gets the current system date when the statement is processed.

**Value:** Six-digit number representing the date in the system date format on your system. If the system date format is month, day, year (mmddy), April 15, 1996 is 041596.

**Note:** This keyword cannot be used with the **CHGTO** operation.

**Example:** The following condition is true if the current date is before July 1, 1996, and the system date format is mmddy.

Logic Operand	Variable	Operation	Operation Values
IF	SYSDATE	LT	070196



■ **OPAL Examples**

■ **Page Operator  
If Job Will Be  
Skipped**

This section provides examples of OPAL code that you could use with your Robot Schedule jobs. These are examples only. Review the code carefully; you may need to modify it for use on your system.

This example shows checking for a device condition that should cause the job to be skipped.

Condition: If TAP01 is not ready, skip the job.

If the job will be skipped—in other words, if SKIPSTATUS=Y, the object uses Robot Alert to page the operator.

Logic	Operand	Variable	Operation	Operation Values	Seq
	IF	ACTDEV	EQ	TAP01	10
			SNDMSG	MSG('TAP01 is NOT ready') TOUSR(...	20
			SKIP		30
			*	Job skipped-no resources available	40
	END				50
	IF	SKIPSTATUS	EQ	YES	60
			PAGE	** Prompt to see command **	80
	END				90

**Send a Message to a Pager (RBASNDMSG)**

Type choices, press Enter.

Message Text . . . . . Backups skipped.

---



---



---



---

Pager ID, or Broadcast List . . . OPERATOR      Character value . . . . .

Response Required . . . . . \*NO              \*SETUP, \*YES, \*NO

Truncate Message for Vendor . . . \*NO              \*YES, \*NO

**Bottom**

F3=Exit    F4=Prompt    F5=Refresh    F10=Additional parameters    F12=Cancel

F13=How to use this display    F24=More keys

## ■ Delay Job If a Certain User Is Active

If someone is signed on to the iSeries as QSYSOPR, send a message to Michelle telling her the job will be delayed for 15 minutes. Notice that you can store message text in the user variable USRLONG (up to 45 characters). Also notice the use of the BCAT operation to concatenate two parts of a message with a blank in between.

Logic Operand	Variable	Operation	Operation Values	Seq
IF	ACTUSR	EQ	QSYSOPR	10
	USRLONG3	CHGTO	Someone is on the system!	20
	USRLONG3	BCAT	Wait 15 minutes!	30
		SNDSMSG	MSG(USRLONG3) TOUSR(MICHELLE)	40
		ADMIN	15	50
END				60
		QUIT		70

## ■ Prevent Job From Running Before a Certain Time

This OPAL code will prevent the job from starting before 11:30 p.m. If the Robot Schedule reserved variable @@TIME (which retrieves the current system time) is less than 2330, it delays the job 30 minutes.

Logic Operand	Variable	Operation	Operation Values	Seq
IF	@@TIME	LT	2330	10
		ADMIN	30	20
END				30
		QUIT		40

## ■ Change Job Setup

This example retrieves information about a specific job. If the current time (SYSTIME) is before 4 p.m. and the time the job can start (STARTRANGE) is less than or equal to 3:30 p.m., the OPAL changes STARTRANGE to 4 p.m. Otherwise, it skips the job.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTVJOB	000000005994	10
IF	SYSTIME	LT	1600	20
AND	STARTRANGE	LE	1530	30
	STARTRANGE	CHGTO	1600	40
		UPDJOB		50
ELSE				60
		SKIP		70
END				80

■ **Change Job Environment**

This example retrieves the PURCHASING environment. If it is a weekend, printer PRT01 will be used. Otherwise, printer PRT05 will be used.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUENV	PURCHASING	10
IF	DAY	GE	6	20
	EOUTQ	CHGTO	PRT01	30
ELSE				40
	EOUTQ	CHGTO	PRT05	50
END				60
		UPDENV		70

■ **Change Page Based on Time of Day**

You might want to include code like this in all your Robot Schedule jobs that use ROBOT's STANDARD environment and the PAGE operation. If it is after 6 p.m. and before 7 a.m., it pages the person listed in DP.ONCALL. Otherwise, it pages QSYSOPR.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUENV	STANDARD	10
IF	SYSTIME	GE	1800	20
AND	SYSTIME	LE	0700	25
	EPAGER	CHGTO	DP.ONCALL	30
ELSE				40
	EPAGER	CHGTO	QSYSOPR	60
END				80
		UPDENV		90

■ **Check Current Conditions**

If you need to make sure that a file exists and contains records before a job runs, this example may help you. First, the status of the file GL101 in library QGPL is retrieved. If the file exists and contains records, the job is okay to run. Otherwise, the job waits for 5 minutes and checks conditions again.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUOBJSTS	QGPL/GL101 *FILE	10
IF	OBJSTS	EQ	EXISTS	20
AND	OBJSTS	EQ	RECORDS	30
		QUIT		40
ELSE				50
		ADMIN	5	60
END				80

## Change Values in a Data Area

This example shows you how you might update a data area. First, it retrieves the Data Area Definition ACCOUNTING. If QSYSOPR is signed on, it sets the value of the user variable USRFLG1 to Y. If the device TAP01 is active, it sets the value of three user variables, as shown. Then it updates the data area.

**Note:** The variables USRFLG1, USRFLG2, USRLONG1, and USRFLD4 have been assigned to portions of the Accounting data area definition via the Data Area Definition Elements panel.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUOTADFN	ACCOUNTING	10
IF	ACTUSR	EQ	qsysopr	20
	USRFLG1	CHGTO	Y	30
END				40
IF	ACTDEV	EQ	tap01	50
	USRFLG2	CHGTO	Y	60
	USRLONG1	CHGTO	MONTHLYUPDATE98	70
	USRFLD4	CHGTO	ACTION2390	75
END				80
		CHGOTADFN	ACCOUNTING	90

## Is Mainframe File Ready for Job?

This example may be useful if you have files that get downloaded from a mainframe. If the file that you need for the job has been downloaded (exists on your system), you want to run the job. If the file does not exist, you want to assume the download was not scheduled for today (or failed) and you want to skip the job. In addition, you want to change the output queue for the job to PRT01 if PRT02 is busy. Here is code to do all these things.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUOBJSTS	PRT02 *OUTQ	10
IF	ACTDEV	EQ	prt02	30
	OUTQ	CHGTO	prt01	40
		UPDJOB		50
END				55
		RTUOBJSTS	ACCOUNTING/GL101 *FILE	60
IF	OBJSTS	NE	EXISTS	70
		SKIP		120
END				130

## Delay Job If a Certain Job is Active

If a certain job is active, you want to delay the current job 10 minutes.

Logic Operand	Variable	Operation	Operation Values	Seq
IF	ACTJOB	EQ	rbtageoutq	10
		ADMIN	10	20
END				30

■ Skip Job If IT Staff is Still Signed On

In this example, you have set up a Data Area Definition named MISOPER that contains information about who is signed on the system. If any of your operations personnel are still on the system, you want to skip the job. You also want to know who is still on the system.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUOTADFN	MISOPER	10
IF	USRFLG1	EQ	Y	20
		SKIP		30
		SNDSMSG	MSG('CLARK is still signed on th...	40
ELSE				50
IF	USRFLG2	EQ	Y	60
		SKIP		70
		SNDSMSG	MSG('Bill is still signed on the...	80
ELSE				90
IF	USRFLG3	EQ	Y	100
		SKIP		110
		SNDSMSG	MSG('Brian is still on the syste...	120
ELSE				130
IF	USRFLG4	EQ	Y	140
		SKIP		150
		SNDSMSG	MSG('Barbara is still signed on ...	160
ELSE				170
IF	USRFLG5	EQ	Y	180
		SKIP		190
		SNDSMSG	MSG('Diane is still signed on th...	200
END				210
END				220
END				230
END				240
END				250

## ■ Look for Less Busy Job Queue

You want to find a job queue that has four or fewer jobs in the queue. The following example retrieves information about four job queues. Once it finds one with four or fewer jobs in it, it uses that queue. If no queue has that few jobs, it uses whatever queue is listed on the job record.

<b>Logic Operand</b>	<b>Variable</b>	<b>Operation</b>	<b>Operation Values</b>	<b>Seq</b>
		RTUJOBQ	QBATCH	10
IF	JOBQCNT	LE	4	20
	JOBQ	CHGTO	qbatch	30
ELSE				40
		RTUJOBQ	QPGMR	50
IF	JOBQCNT	LE	4	60
	JOBQ	CHGTO	qpgmr	70
ELSE				80
		RTUJOBQ	COMPILE	90
IF	JOBQCNT	LE	4	100
	JOBQ	CHGTO	compile	110
ELSE				120
		RTUJOBQ	RBTSLLEEPER	130
IF	JOBQCNT	LE	4	140
	JOBQ	CHGTO	rbtsleeper	150
END				160
END				170
END				180
END				190
		UPDJOB		200

## ■ Run Restricted State Utilities Unless QSYSOPR is Active

You want to make sure QSYSOPR is off the system before the restricted state utility job runs. Otherwise, skip the job.

<b>Logic Operand</b>	<b>Variable</b>	<b>Operation</b>	<b>Operation Values</b>	<b>Seq</b>
IF	ACTUSR	EQ	Qsysopr	10
		SENDMSG	MSG('Someone is signed on the sys...)	20
		SKIP		30
ELSE				40
		SENDMSG	MSG('No one is signed on the sys...)	50
END				60

■ **Choose a Job Queue Based on Conditions, Check for Active Devices**

You want your OPAL code to pick a job queue for the job and check on several resources. If the needed resources are not active, a message should be sent to QSYSOPR for each one.

The following code starts by retrieving information about the job queue QPGMR. If the job queue is not on hold, today is a workday, and the QPGMR subsystem is active, the job will use the QPGMR job queue. However, if any of those conditions is not true, it checks the job queue QBATCH. If QBATCH does not meet the conditions, it checks QWEEKEND. If none of them meet the conditions, then the job will use the job queue on its job record.

The second part of the code looks for active devices. If a device is not active, it sends a message to QSYSOPR to start the device.

<u>Logic Operand</u>	<u>Variable</u>	<u>Operation</u>	<u>Operation Values</u>	<u>Seq</u>
		RTUJOBQ	QPGMR	10
IF	JOBQHOLD	EQ	N	20
AND	WORKDAY	EQ	Y	30
AND	ACTSBS	EQ	QPGMR	40
	JOBQ	CHGTO	QPGMR	50
ELSE				70
		RTUJOBQ	QBATCH	80
IF	ACTSBS	EQ	QBATCH	90
AND	JOBQHOLD	EQ	N	100
	JOBQ	CHGTO	QBATCH	110
ELSE				150
		RTUJOBQ	QWEEKEND	160
IF	ACTSBS	EQ	QWEEKEND	170
AND	JOBQHOLD	EQ	N	180
	JOBQ	CHGTO	QWEEKEND	190
END				200
END				210
END				220
IF	ACTCTL	NE	CTL01	230
		SNMSG	MSG('Start controller CTL01') TO...	240
END				250
IF	ACTDEV	NE	PRT02	260
		SNMSG	MSG('start printer prt02') TOUSR...	270
END				280
IF	ACTLIN	NE	tel_line	290
		SNMSG	MSG('Start weekend line Tel_Line...')	300
END				310
IF	ACTRCL	NE	help_systems	320
		SNMSG	MSG('Start ROBOT/Client') TOUSR(...)	330
END				340
		UPDJOB		350

## ■ Use Only Authorized Printers

Do you want to make sure that certain jobs are only printed at certain printers for security reasons? Set up an OPAL table of authorized printers and use code similar to the following example to check for printers in the table. This code retrieves the status of the PRT01 output queue. If PRT01 is active and listed in the OPAL table named PRINTERS, then change the output queue to PRT01. Otherwise, check PRT02. If neither condition is true, it will use the output queue on the Robot Schedule job record.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUOBJSTS	PRT01 *OUTQ	10
IF	ACTDEV	EQ	PRT01	15
AND	OUTQ	INTABLE	PRINTERS	30
	OUTQ	CHGTO	PRT01	40
		UPDJOB		45
ELSE				50
		RTUOBJSTS	PRT02 *OUTQ	60
IF	ACTDEV	EQ	PRT02	65
AND	OUTQ	INTABLE	PRINTERS	80
	OUTQ	CHGTO	PRT02	90
		UPDJOB		100
END				110

## ■ Use a Different Job Queue When Conditions are Right

If the job queue QPGMR is not on hold and has fewer than five jobs in it and the QPGMR subsystem is active, change the job's job queue to QPGMR.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUJOBQ	QPGMR	10
IF	JOBQHOLD	EQ	NO	20
AND	ACTSBS	EQ	qpgmr	30
AND	JOBQCNT	LT	5	40
	JOBQ	CHGTO	qpgmr	50
		UPDJOB		60
END				70



■ Skip or Delay Job Based on Completion Code

This example shows you how to skip a job if another job completed either normally or abnormally. If the retrieved job is in some other status, it delays the current job. Finally, it sends a message.

Logic Operand	Variable	Operation	Operation Values	Seq
		RTUJOB	000000005982	10
IF	SYSTIME	LT	1800	20
IF	RBTJOBNBR	NE	T	30
OR	RBTJOBNBR	NE	C	40
		SKIP		50
ELSE				60
END				90
		SENDMSG	MSG('Job 5982 has already or is ...	100
END				110

■ Change Time Range

This example shows you how to change the time range during which the job may run.

Logic Operand	Variable	Operation	Operation Values	Seq
IF	STARTRANGE	GE	0800	10
IF	ENDRANGE	LE	2300	20
		*	Change the time range	30
	STARTRANGE	CHGTO	0830	40
	ENDRANGE	CHGTO	2359	50
		UPDJOB		60
END				80
END				90

## ■ Restart Job If Not Running

You want Robot Schedule to check to see if a job is running. If the job isn't running, you want Robot Schedule to restart it.

The following code checks to see if Robot Console is not active. If Robot Console is not active, the SNDRBTDTA command executes a Robot Schedule job that starts Robot Console.

Logic	Variable	Operation	Operation Values	Seq
IF	ACTJOB	NE	ABCMANAGER	10
AND	ACTJOB	NE	ABCNOTIFY	20
		SNDRBTDTA	** Prompt to see command **	30
END				40

## ■ Restart Subsystem If Not Running

The following example checks to see if subsystems QPGMR or QBATCH are not active. If either subsystem is not active, the SNDRBTDTA command executes a Robot Schedule job that starts the subsystems.

Logic	Variable	Operation	Operation Values	Seq
IF	ACTSBS	NE	QPGMR	30
OR	ACTSBS	NE	QBATCH	40
		SNDRBTDTA	** Press F4 to see command **	50
END				70

■ Page Operator If Backups Not Started on Time

The following example checks to see if a backup has started on time. If the job isn't running, Robot Schedule pages the operator.

Logic Operand	Variable	Operation	Operation Values	Seq
IF	TIME	GE	0900	10
AND	ACTJOB	EQ	BACKUP	20
		SKIP		30
ELSE				40
		PAGE	MSG('backups have not started ye...	50
END				60

■ Check for Virtual Day

The following example checks to see if it is a virtual day, defined by the user. If it is, it sends a message to the operator to run the month-end processing.

Logic Operand	Variable	Operation	Operation Values	Seq
IF	DAYMTH	EQ	1	10
AND	TIME	GE	2000	20
		*	Today is Day #1 and the time is	30
		*	after 8pm	40
IF	DAYMTH	EQ	2	50
AND	TIME	LE	0800	60
		*	AND today is day #2 past midnight	70
		*	and the current time is	80
		*	before 8am, then month end	85
		*	processing can continue	86
		SENDMSG	MSG('Run month end processing.')...	90
		*	Run month end processing	100
ELSE				110
		SKIP		120
ELSE				130
		SKIP		140
END				150

## A

ACTCTL variable 63  
ACTDEV variable 63  
ACTJOB variable 66  
ACTLIN variable 64  
ACTRCL variable 64  
ACTSBS variable 66  
ACTUSR variable 67  
ADD operation 70  
Adding a condition 36  
ADDMIN operation 47  
Alphanumeric variables 83  
AND logic operand 36

## B

BCAT operation 72  
BLANK constant 44

## C

CAT operation 72  
Changing  
    An environment permanently 59  
    Job setup temporarily 53  
    next statement processed 77  
    Scan/replace values 24  
    User variable values 70, 71, 72  
Character variables 83  
Check Device Status 63  
Check work management status 66  
CHGDTADFN operation 60  
CHGTO operation 70  
Commands  
    RBASNDMSG 75  
    RBTOPALSRC 25  
    RTVMQDEPTH 48  
    SENDMC 74  
    SNDRBTDTA 78  
Compare values 41, 43  
Completion codes 54  
Concatenating information 72  
Conditional operations 33  
Conditions 79  
Contains comparison operation 42  
CT operation 42

## D

Data Area Definition Elements panel 31  
Data Area Definition panel 30  
DAY named variable 81  
Day number in the month variable 81  
Day of the week variable 81  
DAYMTH named variable 81

DC comparison operation 42  
Delaying a job 47  
Deleting  
    A line of OPAL code 15  
    OPAL objects 19  
Device status  
    DEVSTS variable 65  
    List of operation values 65  
Displaying  
    OPAL source code command 25  
    Trace data 21, 23  
Doesn't Contain operation 42

## E

ELSE logic operand 38  
END logic operand 35, 39  
Environment variables table 59  
EQ operation 41  
Examples  
    Change job environment 91  
    Change job queue 57, 69, 96  
    change job queue 68  
    Change job setup 90  
    Change output queue 53, 58, 62  
    Change pager based on time of day 91  
    Change time range 97  
    Change values in a data area 60, 92  
    Check current conditions 91  
    Check for virtual day 99  
    Check if mainframe file is ready 92  
    Choose job queue, check devices 95  
    Delay if another job is still active 92  
    Delay if another job is still running 52  
    Delay if TAP01 not ready 47  
    Delay if user is active 90  
    Go into restricted state unless a user is active 94  
    Look for less busy job queue 94  
    Page if job will be skipped 89  
    Page operator if backups not started on time 99  
    Prevent from running before a time 90  
    Restart job if not running 98  
    Restart subsystem if not running 98  
    Skip if file doesn't meet conditions 45, 62  
    Skip if retrieved job completed normally 52, 53  
    Skip if staff is signed on 93  
    Skip or delay based on completion code 97  
    Use only authorized printers 96  
    Using a Robot Reserved Variable 90  
    Using RTVJOB 52  
    Using RTVJOBQ 67  
EXISTS value 62  
Extended Command Entry panel 16

**F**

Fields in an OPAL statement  
 Description 32  
 Logic operand 33  
 Operation field 41  
 Operation values field 44  
 Variable field 80  
 Finding OPAL jobs on the JSL 53  
 Flag variables 84  
 Floating point variables 85

**G**

GE operation 42  
 Generating OPAL programs 18  
 Greater Than operation 42  
 Greater Than or Equal To operation 42  
 Group member job setup 52  
 GT operation 42

**I**

IF logic operand 34  
 INTABLE operation 43  
 Integer variables 85  
 Introduction to Robot OPAL 7

**J**

Job operations 64, 66  
 Job variables table 56  
 JOBNAME variable 86  
 JOBNBR variable 86  
 JOBQCNT operation 68  
 JOBQCNT variable 68  
 JOBQHOLD variable 69  
 JOBUSER variable 86

**L**

LASTDAY variable 82  
 LE operation 42  
 Less Than operation 42  
 Less Than or Equal To operation 42  
 LOCKED value 62  
 Logging Level 13  
 Logic checking 14  
 Logic control 33  
 Logic operand field 33  
 Logic operands  
 AND 36  
 ELSE 38  
 END 39  
 IF 34  
 OR 37  
 LT operation 42

**M**

Maintain Data Area Definitions panel 29  
 Maintain OPAL Objects panel 10  
 Maintain OPAL Tables panel 26  
 Make OPAL inactive 13  
 Message queue depth monitoring 48  
 MQ Series messaging software 48

**N**

NE operation 41  
 Nested IFs 34  
 NO constant 44  
 Not Equal To operation 41  
 NOTINTABLE operation 43  
 Numeric variables 85

**O**

OBJSTS variable 62  
 OPAL examples 89  
 OPAL Simulator panel 20  
 OPAL statement processing 32  
 OPAL Tables  
 Entering elements 28  
 OPAL quick reference 104  
 OPAL Table Elements panel 28  
 OPAL Table panel 27  
 Operands  
 AND 36  
 ELSE 38  
 END 39  
 OR 37  
 Operation Values field  
 Completion information 54  
 Introduction 44  
 Robot Schedule job number 54  
 Operations  
 ADD 70  
 ADDMIN 47, 48, 49, 50  
 CHGDTADFN 60  
 CT 42  
 DC 42  
 EQ 41  
 GE 42  
 GT 42  
 INTABLE 43  
 LE 42  
 LT 42  
 NE 41  
 NOTINTABLE 43  
 PAGE 75  
 QUIT 76  
 RTVDTADFN 60  
 RTVJOB 51

RTVJOBQ 67  
SENDMC 74  
SNDRBTDTA 78  
SUB 71  
UPDENV 58  
UPDJOB 53  
OPerator Assistance Language panel 14  
OR logic operand 37

## P

PAGE operation 75  
Pager messages 75  
Prerequisite Job Variables 54  
Printing your OPAL objects 24  
Processing OPAL code 32  
Prompting  
    Commands in OPAL 17  
    Fields in an OPAL statement 15

## Q

Quick reference charts  
    Environment variables 59  
    Job variables 56  
    OPAL 104  
QUIT operation 76

## R

RBTJOBNAME variable 55  
RBTJOBNBR variable 55  
RBTOPALSRC command 25  
Reading job setup information 51  
Reading or Changing Data Areas 60  
Reading or Changing Job Setup 51  
RECORDS value 62  
Renumbering OPAL statements 15  
Retrieving group member information 52  
Robot calendar variables 81  
Robot OPAL Object Menu 9  
Robot OPAL Trace Statements panel 23  
Robot Alert pager 73, 75  
Robot Client interface 64  
Robot Console message center 73, 74  
RTVDTADFN operation 60  
RTVENV operation 57  
RTVJOB operation 51  
RTVJOBQ operation 67  
RTVMQDEPTH command 48  
RTVOBJSTS operation 61

## S

SBMTIME variable 87  
Scan/Replace OPAL Values panel 24  
Scheduling objects  
    Robot OPAL 9  
Scheduling Objects Menu 9  
Selecting  
    Logging Level 13  
    When OPAL should run 13  
Sending messages 73  
SENDMC operation 74  
SKIP operation 45  
SKIPSTATUS variable 46  
SNDBRKMSG operation 73  
SNDMSG operation 73  
SNDRBTDTA operation 78  
SUB operation 71  
Subtraction 71  
Syntax checking 18  
SYSDATE variable 88  
SYSTEM variable 87  
SYSTIME variable 88

## T

Temporary jobs 47, 53  
Testing your OPAL code 20  
THISJOB variable 55  
Tracing which statements executed 23

## U

Updating  
    Data areas 60  
    Environments 58  
    Jobs 53  
UPDJOB operation 53  
User Variables  
    Changing the value of 71, 72  
    USRCNTn 85  
    USRFLGn 84  
    USRLONGn 84  
    USRNBRn 85  
USRCNTn variable 85  
USRFLGn variable 84  
USRLONGn variables 84  
USRNBRn variable 85

**V**

## Variables

- ACTCTL 63
- ACTDEV 63
- ACTJOB 66
- ACTLIN 64
- ACTRCL 64
- ACTSBS 66
- ACTUSR 67
- DAY 81
- DAYMTH 81
- DEVSTS 65
- Environment variables table 59
- Introduction 79
- Job variables table 56
- JOBNAME 86
- JOBNBR 86
- JOBQCNT 68
- JOBQHOLD 69
- JOBUSER 86
- OBJSTS 62
- RBTJOBNAME 55
- RBTJOBNBR 55
- SKIPSTATUS 46
- SYSDATE 88
- SYSTIME 88
- THISJOB 55
- USRCNTn 85
- USRFLDn 83
- USRFLGn 84
- USRLONGn 84
- USRNBRn 85
- Variable field 80
- WORKDAY 81

**W**

- WebSphere MQ messaging software 48
- WEEKNO variable 82
- When OPAL should run 13
- Where is an OPAL Object Used panel 19
- Where to enter
  - Data Area Definition
    - elements 31
    - name 30
  - Logging Level 13
- OPAL
  - active or inactive 13
  - object description 12
  - object name 12
  - statements 14
  - table elements 28
- WORKDAY named variable 81

**Y**

- YES constant 44

**Notes:**



## Robot Schedule OPAL Quick Reference

### Logic Operands

IF	variable	comparison	value
AND	variable	comparison	value
OR	variable	operation	value
ELSE	—	—	—
END	variable*	operation	value*

\*Optional

### OPAL Variable

### Value

*ROBOTJOB	Robot job finder
ACCTCODE	Accounting code
ACTCTL	Controller active?
ACTDEV	Device active?
ACTJOB	Job active?
ACTLIN	Line active?
ACTRCL	Client active?
ACTSBS	Subsystem active?
ACTUSR	User active?
ATPOOL	AutoTune pool size
CALENDAR	Calendar name
CURLIB	Current library name
DATEOBJ	Date object
DAY	Day of week (Mon=1)
DAYMTH	Day of the month
DEVSTS	Device status
EACCTCODE	Environment accounting code
EATPOOL	Environment AutoTune pool size
ECALENDAR	Environment calendar
ECURLIB	Environment current library
EJOB	Environment job description
EJOBDLIB	Environment job description library
EJOBQ	Environment job queue
EJOBQLIB	Environment job queue library
ELIBLNAME	Environment library list name
EMSGQ	Environment message queue
EMSGQLIB	Environment message queue library
EMSGRPY	Environment message reply type
ENDRANGE	Ending time range
EOUTQ	Environment output queue
EOUTQLIB	Environment output queue library
EPAGER	Environment pager
EUSER	Environment user profile
GROUP	Group name
JOB	Job description
JOBDLIB	Job description library
JOBENV	Job environment name
JOBNAME	Job name

### OPAL Variable

### Value

JOBNBR	Job number
JOBQ	Job queue
JOBQCNT	Number of jobs in job queue
JOBQHOLD	Is the job queue on hold?
JOBQLIB	Job queue library
JOBUSER	Job user
LASTDAY	Last day of the month
LIBLNAME	Library list name
MAXRUN	Maximum run time
MAXRUNACT	Action to take when job reaches maximum run time
MSGQ	Message queue
MSGQLIB	Message queue library
MSGRPY	Message reply type
NONWORK	Nonworking day
OBJSTS	Object status (EXISTS, LOCKED, or RECORDS)
OPALNAME	OPAL procedure name
OUTQ	Output queue
OUTQLIB	Output queue library
PAGER	Pager name
RBTJOBNAME	Robot job name
RBTJOBNBR	Robot job number
SBMTIME	Submit job time
SCHEDULE	Schedule override code
SKIPSTATUS	Skip status
STARTRANGE	Start time range
SYSDATE	System date
SYSTEM	System name
SYSTEM	System time
THISJOB	Current Robot job number
TIME	Time to run Robot job
USER	Robot job user
USRCNTn	User defined numeric (n = 1-5)
USRFLDn	User defined field (n = 1-5)
USRFLGn	User defined flag (n = 1-5) Longer
USRLONGn	character variable (n = 1-5) User
USRNBRn	defined numeric (n = 1-5) Week
WEEKNO	number
WORKDAY	Workday or not

### Comparisons

EQ	Equal	NE	Not equal
CT	Contains	DC	Doesn't contain
GT	Greater than	LT	Less than
GE	Greater than or equal to	LE	Less than or equal to
INTABLE	In OPAL table	NOTINTABLE	Not in OPAL table

# Robot Schedule OPAL Quick Reference

---

<b>Operation</b>	<b>Operation Value</b>
*	Comment statement
ADD	Increment
ADMIN	Minutes to increase the scheduled submit time
BCAT	Value to concatenate
CAT	Value to concatenate
CHGDTADFN	Data area definition to change
CHGTO	New value
GOTO	Tag name
PAGE	F4 (enter pager message)
QUIT	None (ends OPAL processing)
RTVDTADFN	Data area definition to retrieve
RTVENV	Environment to retrieve
RTVJOB	Job number to retrieve
RTVJOBQ	Job queue to retrieve
RTVOBJSTS	Object to retrieve
SENDMC	F4 (enter new message)
SKIP	None (skip job when processing)
SNDBRKMSG	F4 (enter break message text)
SENDMSG	F4 enter message to send to message queue
SNDRBTDTA	User job name and status
SUB	Value to subtract
TAG	Tag name
UPDENV	None
UPDJOB	None