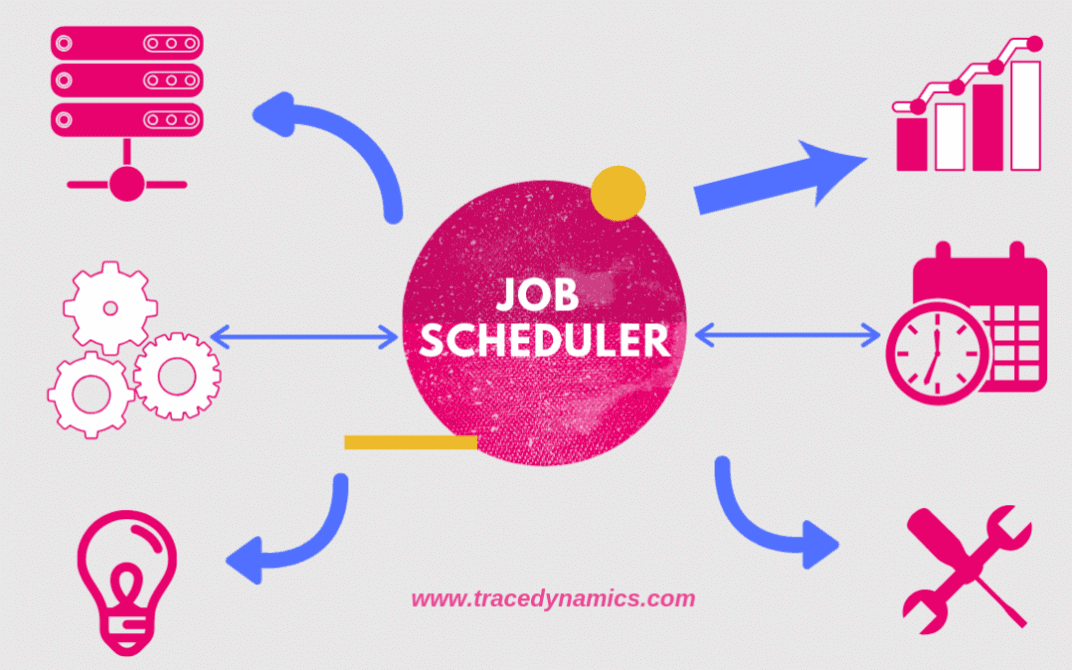
[**AutoSys: A Beginner’s Guide to Master AutoSys Workload Automation**](https://www.tracedynamics.com/autosys/)

by [Admin](https://www.tracedynamics.com/author/admin/) | Jun 28, 2017



Following AutoSys wiki article/AutoSys training help you to understand about this job scheduling tool.

**Contents**[[hide](https://www.tracedynamics.com/autosys/)]

* [1 Definitions](https://www.tracedynamics.com/autosys/#Definitions)
* [2 What is AutoSys?](https://www.tracedynamics.com/autosys/#What_is_AutoSys)
* [3 Features:](https://www.tracedynamics.com/autosys/#Features)
* [4 Benefits:](https://www.tracedynamics.com/autosys/#Benefits)
* [5 AutoSys is the leader among Job Scheduling Management System Tools?](https://www.tracedynamics.com/autosys/#AutoSys_is_the_leader_among_Job_Scheduling_Management_System_Tools)
* [6 Autosys AE Architecture :](https://www.tracedynamics.com/autosys/#Autosys_AE_Architecture)
* [7 How does AutoSys work?](https://www.tracedynamics.com/autosys/#How_does_AutoSys_work)
* [8 Creating AutoSys Jobs:](https://www.tracedynamics.com/autosys/#Creating_AutoSys_Jobs)
* [9 AutoSys Job Types:](https://www.tracedynamics.com/autosys/#AutoSys_Job_Types)
* [10 Job Status/States – Infographic](https://www.tracedynamics.com/autosys/#Job_StatusStates_Infographic)
* [11 ON HOLD vs ON ICE?](https://www.tracedynamics.com/autosys/#ON_HOLD_vs_ON_ICE)
* [12 Managing File Transfer(FTP/SFTP/SCP) Jobs](https://www.tracedynamics.com/autosys/#Managing_File_TransferFTPSFTPSCP_Jobs)
* [13 What are AutoSys JIL file/scripts?](https://www.tracedynamics.com/autosys/#What_are_AutoSys_JIL_filescripts)
  + [13.1 AutoSys update job:](https://www.tracedynamics.com/autosys/#AutoSys_update_job)
* [14 AutoSys Commands – Cheat Sheet](https://www.tracedynamics.com/autosys/#AutoSys_Commands_8211_Cheat_Sheet)
  + [14.1 autocal\_asc command :](https://www.tracedynamics.com/autosys/#autocal_asc_command)
  + [14.2 CRON2JIL command :](https://www.tracedynamics.com/autosys/#CRON2JIL_command)
  + [14.3 sendevent command :](https://www.tracedynamics.com/autosys/#sendevent_command)
  + [14.4 Calling an AutoSys job from Unix Script(AutoSys commands in unix):](https://www.tracedynamics.com/autosys/#Calling_an_AutoSys_job_from_Unix_ScriptAutoSys_commands_in_unix)
  + [14.5 autorep Command:](https://www.tracedynamics.com/autosys/#autorep_Command)
  + [14.6 autorep Command Syntax:](https://www.tracedynamics.com/autosys/#autorep_Command_Syntax)
* [15 AutoSys Alarms:](https://www.tracedynamics.com/autosys/#AutoSys_Alarms)
* [16 Best Practices For AutoSys Errors/Exception Handling:](https://www.tracedynamics.com/autosys/#Best_Practices_For_AutoSys_ErrorsException_Handling)
* [17 Exit Codes/Exit Status](https://www.tracedynamics.com/autosys/#Exit_CodesExit_Status)
* [18 Oracle E-Business Suite Jobs](https://www.tracedynamics.com/autosys/#Oracle_E-Business_Suite_Jobs)
* [19 Interview FAQ](https://www.tracedynamics.com/autosys/#Interview_FAQ)
* [20 What is AutoSys?](https://www.tracedynamics.com/autosys/#What_is_AutoSys-2)
* [21 What is global name in AutoSys batch scheduling tool?](https://www.tracedynamics.com/autosys/#What_is_global_name_in_AutoSys_batch_scheduling_tool)
* [22 What are all the Autosys job status?](https://www.tracedynamics.com/autosys/#What_are_all_the_Autosys_job_status)
* [23 What is AutoSys JIL file?](https://www.tracedynamics.com/autosys/#What_is_AutoSys_JIL_file)
* [24 How to create an AutoSys Job?](https://www.tracedynamics.com/autosys/#How_to_create_an_AutoSys_Job)
* [25 What is file watcher in AutoSys?](https://www.tracedynamics.com/autosys/#What_is_file_watcher_in_AutoSys)
* [26 How to check the job definition history changes in AutoSys?](https://www.tracedynamics.com/autosys/#How_to_check_the_job_definition_history_changes_in_AutoSys)
* [27 How to export Autosys JIL command?](https://www.tracedynamics.com/autosys/#How_to_export_Autosys_JIL_command)
* [28 How to mark job as success in AutoSys?](https://www.tracedynamics.com/autosys/#How_to_mark_job_as_success_in_AutoSys)
* [29 How to run Python script in Autosys?](https://www.tracedynamics.com/autosys/#How_to_run_Python_script_in_Autosys)

**Definitions**

The best definition of AutoSys is

*AutoSys Workload Automation is a multi-platform automated job control system that provides the capability of scheduling, monitoring, and reporting of AutoSys jobs for enterprise applications.*

**What is AutoSys?**

AutoSys Workload Automation is a multi-platform automated job control system.

* This tool provides the capability of scheduling, monitoring and reporting of AutoSys jobs.
* These jobs and their job definitions/JIL defined in the centralized AutoSys server.
* Autosys server will communicate with the remote AutoSys agent.
* i.e. Installed in the remote machine like Windows/Unix to perform the job operations.

How application Dev teams perform AutoSys job monitoring, manage enterprise workloads and scheduling using AutoSys software?

I.e. from anywhere at any time, with a custom dashboard using CA Workload Automation. Check out this video.

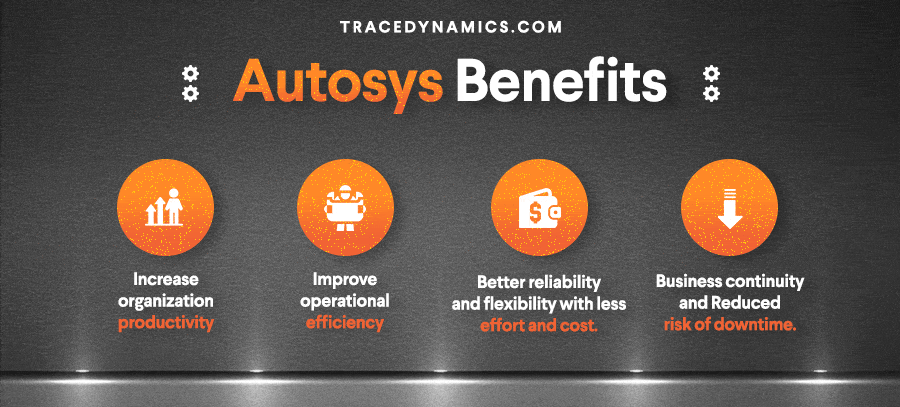
**Features:**

* AutoSys Workload Automation (earlier CA Workload Automation AE) provides end-to-end job control.
* It provides visibility of enterprise application workloads and dependencies for business-critical application workload.
* AutoSys Workload Automation tool is now product of Broadcom Inc.
* It provides one stop solution for job management, scheduling and job [development life cycle](http://tracedynamics.com/sdlc/).
* Acts like autonomous systems for enterprise job management.
* Supports database like Microsoft SQL database, Oracle database, Sybase, and IBM DB2.
* Using DB jobs you can execute [**SQL**](http://tracedynamics.com/sql/) commands, SQL queries or any other search query against a database.
* AutoSys tool supports scheduling and execution of ETL tool jobs like datastage etl.
* Provides workflow details of each job along with event report definition.
* Supports running jobs through virtual machine.
* This tool provides integration feature for Hadoop.
* Allows to visualize and manage the hadoop xml jobs and standard jobs.
* This tool is cloud ready and supports majority of cloud platforms like AWS data pipeline.
* Supports Windows, Unix, Solaris, AIX, Linux, HP-UX servers, Unix Sybase sql server, Mainframes.
* An web server provides run time environment for running web services.
* Supports SSIS/DTS package execution (using dtexec command line utility)
* Other alternate or extra technology’s of AutoSys job scheduling tools in market are BMC Control-M, Skybot, Tivoli Workload Scheduler, Flux, Batchman, Active Batch and so on.
* SDLC scrum friendly and offers unicenter AutoSys service desk
* Workload Automation tool latest installation version is Autosys r11 (also supports java 1.7).
* Versions before to r11.3 called as Unicenter AutoSys Job Management.
* compatible with AWK programming language.
* Powerful Analytics, Visualization, Simulation, Authorization and Automation Capabilities.
* Supports failover server activities done during disaster recovery or business continuity process.
* Providing capability to integrate with ServiceNow by using CA connector.

**Benefits:**

Following were the high level benefits.

* Increase organization productivity.
* Improve operational efficiency.
* Better reliability and flexibility with less effort and cost.
* Business continuity and Reduced risk of downtime.

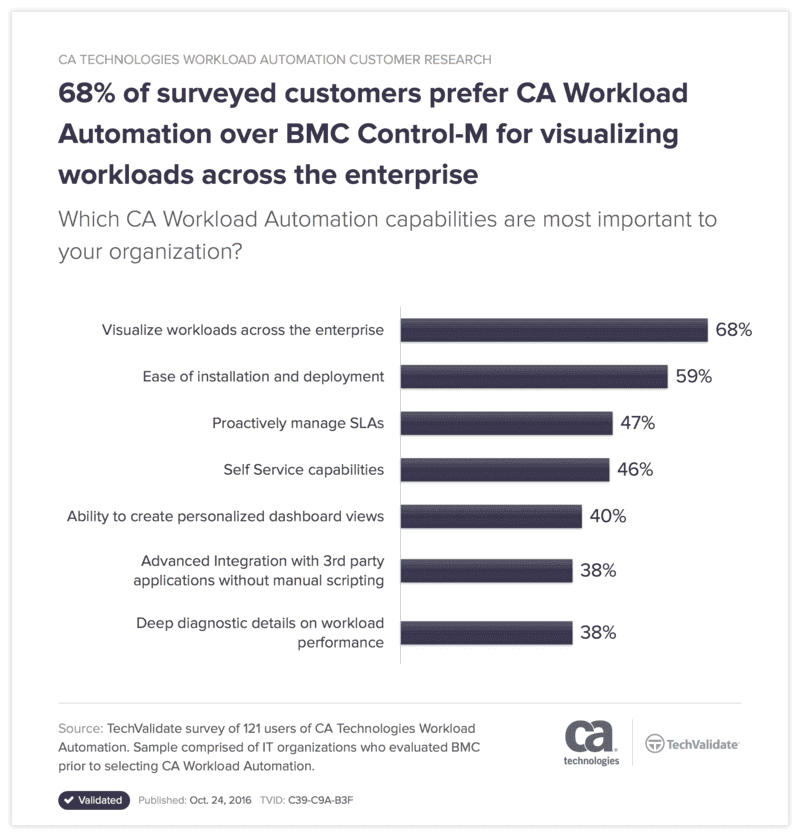


**AutoSys is the leader among Job Scheduling Management System Tools?**

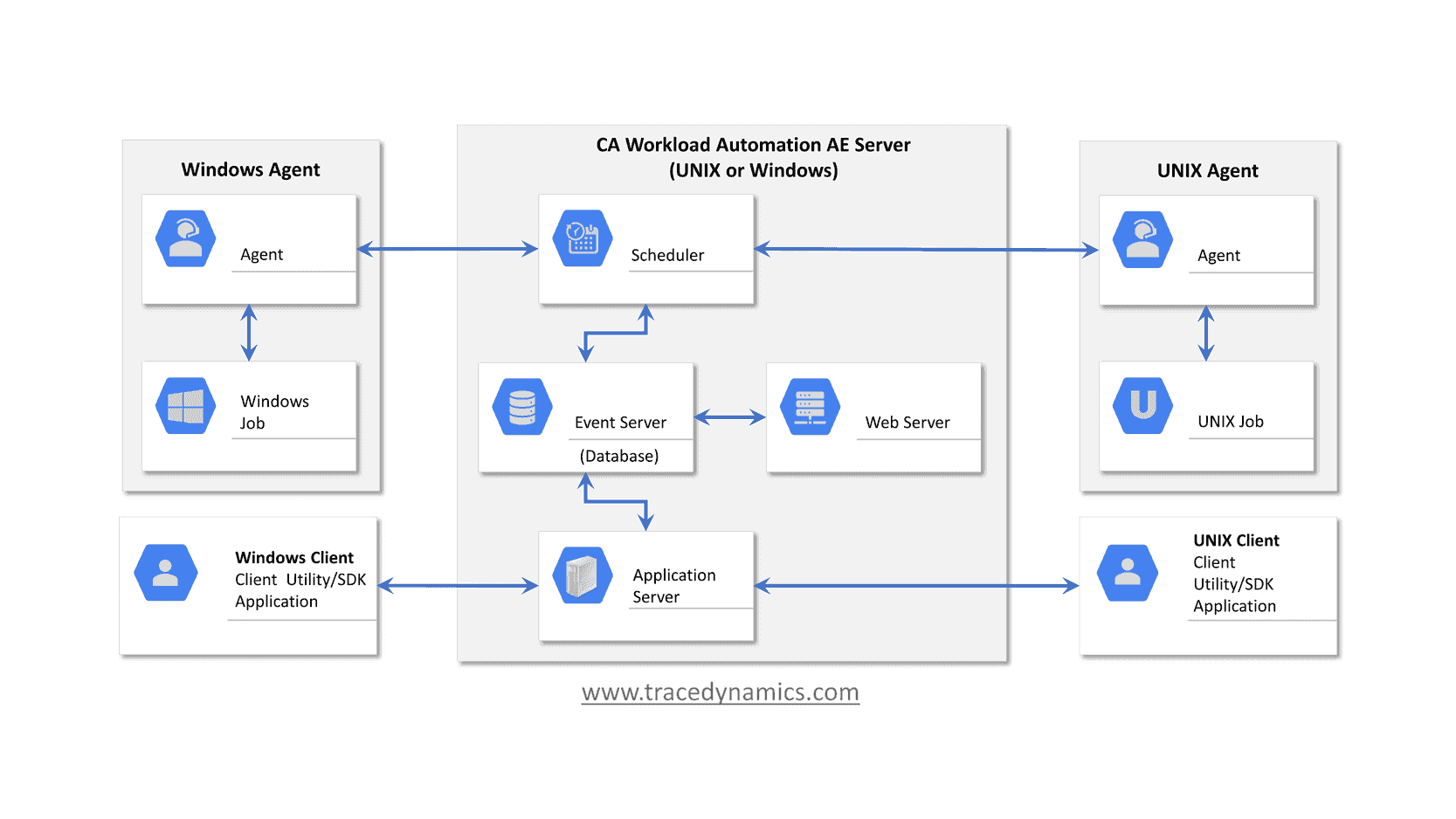
Undoubtedly Yes !

As per the survey conducted by **[TechValidate](https://www.techvalidate.com/)**

“68% of surveyed customers prefer Workload Automation over BMC Control-M for visualizing workloads”.



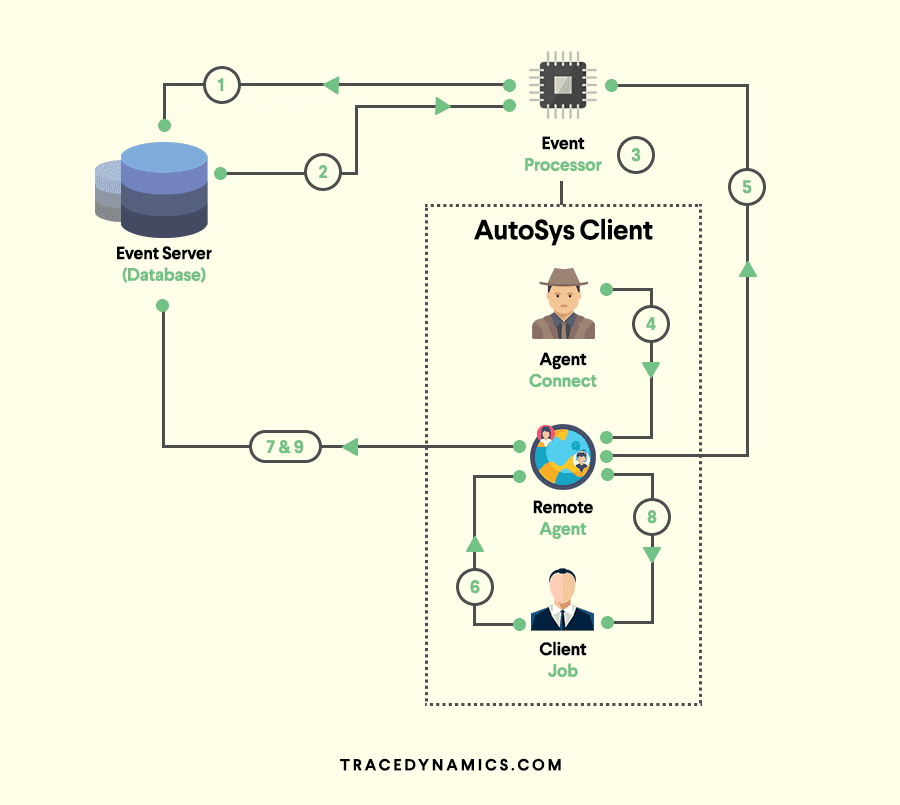
**Autosys AE Architecture :**



CA Workload Automation AE main components are as follows:

* Event server(database)
* Application server
* Web server
* Scheduler
* Agent
* Client

**How does AutoSys work?**



Remote Agent, Event Server, Event Processor are the build blockings.

* **Remote Agent:**The Remote Agent is a process triggered by the event processor to execute a task scheduler on remote machine I.e. on AutoSys client machine.
* **Event Server (AutoSys Database):**The Event Server is the repository where job events and autosys job definition stored.
* **Event Processor:**Event Processor is a crucial component, it reads and processes all the events from the autosys database tables(event server).This process can be a window service or unix and Event Processor will scan the database events continuously.

Majority of the server configurations will be configured by autosys administrator.

**Creating AutoSys Jobs:**

There are two ways to create an AutoSys job definition.

**Using AutoSys GUI(AutoSys Graphical User Interface):**

Through gui or graphical user interface, job attributes can be set to describe where, when, and how a job should run.

**Using Job Information Language (JIL) via command-line interface :**

JIL command is a specification language whichs has its own commands to describe how a job should run.

In other words its a kind of schema which has its own data types and a defined structure/syntax.

Also environment variable passed as profile in JIL script.

**AutoSys Job Types:**

 Jobs in AutoSys classified into following job type.

**Command job:**

The name itself suggests, i.e Command jobs execute commands.

An command job is a single command or executable, windows batch file, or unix script/powershell script/shell script (scripting language).

**File Watcher job:**

The AutoSys file watcher job will watch for the arrival of particular file.

**Box (box jobs):**

AutoSys box jobs are containers that consists other jobs .

The box job in AutoSys used to control and organize process flow.

An impressive feature of this type of job is, boxes can include another boxes.

**Job Status/States – Infographic**

This tool keeps track of the AutoSys status of every job.

The job status used to determine when to start other jobs that are dependent on the job.



The following were the list of job status and their AutoSys change status behaviours.

**STARTING :** An event initiated to the start job procedure with the remote agent.

**RUNNING :**Means the job is in running mode. In case if the job is a box job, this value means that the jobs within the AutoSys box started.

**INACTIVE :** Means the job is inactive. The job has been never executed or its status was intentionally marked to turnoff its previous completion status.

**ACTIVATED :** The top-level box that this job resided now in RUNNING state, but note that job itself has not started yet.

**SUCCESS :** The job completed successfully, by default the Exit Code ‘0’ considered as success. In case of box job, success status means that all the jobs inside the box has completed successfully.

**FAILURE :** The job failed, which means the Exit Code-is greater than zero.

**TERMINATED :** The job terminated while in the running state. A job terminated when user sends a KILL job event.

**RESTART :** The job was unable to start due to application or hardware problems and scheduled to restart.

**QUE\_WAIT :** The job ready to run, but there are not enough machine resources available.

**ON HOLD :** The job is ON HOLD and will not run until it receives job OFF HOLD event.

**ON ICE :** The job will be ON ICE ( AutoSys jobs on ice) until it receives OFF ICE event. In other words, ICE job can be ON/OFF.

**ON HOLD vs ON ICE?**

Its one of the standard AutoSys interview questions.

Here are the details of AutoSys condition which covers ON ICE to OFF ICE vs ON HOLD to OFF HOLD.

* When a job is ON HOLD and If the same job has kept OFF HOLD, Its runs in case of job starting conditions met/satisfied.
* When a job is OFF ICE and If the same Job kept OFF ICE, Its will not run even though the respective job starting conditions met/satisfied.
* If a job is ON HOLD, The dependent jobs will not run.
* Where as If a job is ON ICE, the dependent will run as per the schedule.

**Managing File Transfer(FTP/SFTP/SCP) Jobs**

File Transfer jobs allows transferring of,

binary, ASCII, EBCDIC files between an agent computer/remote location or FTP server.

File transfer jobs defined as following:

**FTP:**

Lets you transfer files using File Transfer Protocol(FTP).

**SFTP:**

We can securely transfer binary or ASCII files using the Secure File Transfer Protocol (SFTP).

Also SFTP protocol supports wildcard transfers,

so you can upload multiple files to a remote FTP server or download multiple files to the agent machine.

**SCP:**

Securely transfer binary files using the Secure Copy Protocol (SCP).

Note that SCP protocol doesn’t support wildcard transfers.

**What are AutoSys JIL file/scripts?**

JIL stands for Job Information Language.

JIL file/scripts used to built an job definition via command-line interface.

JIL scripts contain one or more subcommands/attribute statements.

The following are the jil command syntax for **subcommand** and **attribute**.

**sub\_command:object\_name**

sub\_command -> Defines a JIL sub command.

object\_name -> Defines the name of the object (i,e job / machine) to act on.

**attribute\_keyword:value**

attribute\_keyword -> Defines a valid JIL attribute.

value -> Defines the setting to apply a attribute.

**1. Sample JIL command example for an command job “helloJob.jil”**

insert\_job:helloJob  
machine :unix machine name  
owner :username  
command :echo “Hello this a welcome command job”

note: insert job denotes the job name  
**2. To add the command job “helloJob.jil” in database(db)**

Run the following autosys unix commands through command prompt.

jil < echoJob.jil

If you want to read JIL tag and the corresponding job associated with it,

load the information via datastage, then you can grep the JIL script.

**grep “hello\_job:echo” \*.jil | cut -d ‘ ‘ -f2**

By using above command syntax solution, you can get the job names.

I.e. via filter options available in Sequential file/script and load using datastage job.

**AutoSys update job:**

Lets go through update job syntax and each attribute/parameter in detail.

update\_job: BOX\_N  
start\_times: “18:00”

Here start\_times aceepts [numeric type](http://tracedynamics.com/string-to-int-c-plus-plus/) of time format.

Save into a jil file and run it using jil executable script command **jil < filename.jil**

**Example:**

**insert\_job:** HelloWorld

**job\_type:** c

**box\_name:** box1

**command:** Datasource/scripts/Autosys/oraclesqlquery/storedprocedure.sh

**machine:** localhost

**owner:** admin@localhost

**permission:** gx,ge,wx,we,mx,me

**date\_conditions:** 1

**days\_of\_week:** all

**start\_times:** 0,30]

strong>term\_run\_time: 180

**alarm\_if\_fail:** 1

Let’s go through each attribute of jil commands from above example

**insert\_job:**

Its a JIL keyword which used as Name to identify a job.

**job\_type:**

It defines type of a job and in this case **C** refers to a **Command Job.**

**command:**

This attribute can be any command, UNIX shell script, executable, or batch file.

**machine:**The client machine on which the command should run.

**owner:**

It will be the userID on which the command will be running. For above scenario, I gave autosys admin as owner.

**permission:**  
First letter (g,w,m) combination refers to group user and last letter (x,e) refers to the rights for edit (e) and execute(x).

**date\_conditions:**  
The AutoSys date\_conditions which enables the run calendar for the respective job.

**days\_of\_week:**

Days on which the job should run.

**start\_times:**

autosys start times, This specifies start time of job that it should trigger/run.

**term\_run\_time:**‘

term\_run\_time result in a job status being set to Terminated after specified time.

As per example above, it will set the job status to terminated after running 180 minutes.

**alarm\_if\_fail :**

If the value is 1, then you will receive an failure alert and If the value is 0, then you will not receive any alert.

max run alarm in AutoSys is an classic example for this.

Whenever job runs than the specified time, an max run alarm alert will trigger.

**Note:**AutoSys start\_mins leveraged, if a job needs to trigger every minute/minutes.

The following syntax will start the job for every 10 mins.

**start\_mins: 00,10,20,30,40,50**

**AutoSys Commands – Cheat Sheet**

Following cheat sheet consists list of Autosys basic commands used in unix or any other operating system.

| **Purpose** | **Command** |
| --- | --- |
| To convert from CRON to JIL (only in UNIX) | cron2jil |
| To define objects like jobs/machines | jil |
| For event commands | sendevent |
| Check system status | autoflags autoping autosyslog chk\_auto\_up chase |
| To define calendars | autocal\_asc |
| To maintain Databases | archive\_jobs archive\_events clean\_files autotrack dbspace dbstatistics DBMaint |
| To report job status | autorep autostatad autostatus forecast monbro |
| To monitor jobs | autosyslog autorep monbro |
| To report job dependencies and conditions | job\_depends |
| To manage security | autosys\_secure |
| To manage time zone | autotimezone |

Out of above, let me go through some important commands that are frequently used.

**autocal\_asc command :**

**autocal\_asc** used to define AutoSys calendars.

Calendars are very helpful incase of job/box specific events, processed based on a date in a calendar

eg: holiday batch job or batch jobs processing based on the date etc).

AutoSys condition parameter “date\_conditions” related to calendar concepts.  
`  
For a job to run ‘standalone’,

the ‘Date/Time conditions’ attribute in the ‘Schedule’ section must reflect ‘true’.

If the date\_conditions field is not set to ‘1’ (true),

AutoSys review ignore ‘Days’ or ‘Time’ attributes and rely on dependencies for job scheduling software.

If the ‘Date/Time conditions are ‘true’,

the job or job box will also reference ‘Run days’, or ‘Run calendar’ and/or ‘Exclude Calendar’;

and, ‘Times of day’ or ‘Minutes past hour’.

These job attributes will covered in more details in other sections of autosys documentation.

**CRON2JIL command :**

[**Cron**](http://tracedynamics.com/cron-jobs/) is a job scheduler in unix based operating systems.

Used to schedule a job (either command or the scripts )that runs at fixed intervals, schedule.

Cron controlled by a crontab file,

a config file that defines shell commands to run cyclically on a specified schedule.

**cron2jil** command is used to convert from CRON to JIL.

**sendevent command :**

The sendevent AutoSys command is a component that sends events,

to start or stop jobs, stop the AutoSys scheduler, put a job on hold,

set a AutoSys global variable, cancel a scheduled jobs event and so on.

Basically this command will send event in AutoSys scheduling tool.

Following are sample sendevent commands, generally denoted by using sendevent e job name.

To start a job:

**sendevent -E STARTJOB -J**

To force start a job : **sendevent -E FORCE\_STARTJOB -J**

AutoSys ON ICE:

**sendevent -E ON\_ICE -J**

Put job OFF ICE :

**sendevent -E OFF\_ICE -J**

AutoSys kill job:

**sendevent -E KILLJOB -J**

Event logs verified to check the output, errors.

**Calling an AutoSys job from Unix Script(AutoSys commands in unix):**

We can call an AutoSys job via unix executable script, following is an example.

**JobName :** SchedulerCommand

**Custom Script Name:** callJob.ksh

**Server:** AutoSys Server

In shell script “callJob.ksh”, use the below sendevent command to pass the event type and job name.

**sendevent -E FORCE\_STARTJOB -J SchedulerCommand**

Now save the unix script and invoke it using command **./callJob.ksh** from AutoSys server.

that’s it, the job ‘SchedulerCommand’ will be successfully force started.

**autorep Command:**

**autorep command** generates reports about jobs, machines, and autosys global variables defined in database.

It pulls data from the database to formulate the reports.

autorep command used to achieve the following.

* Displaying the current machine load information.
* Backing up job definitions by extracting definitions and saving them to an output file.
* Displaying a summary of all defined jobs.
* Listing relvant information of any job last run (previous run).

If you want to check the job definition history changes, then you can acieve it using autorep command below

**autorep -j “mention the job name”**

Also by using wildcards(%) with autorep command, you can export the AutoSys JIL’s.

**autorep -j % -q > /temp/filename.jil**

**autorep Command Syntax:**

| **command purpose** | **autorep command syntax** |
| --- | --- |
| To report on jobs in a group | autorep -B group [-d | -s | -q | -o overnum] [-I application] [-n] [-w] [-t] [-R run\_num] [-L print\_level] |
| To report on jobs in an application | autorep -I application [-d | -s | -q | -o overnum] [-B group] [-n] [-w] [-t] [-R run\_num] [-L print\_level] |
| To report on a machine | autorep -M machine [-p | -q] [-d | -s] [-n] [-w] |
| To report on a user-defined job type | autorep -Y job\_type [-d | -s | -q ] [-n] [-w] |
| To report on global variables | autorep -G global\_name [-n] [-w] [-d | -s] |
| To report on a external instance | autorep -X ext\_instance [-s | -q ] [-n] [-w] [-t] |
| To report on a virtual resource | autorep -V virtual\_resource\_name [-d | -s | -q ] [-n] [-w] |
| To display the version number for the command | autorep -x |
| To display help for the command | autorep -? |

**AutoSys Alarms:**

For an instance, jobs could be dependent on a arrival file and if the file not yet arrived in an expected time frame.

Then the job fails and then alarm notification alert the respective groups or owners to take appropriate action.

* Alarms provides notification mechanism during alarming situations like AutoSys job scheduler failures, dependencies.
* Upon these kind of notifications, it will allow manual interventions to address the issue.
* Alarms are system specific messages about a detected issue.
* Typically alarms sent through system as a event.

**Best Practices For AutoSys Errors/Exception Handling:**

Exception Handling is one of the key attribute in the software life cycle.

w.r.t AutoSys, its best practice to have exception handling on failed operator which have exit path for only “retry” scenario.

Else having retry without ever exiting the process results in endless loops.

Error/Err files or log files can found by using following commands.

**Scheduler Server and Application Server logs:**

(default) /opt/CA/WorkloadAutomationAE/autouser.ACE/out

**AutoSys WCC logs:**

(default) /opt/CA/WorkloadCC/log

**AutoSys agent logs:**

(default) /opt/CA/WorkloadAutomationAE/SystemAgent/WA\_AGENT/log

**Exit Codes/Exit Status**

In general, a zero Exit Code- indicates success; while a non zero Exit Code- indicates an error. Following are exit or error codes which are frequently seen.

**Exit Code-1**:  
Missing System Agent configuration parameter “oscomponent.cmdprefix.force=true”

and/or the Windows machine definition missing the “opsys: windows” (JIL) attribute.

**Exit Code-2**:

This error is coming from the first line in the job script where the shell was being sourced.

The first line read ‘#!/usr/bin/ksh’.

However, this was an invalid location for setting the shell interpreter which are typically located in /bin and not /usr/bin.

**Exit Code-13**:

Due to a permission error on the std\_out\_file.

**Exit Code-15**:

The job terminated.

**Exit Code-101**:

CHANGE\_STATUS performed on the job. for eg: The job changed to FAILURE or TERMINATED status.

**Exit Code-121**:

Cannot open std\_in\_file. File does not exist or it is inaccessible.

**Exit Code-122**:

Cannot open std\_out\_file. File does not exist or it is inaccessible.

**Exit Code-123**:

Cannot open std\_err\_file. File does not exist or it is inaccessible.

**Exit Code-127**:

Directory that holds the executable is not in the command search PATH.

**Exit Code-256**:

Unable to execute the command.

**Exit Code-512**:

Incorrect command options.

**Exit Code-655/-655 SYSTEM\_ERROR**:

STARTJOB failures because auto\_remote will not start.

**-656 NO\_EXIT\_CODE**:

exit\_code field in database initialized to this.

**-657 PROCESS\_MIA**: Set by a chase-generated FAILURE event.

for eg: chase cannot find the process.

**Oracle E-Business Suite Jobs**

Oracle job scheduler defined to run Oracle E-Business Suite workload.

It is illustrated in the following ways.

* **Request Set Job :** To run multiple programs in oracle application.
* **Single Request Job :** To run single programs in oracle application.
* **Copy Single Request Job :** copying the existing Single Request Job and running it under the AutoSys agent.

[**CA Workload Automation Reviews**](https://www.itcentralstation.com/products/ca-workload-automation?tid=c248-sq-49) from subject matter experts posted in itcentralstation.com.

In this tutorial, I tried to cover as vast i can when compared to other AutoSys tutorials.

Hope you liked AutoSys user guide topics covered in this post , please feel free to share above autosys reference manual.

**Interview FAQ**

**What is AutoSys?**

AutoSys Workload Automation is a multi-platform automated job control system that provides scheduling, monitoring, and reporting of AutoSys jobs.

**What is global name in AutoSys batch scheduling tool?**

To report on global variables, use below command  
  
autorep -G global\_name [-n] [-w] [-d | -s]

**What are all the Autosys job status?**

STARTING, RUNNING, INACTIVE, ACTIVATED, SUCCESS, FAILURE, TERMINATED, RESTART, QUE\_WAIT, ON HOLD, ON ICE

**What is AutoSys JIL file?**

JIL stands for Job Information Language.  
  
JIL file/scripts used to built an job definition via command-line interface.  
  
JIL scripts contain one or more subcommands/attribute statements.

**How to create an AutoSys Job?**

There are two ways to create an AutoSys job definition.  
  
1) Using AutoSys GUI(AutoSys Graphical User Interface).  
2) Using Job Information Language (JIL) via command-line interface.

**What is file watcher in AutoSys?**

File watcher job will watch for the arrival of specific file defined.

**How to check the job definition history changes in AutoSys?**

By running command “autorep -j”. where -j refers to job name.

**How to export Autosys JIL command?**

By applying wildcards(%) with autorep command, we can export the AutoSys JIL’s.  
  
autorep -j % -q > /temp/filename.jil

**How to mark job as success in AutoSys?**

1)Login to autosys wcc client url.  
2)Locate the job you want to mark it as success.  
3)Click on send event and select ‘mark as success’ from the drop-down list, then the job marked as success.  
  
You can perform the same through [“send event”](http://tracedynamics.com/autosys/#sendevent) command in server command line.