Syzygy Incorporated

[Version 7]
October 10, 2013

SyzAUTO/z

```
E3C8C500 E2E8E2E3 C5D4E200 C5E7D7C5    *THE.SYSTEMS.EXPE*
D9E3....00 00000E3 C8C500E2 E8E2E3C5    *RTS....THE.SYSTE*
D4E2..C5 E7D7C5D9 E3E00000 0000E300    *MS.EXPERTS....TH*
C500.. E2E8.. E200..E7 D.C5E3..         ..STEM. EXPERT*
E0000.. 00E..C5 00E8E2 E..4E2           *....THE.. STEMS*
..D7 C5D..3E2 00000.00 E..8C500..       *PERTS..THE.*
E2E8E2E3 C..4E200 C5E7D7C5..E3E2..      *SYSTEMS.EXI..*
000000E3 C8C500E2 E8E2E3C5 D4E200C5..   *STEMS.E*
```
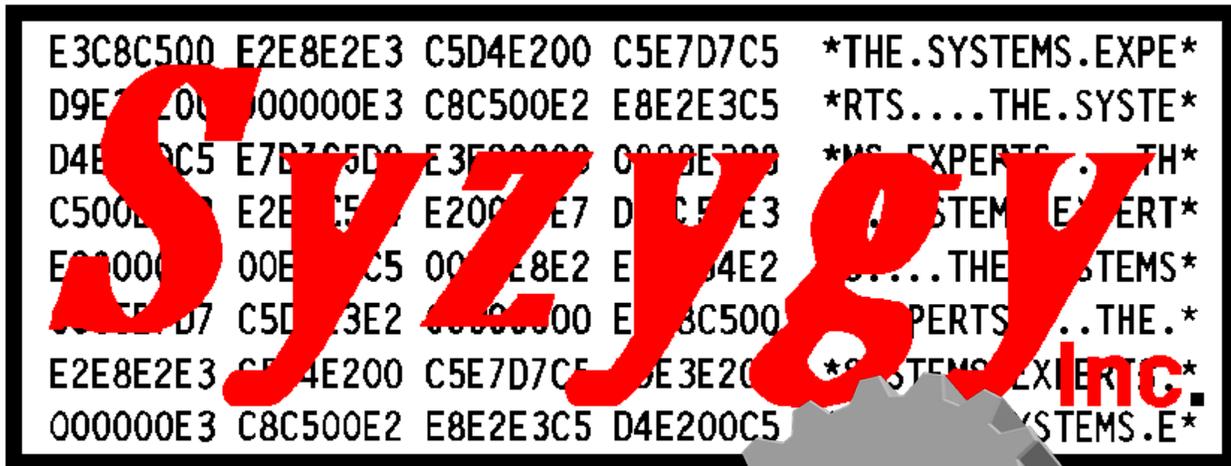
[Installation and User's Guide]

Automatic Event Scheduling. Date and Time based Task and Command Scheduling and Automation System

Revision History

Version	Date	Revision Description
7	10/10/2013	• Main modules re-written to support 64-bit operation. • Every internal "IF" command now has a supported ELSE entry via "	" vertical bar use. • All internal SyzAUTO/z commands are enclosed in Less-Than and Greater-Than characters, i.e. <IFSTARTED=taskname> do something	do something else. • New internal commands added, including SUBMIT, (see this guide). • New IFWEEK1,2,3,4 command which allows the site to issue a command or start a job if the day is (or ELSE is not) within the first, second, third or fourth week of the month (i.e. second Monday of the month would be the Monday of the second week) • New IFLDOM provides support for issuing a command or starting a job only IF it is (or is not) the last day of the month • New IFLWOM (and IFLWEEK) provide support for issuing a command or starting a job if it is (or is not) the last week of the month (i.e. the last Sunday of the month can be handled in this manner) • New IFSYSPLEX internal command provides support for issuing a command or starting a job/task if it is (or is not) queried to be a certain sysplex. • Ability to submit a job/task from within a command line entry without coding the "F stepname,JOBNAME". Now the site can replace the "F stepname" with simply the "SUBMIT" keyword. This means that you no longer need to know the stepname of the SyzAUTO/z you are running under to be able to submit a jobtask from the "SYS1.JOBnn" dataset(s).
6.3	6/23/2013	• Added support for last day of month, first and last week of month processing, as well as second, third and fourth week of month processing to provide more flexibility in control commands.
6.2	2/23/2013	• Minor changes to some commands, better instructions in this manual for using the new options.
6.1	6/23/2012	• Added WARN=Y|N to startup parms to allow site to decide whether or not to show the non-fatal errors that SyzAUTO/z finds during execution.
6.0	12/13/2011	• SyzAUTO/z no longer requires the use of "real" JCL DD's for the allocation of datasets used by the program. All of these datasets can be dynamically allocated via parameters located in the start configuration member of the system parmlib

concatenation. If the site elects not to use the system parmlib concatenation, the a //COMMANDS DD must be used so that SyzAUTO/z can either locate the startup parameters in that library. If all else fails, SyzAUTO/z will attempt to find a dataset called "SYS1.COMMANDS", failing that determination, SyzAUTO/z will stop operation after producing messages that outline the lack of the COMMANDS dataset. Programmed defaults for all settings are located later in this manual.

• New feature – Restart Window. This feature allows the site to stop and start SyzAUTO/z and not lose any scheduled events. Startup settings are available to go back as far as (currently) up to 24 hours to perform all or selected events which were missed while SyzAUTO/z was inactive. In the event of an IPL, when SyzAUTO/z is restarted, it will compute the amount of time that it was inactive and, depending on the settings, perform any missed events.

• New Feature – Command Log. This feature allows the site to create and maintain a detailed log of commands and other scheduled events that SyzAUTO/z has performed. The detail is controlled by startup parameters and is configurable after startup via new operator commands.

• New Feature - Reply to outstanding WTORs. SyzAUTO/z now supports the ability to respond to console requests. The capability exists to respond by number, by messageID, or by the task that issued the request. The response can be any text provided, including variables.

• New Operator Command - SIMulate command comparator. This internal command allows for the simulation of activities which SyzUTO/z performs. It is meant as a testing vehicle only and would normally not be used in actual day to day operations.

• New Operator Command – RESET/REFRESH. When a new event is added to the SyzAUTO/z command PDS, if the event was to take place BEFORE the next already scheduled event, SyzAUTO/z had to be restarted to change the next scheduled event time, but now the RESET/REFRESH command makes that restart unnecessary. RESET/REFRESH causes SyzAUTO/z to recomputed the next scheduled event.

• New feature – comparative command logic. Several new commands have been added to SyzAUTO/z (see detail below for individual commands) so that SyzAUTO/z can perform individual events or commands "only IF" the comparative logic is true. i.e. issue a command only if running on a certain LPAR, or only if a specific task is running (or not running). This exciting new feature is expected to be enhanced over time even more. Currently, only one command operator is allowed per command

line, but there is no limit to the number of command lines in a event member of the //COMMANDS PDS. The comparative command is ended by the use of a semi-colon ";". Any supported command or operation can follow the semi-colon. Preceding spaces are removed (the command is then left-justified).

• New Operator Command – NEXT/PREV. These two operator commands allow the console operator or user to query SyzAUTO/z for the next scheduled event or the previous executed event time.

• New Operator Command – DEBUG. This command causes SyzAUTO/z to start (or stop) detailed debugging messages to the command log dataset. These messages provide a detailed look into what SyzAUTO/z is doing and when/why.

• New Comparative Commands – IFSTARTED and IFSTOPPED. This command is used on the individual command line of the scheduled command within a timed event (or any event) @hhmm member. If used, it will execute the remainder of the command only IF the TASK which is the object of the command is active. i.e.
IFSTARTED=CICS001;F CICS001,CEMT P SHUT

The above command "F CICS001,CEMT P SHUT" will only be executed by SyzAUTO/z if the task "CICS001" is found to be active. Similarly,

IFSTOPPED=CICS001;S CICS001

Will start CICS001 only if it is found that the CICS001 task is NOT active (i.e. is STOPPED).

• New Comparative Command – IFSYSID. This command, when used, will execute the event only IF SyzAUTO/z is executing under the control of the specified SYSID. i.e.

IFSYSID=PROD01;F A,PRD1XA,D=02

The above command to submit batch job "PRD1XA' from the JOB02 dataset will only be performed if the current SYSID that this copy of SyzAUTO/z is running on is "PROD01".

• New Comparative Command – IFLPAR. This command, when used, will execute the event only IF SyzAUTO/z is executing under the named LPAR. i.e.

IFLPAR=TESTA; $S ALL

		will issue the JES "$S ALL" command only IF SyzAUTO/z is running on LPAR "TESTA" • New Comparative Command – WTO. Unlike previous comparative commands, this one does not support a controlled event, it does however allow the site to issue a WTO to the console (up to 110 characters in length). i.e. WTO=This is a test Will issue the WTO "This is a test" to the operator console. • New Comparative Command – WTOH. These two commands are very similar to the previous WTO command except that it will issue the WTO and make it highlighted and have it "stick" to the console so that the operator will see it and have to take action to remove it from the console. • New Comparative Command – SUBMIT. This operator command is similar to the issuance of "F A,jobname" but does not require the site to know the step-name of the SyzAUTO/z task. It is intended for use internally. i.e. SUBMIT=jobname(,D=nn) The above command will submit a job for batch execution, optionally the site can use the ",D=nn" operation to control the JOBnn dataset that SyzAUTO/z is to use to submit the BATCH JOB from. • New //LOG DD has been added for use by SyzAUTO/z. If the site provides the DD, then SyzAUTO/z will use it for the command log file, (if directed to in the startup commands). If the DD is not provided or if it is not identified in the startup, SyzAUTO/z will not keep a command or DEBUG log. • New VSAM Window file. This file us used by SyzAUTO/z to control the windowing feature(s) of the product. It is VERY small (less than 1 track of DASD is required) and is only accessed when SyzAUTO/z performs event work. This file may be read by other facilities or site programs, but should not be written to, except under control of SyzAUTO/z. If the file is not defined, Windowing will not be performed, or possible. The increase in resources used by SyzAUTO/z to maintain this file has been measured to be 1.4 extra seconds of CPU time per MONTH of operation on a heavily used (over 5,ooo SyzAUTO/z commands/day) site.
5.0	5/23/2011	• SyzAuto/z now allows you to keep the startup configuration member in your system parmlib concatenation. This will allow access to the startup parameters from any point in the execution process for SyzAuto/z and for the planned

		additions to Version 5.x which will allow for a new series of updates and enhancements.
		
		• The startup configuration member (default is @AUTO00) can now exist in "SYS1.PARMLIB" or any library in the system parmlib concatenation. You can override the suffix (00) with any other characters via the SyzAuto/z execution parm of PARM='ID=xx'. You can also override the entire configuration parm name via PARM=@AUTO01 or any other parameter name. To exist in the parmlib concatenation, then configuration parameter name must start with "@AUTO" and can have any other 1, 2 or 3 characters following it. You can use other configuration names (aside from @AUTOxx, but they cannot exist in the parmlib concatenation, they will be searched for in the local SyzAuto/z command library (along with the @hhmm members).
		
		• The search order for the startup parameters is:
		    1) System Parmlib Concatenation
		    2) Local SyzAuto/z command library
		    3) pre-programmed defaults of SyzAuto/z

Table of Contents

Contents

1. Introduction

SyzAUTO/z is part of the Syzygy Automation suite of products. The Syzygy Automation Suite provides z/OS sites with a complete and comprehensive capability to automate the entire processor complex. The Suite is made up of 4 main products and one new addition as of 2012;

SyzAUTO/z, the command and task scheduling facility, which allows the site to schedule any system or subsystem command, submit batch JOBs and control timed operations on a 24x7 basis.

SyzCMD/z, the Command Scripting facility, which allows the site to start automated scripts to perform automated and complex functions with complete nested IF/THEN/ELSE logic, allowing the site to control all simple and complex operations in an orderly and efficient manner.

SyzSPOOL/z, the JES SPOOL maintenance facility, which offloads spool data from JES2 or JES3 to separately addressable data sets, and allows users to access the offloaded output interactively from TSO/ISPF or via the Web from any standard web browser. Full security is maintained for access to the offloaded spool data, which is able to be controlled via any storage manager (HSM, ABR), or via SyzSPOOL's own internal dataset allocation control methods. The Data may be viewed, sent via FTP or via Email in any of several output formats including PDF, WORD, HTML, XML and several others.

SyzMPF/z, the Console Message Processing Facility, allows the site to respond to any Console-type message with predefined scripts allowing complete interactive control of any console situation or message. SyzMPF/z allows answering any request or responding to any console based event, (job end, start, abend, message from any job or task, etc.). In short, any console event or message can be handled by the facilities of SyzMPF/z.

SyzNotify/z, the end of task notification facility. SyzNotify/z captures the maximum condition code, plus all (or any) of the Step condition codes, plus other task related information (execution time, CPU time used, I/O's etc.) and sends them to any Email address or destination. User's no longer have to logon to the system to see if a task completed and what it's condition codes were, they can be notified via email, which can be encrypted and delivered directly from the mainframe.

Purpose

The products that make up the Syzygy Automation Suite are designed to offload the burden of supporting the normal day to day operation of the computer center. The facilities allow the site to spend their time and money on things that really matter, instead of maintaining the same old processes which can be automated by the components of the Automation Suite. The site will conserve manpower, and resources.

Related Documents

SyzAUTO/z   Installation and Operations Guide

SyzCMD/z   Installation and Operations Guide (this guide)

SyzSPOOL/z   Installation and Operations Guide

SyzMPF/z   Installation and Operations Guide

SyzNotify/z   Installation and Operations Guide

Conventions

Where present, z/OS, OS/390 and MVS may be used interchangeably. It is not meant that these products are exactly the same, but they are sufficiently alike that for the purposes of this

manual, they can be thought of as the same family with similar or identical support constraints with respect to the products covered in this manual.

Problem Reporting Instructions

Problems should be reported to:

ClientSupport@SyzygyInc.com

Problems with Beta versions of this product should be reported to:

Beta@SyzygyInc.com

2.   Overview

The SyzAUTO/z utility is designed to provide an easy and intuitive way of presenting Batch JOBs and Started Tasks, as well as any z/OS(MVS), JES (2 or 3) or sub-system (like USS, DF/SMS, CICS, DB/2, IMS, Oracle, Adabas, etc.) command to the operating system via a simple and easy to use automated method. SyzAUTO/z provides an easy way of presenting multiple operator commands, Batch JOBs or Started Tasks to be processed/performed via preset time-oriented operation or simple operator modify commands. SyzAUTO/z provides the ability to submit any JOB or enter any operator console-type command, or any number of commands utilizing easy to set up time-of-day/day-of-week entries which can be used to issue automatic commands and submit batch JOB or Started Tasks.

This system function routine, designed and written originally in 1980 and upgraded and greatly enhanced since that time, provides for job submission and automatic commands in a z/OS environment.

SyzAUTO/z's main functions are three-fold, extremely powerful, and can be used in conjunction with other system and sub-system utilities as explained later in this manual. The main functions are:

1)   Allow an easy way of presenting jobs to JES2 or JES3 via a simple and easy to use operator modify command or automatically via time-of-day/day-of-week processing.

2)   Allow an easy way of presenting multiple commands to z/OS, JES2, JES3 or any sub-system (like DB/2) or JOB via a simple and easy operator modify command.

3)   Allow for time-of-day/day-of-week automatic commands and JOB scheduling.

Functional Description

SyzAUTO/z is meant to be run as a separate Batch JOB or (suggested) Started Task.   It is possible to run as a called program from other programs (i.e. from within a COBOL program), but the main intention of SyzAUTO/z is to run as a separate task or step. SyzAUTO/z is composed of several 24 bit, 31bit and 64bit subroutines (CSECTS) generated into a single load module, which must be executed from a "APF authorized" library. Further, many of the functions of SyzAUTO/z are protected by calls to the sites Security system (RACF, ACF/2 Top

Secret, etc.), and care should be taken to provide necessary security system authorization to SyzAUTO/z so that it can monitor and control users, subsystems and tasks as well as submit Production JOBs for execution.

Automatically Start any JOB or Task on any schedule. Automatically Issue any command, any time or at any interval.

Time and/or Date based events

Command & Control Requests

Automated JOB and Task Starting and System Command Scheduling

Email JOB/Task Condition Codes to any Email Address(es)

SyzAUTO/z

Activity Log

Mainframe   Mainframe   Mainframe
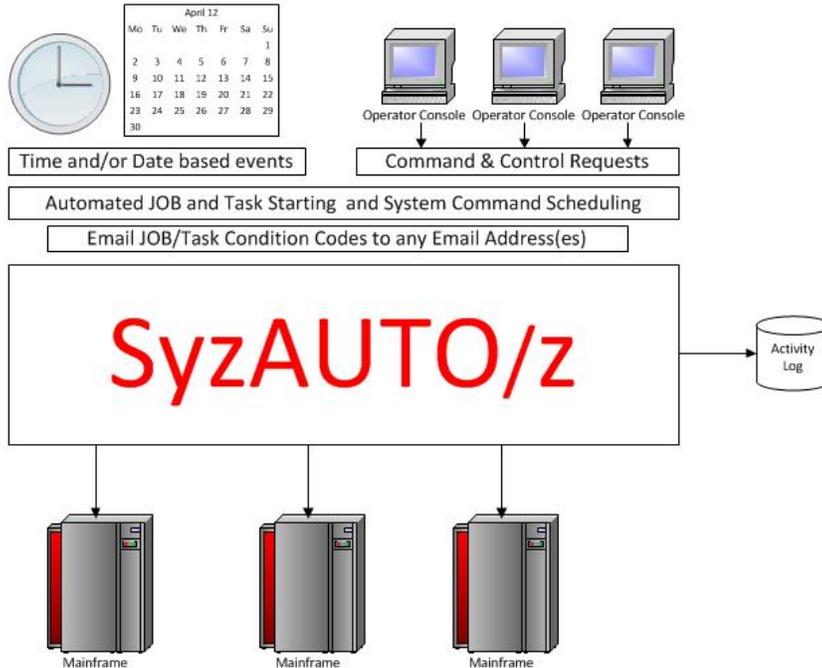
* 24/7 operation
* Built-In IF based Logic Control
* Built-In Restart windowing of missed events
* Built-In SYSPLEX and LPAR aware
* Built-In Simulation Mode
* Built-In automated Reply to WTORs
* Email the condition codes from any JOB or TASK to any Email address

3.   Installation instructions

Procedures

STEP 1:

The software you have ordered is attached to this E-Mail communication as a .ZIP file. You should save or copy this file to a suitable directory on your PC.   The ZIP file will contain 1 load module library, and 2 sample data sets that can be used by SyzAUTO/z. You need not install the sample PDS(es) if you already have them. The two PDS(es) are a sample COMMANDS PDS, which contains sample Time-of-day entries (one for each hour of the day) to which you can add additional entries, and a sample JOBS PDS which contains some sample JOBs that can be used by SyzAUTO/z for testing, you can enlarge or replace these data sets as you wish.

The sample COMMANDS data set contains sample SyzAUTO/z startup JCL in member "$STARTUP".

STEP 2:

Use PKZIP or some other ZIP/UNZIP program to unzip the file to the same directory. You will thereby obtain 3 EBCDIC format files with the following name structure:

Software.datasettype.XMI where "software" is the name of the name of the Licensed Software Product that you have ordered and "datasettype" is either LOAD, COMMANDS and JOBS.

STEP 3:

PRE-ALLOCATE THE <u>THREE</u> RECEIVING XMIT FILES ON YOUR MAINFRAME

   Pre-allocate a file on MVS (e.g. 'userid.software.datasettype.xmit') with the following DCB (use the same attributes for all 3 files).

    Space: 1 Cylinder

    Organization : PS

    Record format : FB

    Record length : 80

    Block size   : 3120

STEP 4:

UPLOAD THE software.datasettype.XMI FILES FROM YOUR PC TO YOUR MAINFRAME.

Using FTP or the file transfer component of your TCP/IP telnet emulator, specifying BINARY transfer, upload the "software.datasettype.XMI" files to your mainframe. You should pre-allocate the destination dataset as outlined in STEP 3 because some mainframe site configurations do not automatically allocate the proper format container for the uploaded XMIT file.
BE SURE TO USE BINARY FILE TRANSFER, DO NOT USE ASCII TRANSLATION.

STEP 5:

RECEIVE THE XMIT FILE TO CREATE THE LOAD MODULE PDS

   On your MAINFRAME perform a RECEIVE command on the uploaded XMIT load module file from step 4 as follows:

(*note If operating under ISPF use panel option 6, or exit ISPF to TSO READY state.)

RECEIVE INDATASET('userid.software.load.xmit')

***where 'userid.software.load.xmit' is the pre-allocated file in STEP 3.

After doing the above command you will be prompted with something similar to the following:

INMR901I Dataset DATA.SET.NAME from userid on ????????
INMR906A Enter restore parameters or 'DELETE' or 'END' +

at this prompt you should enter:

DA('userid.software.loadlib')

... where "userid.software.loadlib" will be a new load module PDS that you have selected which will contain all of the modules required for the correct operation of the licensed software that you have ordered. Do not pre-allocate this dataset.

PLEASE BE SURE TO CHOOSE A NAME THAT DOES NOT ALREADY EXIST AND WHICH WILL BE RACF ACCESSABLE BY YOU. We suggest that you use your TSO userid as the HLQ (High Level Qualifier) to assure RACF permission.

*Note: Some mainframe sites will default to a PUBLIC volume if DF/SMS is not set up properly. If your site falls into this category, you might want to use the following format of the response to keep your dataset from being allocated to your WORK volumes and being scratched before you are ready:

DA('userid.software.loadlib') VOL(volser) UNIT(unitname)

*** where "volser" is a DASD volume at your site (e.g. TSO001) and "unitname" is
the esoteric unitname which governs that volser (e.g. SYSALLDA or 3390).

STEP6:

***REPLACE THE 30-DAY VERSION OF YOUR LICENSED SOFTWARE

You must copy the load module(s) shipped and received by you in the previous steps to the
library that currently contains your 30-day trial version of the licensed software. You do not
have to use the same library, but you will need to delete the old 30-day trial version of the
software if you do not use the same library or else the software will cease to function when the
30-day trial period ends. In some cases the load module(s) shipped will need to be placed into
an "APF AUTHORIZED" library. If APF authorization is required it will be noted in the
identification section(s) above. Normally it is always safe to copy the load module(s) to the
same library as your 30-day trial versions so that no system or user JCL members will need to
be updated.

STEP7 (optional):

optionally, repeat steps 3 through 5 above for the COMMAND.XMIT and JOBS.XMIT members
that were included within your ZIP file.

If you decide not to use the sample files for the COMMANDS and JOBS data sets, you will need
to create them for yourself manually. The data set specifications are as follows:

COMMANDS Data Set DCB:
Space: 10 Cylinders
Organization :   PS
Record format :   FB

Record length :   80

Block size   :   3120

Directory Blocks : 50

Suggested DSN: "SYS1.COMMANDS"

     JOBS Data Set DCB:

Space: 50 Cylinders   (Supports about 4,000 JOBs)

Organization :   PS

Record format :   FB

Record length :   80

Block size   :   3120

Directory Blocks : 250

Suggested DSN: "SYS1.JOBS"

STEP8 (optional, but highly suggested):

Optionally, create a startup configuration member in "SYS1.PARMLIB" (or any other parmlib in the system parmlib concatenation). The default name is **@AUTO00** (00=ZeroZero). A sample is delivered in the Sample Commands dataset created in step 7 above (named @AUTO). You can make any local changed that you deem necessary and copy this entire member to your system parmlib concatenation. If you name the member @AUTO00, no SyzAUTO startup execution parameter is necessary because SyzAuto/z defaults to @AUTO00 for the search. The default member name is @AUTO00 (zero-zero). You can override that default at run time, (see Control Commands), and there are several supported parameters which are outlined in the PARMLIB Startup Member Parameters section below. Any information which is not understood to be a valid parameter is ignored, thus to have comments in the member, you "should" use a "*" (asterisk) in column 1 of the line, and while it is not absolutely necessary, it would be a good idea because at some future time the restrictions on "invalid" parameters might be implemented.

STEP9:

Create a WINDOW(ing) VSAM file. A sample job is provided in the installation PDS with the name "WINDOWS". This will allocate a very small file, only 1 block is necessary (or 1 track if your site requires a minimum). This file is used to allow SyzAUTO/z a place to actively maintain its past, present and future plans for execution, as well as the YTD totals for various fields. If you decide **not** to create this file and provide its location via a //WINDOWS DD in the startup JCL or via the WINDSN startup parameter, then **no** windowing capability will exist, and many query commands and YTD totals will not be possible.

```
//*--------SAMPLE----------------------------------------------
//* BEFORE RUNNING:
//*   1) CHANGE VOLUME
//*   2) CHANGE DSN (or not)
//*   3) CHANGE dataset sizes if desired (Data and Index 1 track minimum)
//*------------------------------------------------------------
//IDCAMS    EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
   DELETE (SYS1.SYZAUTO.WINDOW) PURGE CLUSTER

  DEF CLUSTER (NAME(SYS1.SYZAUTO.WINDOW) -
   INDEXED -
   VOL(volser) -              <---MUST change
   SHR(2 3) -
   KEYS(2 0) -
   RECSZ(150 150) -
   FSPC(10 10) -
   SPEED ) -
  DATA (NAME(SYS1.SYZAUTO.WINDOW.DATA) CYL(1,2) CISZ(4096)) -
  INDEX (NAME(SYS1.SYZAUTO.WINDOW.INDEX) CYL(1,2) )
/*
```

Sample JCL and Possible RACF Changes.

To utilize SyzAUTO/z to it's full potential the user must provide a started task procedure in (normally) SYS1.PROCLIB which is normally started at IPL time. Once active, it will process user defined commands by minute, hour, day of week, month, year or any combination of these. Also, SyzAUTO/z will support operator modify commands (on demand) to either submit JOBs (known as MEMBERS) to the Internal Reader or execute Commands (Member Lists)
For the job submission feature, one or more partitioned dataset(s) are used. For the command/auto-command functions, another partitioned dataset is used. Each are described below (by ddname).

DDNAMES   Use and Function

INTRDR   Optional. This DD is used for JOB submission. It may be allocated in JCL or can

be defined via the startup configuration member, which will create it via dynamic

allocation. If not defined in the startup configuration member, it will be

dynamically allocated to SYSOUT=(A,INTRDR).   It should specify the JES2

internal reader.

JOB00   Optional, required if the job-submission function is to be used. It should specify

the name of a partitioned dataset which contains members to be submitted to

JES2. This dataset may be allocated dynamically if desired via parameters in

the startup configuration member. This partitioned dataset should be RACF

protected and limited to only those personnel who have a need to access it. It is

not normally a place for users or applications programmers to commonly have access.

JOB??   Optional. These datasets may be allocated dynamically via the startup parameter member of the parmlib concatenation. When the job-submission function is used any number of additional 'job??' ddcards may be used to allow for specialized function (with duplicate member names if necessary). This option is similar to the multiple 'proc??' DD cards within the jes2 startup proc. Use of these optional libraries is controlled by the operator (via the 'd=??' keyword, as detailed below).

COMMANDS REQUIRED, but can be optionally defined in the startup parameter member that is located in the system parmlib concatenation. If not located in the concatenation, then it MUST be supplied in the JCL, otherwise SyzAUTO/z will NOT be able to locate its startup parms and the programmed defaults will try to allocated "SYS1.COMMANDS". This DD statement is required for either the operator command-list or the auto-command function. The members in the library pointed to by this DD can have any member name (up to 8 characters), and there are "special" names used to designate Time-of-day functions. It should contain members (lists of commands) in the format described under the heading 'command record syntax' below.

Sample STARTUP JCL.

The following is an example of the type of started procedure that needs to be added to your SYS1.PROCLIB. Note that the 'JOB00 DDNAME is required to use the job submission function while 'job01' and 'job02' are optional (used in this case as remote job control libraries)

```
//AUTO     EXEC PGM=SYZAUTO,TIME=1440          <—— Time 1440 suggested

//STEPLIB      DD DISP=SHR,DSN=SYS1.AUTH.LIBRARY   <—— AUTHORIZED

//INTRDR       DD SYSOUT=(A,INTRDR)               <—— For job submission

//COMMANDS  DD DISP=SHR,DSN=SYS1.COMMANDS       <—— Location of time/cmd packages

//JOB00        DD DISP=SHR,DSN=SYS1.JOBS          <—— Location of BATCH jobs to submit

//JOB01        DD DISP=SHR,DSN=RMT1.JCL.LIB

//JOB02        DD DISP=SHR,DSN=RMT2.JCL.LIB
```

Use of the optional 'job??' DDnames is specified by the operator using the 'D=' OPERAND (detailed later). The following are examples of normal operator commands:

4.   Reference

PARMLIB Startup Member Parameters:

SAMPLE @AUTO00 'SYS1.PARMLIB' member

```
* ("*" in CC1 makes entire line a comment)
* This is the V6.x SYZAUTO startup parm member
*
* CMDDSN=Data.Set.Name          DSN where the @HHMM and scripts are kept
*
* CONFIRM=Y | n                 Confirm delivery of Command packages
*                                     Default is Y
*
* DEBUG=y | N                   Detail DEBUG MODE to the COMMAND log
*                                     Default is N
*
* ECHO=y | N,                   ECHO commands to the console
*                                     Default is N
*
* JOB00=SYS1.JOBS               First JOB submission DSN (required)
*
* JOBnn=JOBS.PDS.DATA           Alternate JOB submit DSN's (optional)
*                                     Max of 00-09 (10 datasets)
*
* LOGCLASS=A                    JOB class for LOG dataset (default A)
*
* LOGDSN=SYS1.SYZAUTO.LOG       DSN to use for LOG (if not SYSOUT)
*                                     Required if SyzAUTO/z is SUB=MSTR
*                                     RECFM=FBA LRECL=133 BLKSIZE=any
*
* RDRCLASS=A                    JES JOB Class that all submitted JOBs
*                                     default is A
*
* STARTUP=membername            The initial Command script to run at
*                                     SyzAUTO/z startup time.
*
* WARN=Y | n                    Inform of not critical messages
*                                     Default is Y
*
* WINBLANK=y | N                Treat Blank Windowing field as if set ON
*                                     Default is N
*
* WINDOW=y | n                  Windowing will be supported
*                                     Default is N
*
* WINDSN=data.set.name          Name of VSAM KSDS Windowing dataset
```

```
*                                        Sample DEFINE in INSTALL(WINDOW)
*
DEBUG=Yes
CMDDSN=SYS1.COMMANDS
LOGDSN=SYS1.SYZAUTO.LOG
*LOGCLASS=C
RDRCLASS=C
ECHO=Yes
STARTUP=STARTUP
JOB00=SYS1.JOBS
*WARN=N
WINDOW=Y
WINDSN=sys1.syzauto.window
```

CMDDSN=   dataset name

The CMDDSN parameter entry allows the site to specify the COMMANDS dataset, which is a PDS, to dynamically allocate via the startup configuration. This entry is ignored if a //COMMANDS DD is included in the SyzAUTO/z Startup JCL. This statement is used in instances where you would like to exclude the inclusion of the //COMMANDS DD in the SyzAuto/z startup JCL

Parameters:
          Datasetname   =   COMMANDS dataset (PDS)

Default   SyzAUTO/z uses the startup JCL "//COMMANDS" if available; Otherwise no (if this paramater is not found) , it will attempt to dynamically allocate SYS1.COMMANDS. Failing that allocation, SyzAUTO/z will fail.

Example:
....
*
CONFIRM=Yes
STARTUP=IPL
ECHO=No
PREFIX=++
CMDDSN=SYZAUTO.COMMANDS.PDS
JOB00=SYS2.TEST.JOBS
.....

CONFIRM=   Y/N

The CONFIRM parameter entry is used to allow SyzAUTO/z to present a "confirmation" message on the console and SyzAUTO/z log to confirm when execution of a command member or JOB member has started and completed processing by SyzAUTO/z. At the end of this point in processing SyzAUTO/z is ready to move to the next task. This feature is necessary to support the Windowing support for restarting SyzAUTO/z after a system outage as planned for Version 5.0 of SyzAUTO/z, but whether or not you wish to have these messages presented on the Syslog and/or Operators Console is a site determination.

Parameters:
                    Yes   =   YES
                    No   =   NO

Default   Yes

Example:
....
*
CONFIRM=Yes
STARTUP=STARTME
ECHO=No
PREFIX=++

.....

DEBUG= Y/<u>N</u>

The DEBUG parameter entry is used to control the ability for SyzAUTO/z to send detailed debugging messages to the //LOG dataset.

Parameters:
Yes	=	YES
No	=	NO

Default   No

Example:
```
........
*
CONFIRM=Yes
STARTUP=STARTME
DEBUG=No
PREFIX=++
```

.....

ECHO= Y/<u>N</u>

The ECHO parameter entry is used to control the ability for SyzAUTO/z to "echo" the commands it issues to the Operator Console. These commands are already issued and logged, but sites have requested that the ECHO capability be added so that system automation might have a chance to monitor or alter processing based on commands issued by SyzAUTO/z.

Parameters:

Yes	=	YES
No	=	NO

Default   No

Example:
....
*
CONFIRM=Yes
STARTUP=STARTME
ECHO=No
PREFIX=++

.....

JOBnn=   dataset name

The JOBnn parameter entry allows the site to specify the JOB submission libraries (JOB00 through JOB09) that are used by SyzAUTO/z to submit batch jobs to the system. (See the D=nn parameter for instructions on how to use alternate JOB submission datasets for addition information). The specified datasets are dynamically allocated to the JOB00 through JOB09 DD's as required. The default, if not specified used is still "SYS1.JOBS" being allocated to JOB00.

Parameters:

nn	=	numbers "00" through "09"
Datasetname	=	Any valid partitioned dataset to submit jobs

from

Default   JOB00 is allocated to "SYS1.JOBS"

Example:
....
*
CONFIRM=Yes
STARTUP=IPL
ECHO=No
PREFIX=++
* Set JOB00 through 03
JOB00=SYS1.PROD.JOBS
JOB01=SYS2.TEST.JOBS
JOB03=SYS2.RECOVERY.JOBS
.....

LOGCLASS= 'class'

Class is a single byte and can be any valid SYSOUT class

The LOGCLASS parameter entry is used to provide a SYSOUT class for dynamic allocation of the //LOG file should the site decide that it does not want to use the //LOG JCL statement of the startup. The log is used by SyzAuto/z to echo commands, send messages and (if DEBUG is turned on) to send detailed debugging information to SYSOUT.

Parameters:
class

Default class "A"

Example:
....
*
CONFIRM=Yes
STARTUP=STARTME
ECHO=No
PREFIX=++
LOGCLASS=F
LOGDSN=SYZAUTO.CMDLOG

.....

LOGDSN= dataset name

The LOGDSN parameter entry allows the site to specify the System LOGGING dataset (a SEQUENTIAL FB=133 dataset) to be used for SyzAuto/z. This log is used to implement enhanced DEBUG features to SyzAuto/z and logging of normal activities. This dataset can be a SYSOUT dataset, and if set up as such then the name is not used, instead the LOGCLASS parameter would apply. This statement is used in instances where you would like to exclude the inclusion of the //LOG DD in the SyzAuto/z startup JCL. In instances where you would like to execute SyzAUTO/z under control of the master scheduler, (instead of JES2/3) you would need to specify a dataset with this (or in the startup JCL) because the master scheduler does not support SYSOUT datasets.

Parameters:
               Datasetname   =   Valid Sequential Log file

Default   SyzAUTO/z uses the startup JCL "//LOG" if available; Otherwise no LOG dataset is available and no features that require the allocation of that dataset are available.

Example:
....
*
CONFIRM=Yes
STARTUP=IPL
ECHO=No
PREFIX=++
LOGDSN=SYZAUTO.LOG
JOB00=SYS2.TEST.JOBS
.....

PREFIX=   xx

The PREFIX parameter entry can be used in the @AUTOxx startup parameter list to set a "prefix" for the new internal commands which can be used with SyzAUTO/z. These new commands provide quick access to features which were previously only available with the SyzCMD/z or SyzMPF/z products. Long involved scripts such as are used by SyzCMD/z or SyzMPF/z are not supported, but simple logic is now within the capability of SyzAUTO/z and this command allows the site to change the prefix to any two characters that they wish. The default is "++", but since it's possible that some sites may already be using that prefix for actual commands (i.e. DB/2), we made the ability to change that prefix available to the site. The PREFIX=xx command is available within the @AUTOxx startup member, and dynamically in both the command members themselves and via an Operator Console modify command.

Parameters:

        XX   =   Any two characters (except blank and/or – (minus))

Default   ++   (Plus sign, Plus Sign)

Example:
....
*
CONFIRM=Yes
STARTUP=STARTME
ECHO=No
PREFIX=++

.....

RDRCLASS= 'class' | <u>A</u>

Class is a single byte and can be any valid SYSOUT class

The RDRCLASS parameter entry is used to provide a class for dynamic allocation of the internal reader //INTRDR file should the site decide that it does not want to use the //INTRDR JCL statement of the startup. The class used here will be the "default" job execution class of the submitted jobs, if they don't specify one of their own.

Parameters:
> class

Default   class "A"

Example:
....
*
CONFIRM=Yes
STARTUP=STARTME
ECHO=No
PREFIX=++
RDRCLASS=C
LOGDSN=SYZAUTO.CMDLOG

.....

STARTUP=   member

The STARTUP parameter entry allows the site to specify the first member to be loaded at startup of SyzAUTO/z. This facility is to maintain the capability of the old EXEC parm "PARM=member" to have SyzAUTO/z run an initial command member upon startup of SyzAUTO/z. The member can be any existing member of the "COMMANDS" dataset.

Parameters:
> Member   =   Any member of the COMMANDS dataset
> No   =   NO

Default   no member will be automatically executed

Example:
....
*
CONFIRM=Yes
STARTUP=IPL   <---- Will execute the "IPL" member of the COMMANDS dataset.
ECHO=No
PREFIX=++

.....

WARN= Y/<u>N</u>

The WARN parameter entry is used to control the ability for SyzAUTO/z to send the console warning messages when some lesser importance events occur. These messages will always be provided to the LOG if allocated.

Parameters:
Yes	=	YES
No	=	NO

Default   No

Example:
....
*
CONFIRM=Yes
STARTUP=STARTME
WARN=No
PREFIX=++

.....

WINBLANK= Y/<u>N</u>

The WINBLANK parameter entry is used to tell SyzAUTO/z what to do with the commands in the command packages that contain a blank in column 1 of the command line. The older versions of SyzAUTO/z only used this position to denote that the line was to be treated as a comment. It is now used to denote windowing processing control. This setting will tell SyzAUTO/z how to treat a "BLANK" space as the character that will/can be located in column 1 of the command package lines. Since older versions of SyzAUTO/z did not use that space, it is probable that it will already contain a blank space. This will tell SyzAUTO/z whether to treat that "BLANK" character as the same as a "+" (Plus sign) which means that it should be supported as a windowed command, or a "-" (minus sign) which means that we should treat it as a non-Windowed line. A non-windowed line will not be eligible for windowing support on a restart of SyzAUTO/z.

Parameters:
Yes	=	YES
No	=	NO

Default   No

Example: (tell SyzAUTO/z to not "window" lines with a blank in column 1)
....
*
CONFIRM=Yes
STARTUP=STARTME
WINBLANK=No
PREFIX=++

.....

WINDOW= Y/<u>N</u>

The WINDOW parameter entry is used to control the ability for SyzAUTO/z to utilse the new WINDOW(ing) facility. This facility allows SyzAUTO/z to maintain the past, present and future status of the utility so that if SyzAUTO/z is ended, when it is restarted, it will be able to go "back in time" up to 24 hours to perform any work that it may have missed. To fully support this feature, only lines that specify a "+" (plus sign) will be windowed if this setting is set to YES. Also, if the site has set "WINBLANK=YES", then either a "+" or a blank space in column 1 of the command line will both be treated the same as if it contained a "+". If this option is set Or defaulted to "NO", then column 1 characters are ignored.

Parameters:

Yes	=	YES
No	=	NO

Default   No

Example:
....
*
CONFIRM=Yes
STARTUP=STARTME
WINDOW=No
PREFIX=++

.....

WINDSN=   dataset name

The WINDSN parameter entry allows the site to specify the Startup "WINDOWING" dataset, which is a VSAM KSDS file, to dynamically allocate via the startup configuration. This file is necessary to allow SyzAUTO/z to keep track of it's past, current and future status, so that if stopped and restarted, it can go back in time (up to 24 hours) to perform any tasks that may have been missed. There are special settings to control which command lines of the command package this applies to, (they will contain a "+" in column 15, a "-" in column 1 denotes NEVER to perform that line under windowing, and if a blank in column 1, the WINBLANK setting tells SyzAUTO/z whether to process that line or not). This entry is ignored if a //WINDOWS DD is included in the SyzAUTO/z Startup JCL. This statement is used in instances where you would like to exclude the inclusion of the //WINDOWS DD in the SyzAuto/z startup JCL

Parameters:
        Datasetname   =   WINDOW(ing) VSAM KSDS file.

Default   SyzAUTO/z uses the startup JCL "//WINDOWS" if available; Otherwise, no windowing will be performed and no settings will be saved to provide for future windowing.

Example:
....
*
CONFIRM=Yes
STARTUP=IPL
ECHO=No
PREFIX=++
WINDSN=SYZAUTO.WINDOWS.FILE
JOB00=SYS2.TEST.JOBS
.....

Internal Line Control Commands

Note: All internal line control commands are designed to be used within the actual command areas of the command package lines (starting anywhere at or after column 16 through 71) and can be continued on 1 line. These commands create the capability for the site to include logic with the commands they wish SyzAUTO/z to execute. These control commands make it possible for SyzAUTO/z to have a much more powerful capability to decide which commands should be entered. Previously, if a command matched the date and time, SyzAUTO/z would attempt to issue it, or start the TASK or JOB, but now the site can perform work only "IF" a condition is true (or not true). The Command before the vertical line "|" will be performed if "TRUE" and the command after the vertical line (if it exists) will be performed if the command tested is "FALSE". Also, the site can issue WTO's or WTOR (sticky notes), and can reply to outstanding messages that exist at the time of the execution for that particular command.

        The following are the currently supported ILCC's:

<IFLPAR   lparname>   -   Execute line only if on this LPAR

<IFREPLY TASK|* MSGID|* TEXT>   -   Reply to an outstanding WTOR

<SIMulate>   -   Simulate the command on this line

<IFSTARTED taskname>   -   Execute line only IF this task|job|user is active

<IFSTOPPED taskname>   -   Execute line only IF this task|job|user is NOT active

<IFSYSID   systemname> -   Execute this line only IF on this System

<IFSYSPLEX sysplexID>   -   Execute this line only IF on this SYSPLEX member

<IFWEEKn.>   -   Execute this line only if we are in this week of month

<IFLDOM>   -   Execute this line only if Last Day of Month

<IFLWEEK>   -   Execute this line only if it's the last week of the month

<WTO message>   -   Issue this message as a WTO

<WTOH message>   -   Issue this message as a HIGHLIGHTED WTO

<IFLPAR   lparname   > command if true | command if false

The IFLPAR Internal control command is used to control the execution of the command after the greater than sign "`>`". If SyzAUTO/z is executing on the LPAR named in the lparname area, then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

         Lparname   =   (name of the LPAR you want to verify)
         Command if true   =   the command you want executed if true.
         Command if false   =   the command you want executed if false.

Default   none

Example:

.... @0300 member (3am)
* execute the following line under these conditions:
* 1) all months, all days
* 2) only on weekends
* 3) only if executing on the "SYSA" LPAR
*

```
COL-1---->*MM/DD MTWTFSS COMMAND----------------------------------------->72
          *if lpar SYSA, start CICS cleanup on Saturday AND Sundays
           **/**       SS <IFLPAR SYSA>; S CICSCLN
........
```

<REPLY   taskname*|* MsgID*|* 'reply text'>

The REPLY Internal control command is used to reply to an outstanding WTOR (on the system. SyzAUTO/z has the flexibility with this command to reply to a single task by name (TASKNAME), by generic (TAS*) or any task (*), This is handy for replying to tasks that have an always outstanding reply request, (Netview, IMS, etc.). The message that you wish to reply to can be any length (enclosed in quotes if it contains embedded blanks), and can be generic (i.e. any text, that is issued from the task named in the TASKNAME field. The TASKNAME and MESSAGEID fields cannot BOTH be a single ASTERISK. That would be meaningless since it would mean any message from any task. The final field, the "REPLY TEXT" can be any text you wish to reply to the outstanding message. If that text contains embedded blanks, then it must be enclosed in single quotes

Parameters:

        Taskname(*)|* =   Full or generic taskname or "*" (any task)
        MsgID(*)|*   =   Full message text, partial message text, or "*" any
        "REPLY TEXT" =   The text to reply to the message

Default   none, but TASKNAME and MSGID cannot both be "*"

Example:
.... @0300 member (3am)
* execute the following line under these conditions:
* 1) all months, first day of month
* 2) any day of week
* 3) Do not do this on a windowing restart (column 1 = '-')
* 4) REPLY to nn *DSI802A* CNM01 REPLY WITH VALID NCCF SYSTEM OPERATOR CMD
*
```
COL-1---->*MM/DD MTWTFSS COMMAND---------------------------------------->72
          *Wake up Netview  task is NETVIEW, make sure MSG contains NCCF
          -**/01 MTWTFSS  <REPLY NETV* 'VALID NCCF' WakeUP>
```
....

<SIMULATE>   command

The SIMulate Internal control command is used to simulate the execution of the command after the greater than sign "`>`". If SIMulate is specified, then the command is not actually executed, but is displayed on the console and in SYSLOG with a "Simulating: _command_ message..

Parameters:

        command   =   the command you want to simulate.

Default   none

Example:

```
........ @0300 member (3am)
* Simulate this command, we can make it real later:
* 1) all months, all days
* 2) only on weekends
*
COL-1---->*MM/DD MTWTFSS COMMAND------------------------------------------->72
          *
           **/**        SS <SIMULATE> S CICSCLN   *start CICS
........
```

<IFSTARTED   Taskname> > command if true | command if false

The IFSTARTED Internal control command is used to control the execution of the command after the greater than sign ">". If the Task (JOB, STC or TSU) is executing at the time the command package is executed, then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

TaskName	=	(name of the Task (job, stc, or tsu you want to be running)
Command if true	=	the command you want executed if true.
Command if false	=	the command you want executed if false.

Default   none

Example:
```
........  @0300 member (3am)
* execute the following line under these conditions:
* 1) all months, all days
* 2) only on weekends
* 3) only if CICSA is running already
*
COL-1---->*MM/DD MTWTFSS COMMAND---------------------------------------->72
          *if CICSA is up, start CICS cleanup on Saturday AND Sundays
           **/**        SS <IFSTARTED CICSA> S CICSCLN
........
```

<IFSTOPPED   Taskname>   command if true | command if false

The IFSTOPPED Internal control command is used to control the execution of the command after the greater than sign "`>`". If the Task (JOB, STC or TSU) is **NOT** executing at the time the command package is executed, then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

TaskName   =   (name of the Task (job, stc, or tsu you want
                NOT to be running)
Command if true   =   the command you want executed if true.
Command if false   =   the command you want executed if false.

Default   none

Example:
.... @0300 member (3am)
* execute the following line under these conditions:
* 1) all months, all days
* 2) only on weekends
* 3) only if CICSA is running already
*
```
COL-1---->*MM/DD MTWTFSS COMMAND---------------------------------------->72
          *if CICSA is stopped, start CICS cleanup on Saturday AND Sundays
           **/**        SS <IFSTOPPED CICSA> S CICSCLN
```
....

<IFSYSID   SystemID> command if true | command if false

The IFSYSID Internal control command is used to control the execution of the command after the greater than sign ">". If SyzAUTO/z is executing on the System ID named in the SyztemID area, then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

        SystemID   =   (name of the System you want to verify)
        Command if true   =   the command you want executed if true.
        Command if false =   the command you want executed if false.

Default   none

Example:
.... @0300 member (3am)
* execute the following line under these conditions:
* 1) all months, all days
* 2) only on weekends
* 3) only if executing on the "PROD" system
*

```
COL-1---->*MM/DD MTWTFSS COMMAND------------------------------------------>72
          *if we are on PROD, start CICS cleanup on Saturday AND Sundays
           **/**      SS <IFSYSID PROD>   S CICSCLN
........
```

<IFSYSPLEX SysPLEXID>command if true | command if false

The IFSYSPLEX Internal control command is used to control the execution of the command after the greater than sign ">". If SyzAUTO/z is executing on the SysPLEX ID named in the SysPLEXID area, then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

SysPLEXID   =   (ID of the Sysplex you want to verify)
Command if true   =   the command you want executed if true.
Command if false   =   the command you want executed if false.

Default   none

Example:
.... @0300 member (3am)
* execute the following line under these conditions:
* 1) all months, all days
* 2) only on weekends
* 3) only if executing on the "41" member of the plex
*
```
COL-1---->*MM/DD MTWTFSS COMMAND-------------------------------------------->72
          *if we are on PROD, start CICS cleanup on Saturday AND Sundays
           **/**      SS <IFSYSPLEX=41>   S CICSCLN
........
```

<IFWEEK1>   command if true | command if false

The IFWEEK1 Internal control command is used to control the execution of the command after the greater than sign "**>**". If SyzAUTO/z is executing on ANY day within the FIRST week of the month (first 7 days of the month), then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

Command if true	=	the command you want executed if true.
Command if false	=	the command you want executed if false.

Default   none

Example:
```
........  @0300 member (3am)
* execute the following line under these conditions:
* 1) First Sunday of the Month (only)
*ELSE
* all other Sundays, execute the data copy job
*
COL-1---->*MM/DD MTWTFSS COMMAND---------------------------------------->72
          * start CICS cleanup on 1st Sunday, else datacopy
           **/**       S <IFWEEK1>   S CICSCLN   | SUBMIT DATACOPY
........
```

<IFWEEK2>   command if true | command if false

The IFWEEK2 Internal control command is used to control the execution of the command after the greater than sign ">". If SyzAUTO/z is executing on ANY day within the SECOND week of the month (Second occurrence of Weekdays/weekend of the month), then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

Command if true	=	the command you want executed if true.
Command if false	=	the command you want executed if false.

Default   none

Example:
```
........  @0300 member (3am)
* execute the following line under these conditions:
* 1) Second Sunday of the Month (only)
*ELSE
* all other Sundays, execute the data copy job
*
COL-1---->*MM/DD MTWTFSS COMMAND------------------------------------------->72
          * start CICS cleanup on 2nd Sunday, else datacopy
           **/**        S <IFWEEK2>   S CICSCLN   | SUBMIT DATACOPY
........
```

<IFWEEK3>   command if true | command if false

The IFWEEK3 Internal control command is used to control the execution of the command after the greater than sign ">". If SyzAUTO/z is executing on ANY day within the THIRD week of the month (Third occurrence of Weekdays/weekend of the month), then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

Command if true	=	the command you want executed if true.
Command if false	=	the command you want executed if false.

Default   none

Example:

```
........  @0300 member (3am)
* execute the following line under these conditions:
* 1) Third Sunday of the Month (only)
*ELSE
* all other Sundays, execute the data copy job
*
COL-1---->*MM/DD MTWTFSS COMMAND-------------------------------------------->72
          * start CICS cleanup on 3rd Sunday, else datacopy
           **/**        S <IFWEEK3>   S CICSCLN   | SUBMIT DATACOPY
........
```

<IFWEEK4>   command if true | command if false

The IFWEEK4 Internal control command is used to control the execution of the command after the greater than sign ">". If SyzAUTO/z is executing on ANY day within the Fourth week of the month (Fourth occurrence of Weekdays/weekend of the month), then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

Command if true	=	the command you want executed if true.
Command if false	=	the command you want executed if false.

Default   none

Example:
.... @0300 member (3am)
* execute the following line under these conditions:
* 1) Fourth Sunday of the Month (only)
*ELSE
* all other Sundays, execute the data copy job
*

```
COL-1---->*MM/DD MTWTFSS COMMAND---------------------------------------->72
          * start CICS cleanup on 4th Sunday, else datacopy
           **/**        S <IFWEEK4>   S CICSCLN   | SUBMIT DATACOPY
........
```

<IFLDOM>   command if true | command if false

The IFLDOM Internal control command is used to control the execution of the command after the greater than sign ">". If SyzAUTO/z is executing on ANY day that is the LAST DAY of that particular MONTH, then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

Command if true	=	the command you want executed if true.
Command if false	=	the command you want executed if false.

Default   none

Example:
.... @0300 member (3am)
* execute the following line under these conditions:
* 1) Last Day of the month (only)
*ELSE
* all other days, execute the data copy job
*
```
COL-1---->*MM/DD MTWTFSS COMMAND------------------------------------------>72
          * start CICS cleanup on 2nd Sunday, else datacopy
           **/** MTWTFSS <IFLDOM>   S CICSCLN   | SUBMIT DATACOPY
```
....

<IFLWEEK> or <IFLWOM>   command if true | command if false

The IFLWEEK Internal control command is used to control the execution of the command after the greater than sign "˃". If SyzAUTO/z is executing on ANY day within the LAST week of the month (LAST occurrence of Weekdays/weekend of the month), then the command is executed, otherwise it is skipped unless the ELSE option (vertical bar) is used in which case the "IF NOT TRUE) condition is raised and that command is issued instead. If skipped, a message is issued on the console or in syslog (if ECHO is "yes").

Parameters:

Command if true	=	the command you want executed if true.
Command if false	=	the command you want executed if false.

Default   none

Example:
.... @0300 member (3am)
* execute the following line under these conditions:
* 1) LAST Sunday of the Month (only)
*ELSE
* all other Sundays, execute the data copy job
*

```
COL-1---->*MM/DD MTWTFSS COMMAND------------------------------------------>72
          *start CICS cleanup on last Sunday, else datacopy
           **/**        S <IFLWEEK>   S CICSCLN   | SUBMIT DATACOPY
........
```

<WTO   message text... >

The WTO Internal control command is used to issue a write to operator of the text that follows the WTO command. The WTO is issued as a non-important message (i.e. rollable) on the console and SYSLOG.

Parameters:
            Message text   =   the text you want to issue.

Default   none

Example:
.... @0300 member (3am)
* tell the operator what we are doing
* Simulate this command, we can make it real later:
* 1) all months, all days
* 2) only on weekends
*
COL-1---->*MM/DD MTWTFSS COMMAND--->72
            *
            /   SS <WTO we are going to start CICS Cleanup now >
            /   SS S CICSCLN   *start CICS
....

<WTOH   message text... >

The WTOH Internal control command is used to issue a HIGHLIGHTED write to operator of the text that follows the WTO command. The WTO is issued as a non-rollable message on the console and SYSLOG.
Parameters:

        Message text   =   the text you want to issue.

Default   none

Example:
.... @0300 member (3am)
* tell the operator what we are doing
* Simulate this command, we can make it real later:
* 1) all months, all days
* 2) only on weekends
*

```
COL-1---->*MM/DD MTWTFSS COMMAND---------------------------------------->72
          *
          **/**       SS <WTOH we are going to start CICS Cleanup now  >
          **/**       SS S CICSCLN   *start CICS
........
```

Operator Control Commands

SyzAUTO/z allows the operator to query SyzAUTO/z to get information on the status of the system, the settings, the next and previous interval and to preform several control functions. These commands can be issued from any valid operator console or from SDSF, or even within SyzAUTO/z itself.

The following Operator Commands are supported:

PREFIX   -   Change or reset the site prefix for all commands

NEXT   -   Query when the next command time is due to occur

PREV   -   Query what the last command time was.

RESET/REFRESH - Reset the product when new command times are entered

DEBUG   -   Turn on (or off) detail tracing of SyzAUTO/z

NODEBUG   -   Turn off detail tracing

WARN   -   Turn on (or off) warning messages from SyzAUTO/z

NOWARN   -   Turn off warning messages

QUERY   -   Display settings and resources used by SyzAUTO/z

STOP   -   Terminate SyzAUTO/z

SUBMIT   -   Submit a Batch JOB from the JOBnn dataset

DEBUG=   Yes|No

NODEBUG

The DEBUG operator command can be used from the operator console (or from within a SyzAUTO/z command member) toturn on (or off) detailed debugging. The debugging information is logged in the SyzAUTO/z LOG dataset or SYSOUT (depending on which is being used). NODEBUG is the same as DEBUG=No. The default is "Yes", i.e DEBUG with no operands is the same as DEBUG=Yes.

Parameters:
            Yes | No   =   Yes or No

Default   Yes

Example of command issued from console for SyzAUTO.A ("A" is the STEPname):
....
Command:

F A,Debug=Y   Modify SYZAUTO.A to turn on detailed DEBUGing messages

Response:

.....

NEXT

The NEXT operator command can be used from the operator console (or from within a SyzAUTO/z command member) to ask SyzAUTO/z to identify the next schedule command package time. There are no parameters to this command.

Parameters:
        None

Default   n/a

Example of command issued from console for SyzAUTO.A ("A" is the STEPname):
....
Command:

F A,NEXT   **Modify SYZAUTO.A to ask what command package will be next**

Response:

.....

PREV

The PREV operator command can be used from the operator console (or from within a SyzAUTO/z command member) to ask SyzAUTO/z to identify the just previous command package that was executed. There are no parameters to this command.

Parameters:
                None

Default   n/a

Example of command issued from console for SyzAUTO.A ("A" is the STEPname):
....
Command:

F A,PREV   Modify SYZAUTO.A to ask what command package we did last

Response:

.....

QUERY   ALL

The QUERY operator command can be used from the operator console (or from within a SyzAUTO/z command member) to query the current counters (default) or the SyzAUTO/z setting ans counters (all).

Parameters:

All	=	Display both the Settings and the counters

Default   None

Example of command issued from console for SyzAUTO.A ("A" is the STEPname):
....
Command:

F A,Query All Modify SYZAUTO.A to display the settings and counters

Response:

.....

RESET

REFRESH

The RESET and (prefix|++)REFRESH operator command can be used from the operator console (or from within a SyzAUTO/z command member) to ask SyzAUTO/z to refresh the schedule of commands. Older versions of SyzAUTO/z needed to be restarted if a command package was added that would have been scheduled before the currently set "next" command package. This was confusing and caused an unnecessary SyzAUTO/z outage, so this command was designed to compensate There are no parameters to this command.

Parameters:
> None

Default   n/a

Example of command issued from console for SyzAUTO.A ("A" is the STEPname):
....
Command:

F A,RESET   Modify SYZAUTO.A to refresh the schedule of command packages

Response:

.....

STOP

The STOP operator command can be used from the operator console (or from within a SyzAUTO/z command member) to ask SyzAUTO/z to shutdown. There are no parameters to this command. (you can still stop SyzAUTO/z via the conventional "STOP" or "P" command.)

Parameters:
> None

Default   n/a

Example of command issued from console for SyzAUTO.A ("A" is the STEPname):
....
Command:

F A,STOP   Shutdown SyzAUTO/z.

Response:

.....

SUBMIT= JOBNAME,jobanme,jobname, D=nn

The SUBMIT operator command can be used from the operator console (or from within a SyzAUTO/z command member) to submit any JOB or JOBs from any of the JOBnn datasets available to SyzAUTO/z.

Parameters:

        Jobname   =   The JOB you wish to submit

        Jobanme,jobname…jobanme   =   Multiple jobs

        D=nn   =   The JOBnn library you want to submit from
                                 (the default is JOB00)

Default   D=00

Example of command issued from console for SyzAUTO.A ("A" is the STEPname):

....

Command:

F A,SUBMIT=BACKUPS,D=03 Modify SYZAUTO.A to submit BACKUPS from //JOB03

Response:

.....

WARN=   Yes|No

NOWARN

The WARN operator command can be used from the operator console (or from within a SyzAUTO/z command member) toturn on (or off) detailed Warning WTO's that SyzAUTO/z can issue under some circumstances. For instance, if you have added a @hhmm member that is incorrect (i.e. "@2305old"), SyzAUTO/z would normally issue a warning that this member is not valid for Time of Day processing. WARN=No would keep these messages from being produced. NOWARN is the same as WARN=No. The default is "Yes", i.e WARN with no operands is the same as WARN=Yes.

Parameters:
            Yes | No   =   Yes or No

Default   Yes

Example of command issued from console for SyzAUTO.A ("A" is the STEPname):
....
Command:

F A,Warn=No   Modify SYZAUTO.A to turn off Warning WTO messages

Response:

.....

Quick Start Guide

Q: How do I start SyzAUTO/z?

A: First Choice: With The COMMNDxx member of SYS1.PARMLIB

> **COM='S SYZAUTOZ.A,PARM=IPL'**
> Starts SyzAUTO/z and executes the "IPL" member at startup.

> Second Choice: With the operator console command

> **S SYZAUTOZ.A,PARM=STARTME**
> Starts SyzAUTO/z and executes the "STARTME" member at startup

Using the SYS1.PARMLIB member to start SyzAUTO/z at IPL is the "normal" way to start the utility. The first automatic command to be processed will be the member specified on the PARM= keyword of the JCL, or the STARTUP= parmameter of the @AUTOxx startup parm member, or both, if specified. If nothing is specified, the first COMMANDS dataset member to be processed will be the first @nnnn member (where nnnn is the time-of day) with the time greater than the **current** time of day. However, demand command execution as well as JOB submission is available immediately. The purpose of the ".A" at startup following the "S SYZAUTOZ" is two-fold, first, it allows you to have multiple AUTO sub-systems active at one time, i.e. AUTO.A, AUTO.B, AUTO.C, etc, and address them easily. Second, it give you an easy "shorthand" way to ask AUTO to do something. You could call AUTO anything you want, but it becomes hard for the operators to easily issue commands if you make the name difficult, so calling it "STC1234B" is probably not a good idea, but by using the ".A", you could call it "STC1234B.A" and still be able to get to the task via a shorthand "A". This is a feature and a limitation of MVS, not AUTO.

Q: How do I have SyzAUTO/z "AUTOIPL" my system?

A: See previous answer.

Actually this is the same as the standard start format except that SyzAUTO/z is given a command member to process upon startup in the PARM= field. This command list "IPL" will be processed before any other requests (Automatic or Operator requested commands). This provides an alternative to the "SYS1.PARMLIB(COMMNDnn)" method provided by MVS. It also provides for any type of command (JES2 or MVS commands as well as other subsystem

commands which may be necessary). You "can" autoipl your system this way, but you are limited in the types of commands that can be entered. There is no real "LOGIC" in what is accomplished. SyzCMD/z is the product which handles truly AUTOMATIC z/OS IPLs.

Q: How do I Submit a JOB with SyzAUTO/z?

A: Assuming you have started your SyzAUTO/z task with a stepname of "A", you could:

F A,jobname   (Jobname is any job contained in the JOB00 DSN)

To submit the JOB "MYJOB1" to the JES2 internal reader you would specify "F A,MYJOB1". Since the "D=" option was not used, the member will be read from the partitioned dataset pointed to by the JOB00 DD card.

*Note Any JOB started by SyzAUTO/z will run under the RACF attributes of SyzAUTO/z as a surrogate JOB, therefore SyzAUTO/z should be given the adequate RACF authority to access any dataset in the system. One more reason why access to SyzAUTO/z's datasets should be strictly controlled.

Q: Can I submit multiple jobs at a time?

A: Yes, you can submit as many jobs as can be fit on the command line of the console (or in the command field of the SyzAUTO/z command member you are executing) separated by commas.

F A,MYJOB1,MYJOB2,MYJOB3,... etc.

Q: I have multiple JOBxx datasets specified in my SyzAUTO/z startup JCL, can I submit JOBs directly from those datasets dynamically?

A: Yes, you merely need to specify the last 2 characters of the JOBxx Ddname.

F A,MYJOB1,D=03

This would submit a member (MYJOB1) to the JES2 internal reader. Since "D=03" was specified, the JOB00 DD will not be searched, instead the JOB00 DD is closed and the JOB03 DD is opened to read the member. Any number of libraries can be access in this way. This facility allows you to have duplicate names in the JOBnn libraries and submit them from the alternate locations.

Q: What are the Dynamic Operator Command Functions?

A: SyzAUTO/z has the capability of issuing commands as well as submitting JOBs. These commands can be any valid system command, JES, DB/2, MVS subsystem commands, in fact, any command you can normally issue frm the operator console can be issued from SyzAUTO/z.   The following are examples of the operator commands required to schedule "Command-Lists", which are series of commands to be executed in a row. If it is necessary to use structured logic to execute a long list of commands, then the optional companion utility, SyzCMD/z, should be used in place of SYZAUTO/z. SYZAUTO/z should still be used to "schedule" the SyzCMD/z controlled script, but the facilities available with SyzCMD/z far exceed those available, or appropriate, to a utility like SYZAUTO/z, such as IF/THEN/ELSE logic, the ability to issue commands ONLY-IF some other system task or user task is (or is not) active, and other special function logic.

Q: How do I tell SyzAUTO/z to issue a list of commands on demand?

A: To causes a list of commands to be executed by MVS or JES2 or any other subsystem. You issue the MVS modify command to SyzAUTO/z and place a DASH (hyphen) as a prefix to the member you wish to invoke.

F A,-membername

This identifies the request to SyzAUTO/z as a command request. The named member will be read from the //COMMAND DD. Each command read from that member will be executed if it applies to the restrictions specified internally in the command member. This will make sense when we get into the structure of the members later on.

Q: Can I invoke multiple command members at a time?

A: Yes, you can invoke as many command members from the COMMAND DD as can be fit on the command line of the console (or in the command field of the SyzAUTO/z command member you are executing) separated by commas.

F A,-STRTCICS,STRTDB21,STRTDB22,... etc.

Q: How can I stop SyzAUTO/z?

A   You can stop SyzAUTO/z in one of two ways:

P   A   (where "A" is the stepname of SysAUTO/z)
F A,STOP   (where "A" is hte stepname of SyzAUTO/z)

Q: Can you tell me about how a @nnnn time command works?

A: Yes. Automatic command functions are not controlled by the operator with SysAUTO/z. Instead, SyzAUTO/z operates on a 1440 (daily) cycle of 60 minutes per hour and is active and responsive 24 hours per day. SyzAUTO/z is able to issue commands and start JOBs at any specified time of day on minute of any day, within day-of-week or month and year. It can issue commands on daily, weekly, monthly or yearly cycles. Commands to be executed are contained in members that reside in the //COMMAND DD library. Each member to be automatically processed (at some specific time of day) is named as follows:

@0100 This member contains commands to be processed at 01:00 in the morning.

@1423 This member will be processed at 2:23 PM.

@0000 This member will be processed at midnight

*Note Member names to be processed automatically (by time of day) must be prefixed with the "AT SIGN" (@), however, these members are not restricted from "demand" use, so the operator may decide that the midnight processing can be done now just by specifying the on demand "F A,-@0000" command.

Each "automatic" command member in the COMMANDS library must follow the syntax defined below, this is true of both DEMAND command lists as well as AUTOMATIC command lists. The difference between the two is that the member name of the AUTOMATIC commands MUST start with an at-sign "@", and demand members may start with any alpha-numeric or ebcdic standard character that is acceptable for naming members of a PDS.

Format of the MEMBERS of the COMMAND PDS

An Asterisk (*) in column one denotes that the entire line is a comment. Comments are ignored by the SyzAUTO/z processor, and it is encouraged that you use them liberally so that you can make notes to yourself and others as to why and what the commands are that you are performing.

Commands (lines of each PDS member that makes up the command entry) are processed only on the specified days of the week as indicated in the DAYS columns and are in the order of M-T-W-T-F-S-S, (**M**onday-**T**uesday-**W**ednesday-**T**hursday-**F**riday-**S**aturday-**S**unday).

Note: There are no dashes between the days of the week in the PDS member, they are placed here just to make it easier to read.
Any NON_BLANK character in the DAY-OF-THE-WEEK fields requests processing on that day.

        FORMAT:
Column:
    1   ===>Comment | WINDOWING ("*"= comment, BLANK, "+" or "-" = windowing)
    2-6 ===>MM/DD   (MONTH/DAY SPEC - SLASH REQ'D!)
    7   ===>(BLANK)
    8-14 ===>MTWTFSS   (DAY-OF-WEEK SPECIFICATION)
    15 ===>(BLANK_
    16 ===>COMMAND DATA   (THROUGH COLUMN 71)
    72 ===>CONTINUATION   (ONLY ONE CONTINUATION ALLOWED)

Column 1 previously was only used if the line was to be a comment. Now it may now be used to tell SyzAUTO/z whether this line should be "windowed" meaning that we can perform this line on a restart of SyzAUTO/z if we had skipped it while we were down (for an IPL or any other reason). SyzaAUTO/z can "Window" back for up to 24hours or the previous midnight, whichever is less. The current version of SyzAUTO/z (Version 6) cannot window a previous day. Subsequent version may allow that feature. Please note that the WINBLANK=YES|NO command will tell SyzAUTO/z what to do with the commands in the command packages that contain a blank in column 1 of the command line. The older versions of SyzAUTO/z did not use this position and it is now used to denote windowing processing control. This setting (WINBLANK=YES|NO) will tell SyzAUTO/z how to treat a "BLANK" space as the character that will/can be located in column 1 of the command package lines. Since older versions of SyzAUTO/z did not use that space, it is probable that it will already contain a blank space. This will tell SyzAUTO/z whether to treat that "BLANK" character as the same as a "+" (Plus sign) which means that it should be supported as a windowed command, or a "-"(minus sign) which means that we should treat it as a non-Windowed line. A non-windowed line will not be eligible for windowing support on a restart of SyzAUTO/z

An Asterisk in either the MM (columns 2 and 3) or DD areas (columns 5 and 6), allows execution for all MONTHS or all DAYS of a MONTH. Asterisks in both fields effectively nullifies DAY-WITHIN-MONTH criteria, and will allow the command to be processed on any month/day combination, but further processing of the DAY-OF-WEEK can still override processing, as will be described later.

If "**/**" is specified in columns 2-6, the command data on this entry is subject only to DAY-OF-WEEK criteria. This is the same as all BLANKS in columns 2-6.   If the MM/DD field is non-BLANK, then the request will be processed on the MONTH/DAY specified, but is also subject to DAY-OF-WEEK processing specifications.

If a request is to be processed on the same day of every month, specify the MONTH as asterisks. The following will process the command only on the 5th of each month (**/05). If a command is to be processed on EVERY day of a given month you would specify the day as asterisks. The following specification (09/**) in the MM/DD field would perform the command on every day of the month of September (depending on later DAY-OF-WEEK processing, of course).

Continuation is supported only if the subsequent record (the next line of the member) has ALL BLANKS in columns 1 through 15. The continued data begins in column 16.

EXAMPLES OF AUTOMATIC COMMANDS:

```
   COL-1---->*MM/DD MTWTFSS COMMAND-------------------------------------->72
```

MEMBER=@0300   (3 am)
```
   COL-1----> * Submit backups 1,2 and 3 on Mondays through Saturdays
   COL-1---->         XXXXXX  F A,BACKUP1,BACKUP2,BACKUP3 (Monday thru sat)
   COL-1----> * Submit weekly backup on Sunday only
   COL-1---->              X F A,BACKUPWK
   COL-1----> **/01 ******* F A,MONTH01
   COL-1----> **/01       S  F A,FIRSTSAT  SCHEDULE JOB ONLY ON THE
   COL-1----> **/02       S  F A,FIRSTSAT  1ST SATURDAY OF THE MONTH
   COL-1----> **/03       S  F A,FIRSTSAT            "
   COL-1----> **/04       S  F A,FIRSTSAT            "
   COL-1----> **/05       S  F A,FIRSTSAT            "
   COL-1----> **/06       S  F A,FIRSTSAT            "
   COL-1----> **/07       S  F A,FIRSTSAT            "
```

MEMBER=@0755   (7:55 am)
```
   COL-1---->       MTWTF    START TCAM
```

MEMBER=@0800   (8 am)
```
   COL-1---->       MTWTF    F TCAM,TS=START
   COL-1---->       MTWTF    $HQ,P
   COL-1---->       MTWTF    $AQ,T
```

MEMBER=@0900   (9 am)
```
   COL-1---->       -        SEND 'TODAY IS BLUE MONDAY...KEEP SMILING       *
   COL-1---->                '          (This line was continued)
   COL-1---->        -       SEND 'TODAY IS TUESDAY...FOUR MORE TO GO'
   COL-1---->         -      SEND 'TODAY IS WEDNESDAY...YOUR HALF WAY'
   COL-1---->          -     SEND 'TODAY IS THURSDAY...HANG IN THERE'
   COL-1---->           -    SEND 'THANK GOD IT''S FRIDAY'
   COL-1---->            -   SEND 'IT''S SATURDAY.......... DIDN''T YOU FINIS*
   COL-1---->                H?'        (This line was continued)
   COL-1---->             -  SEND 'WOW... EVEN SUNDAY YOU WORK!!!!!!'
```

MEMBER=@1645   (4:45 pm)
```
   COL-1---->       MTWTF    SE 'TSO WILL BE GOING DOWN AT FIVE SHARP'
```

MEMBER=@1655   (4:55 pm)
```
   COL-1---->       MTWTF    SE 'HURRY UP  -  TSO WILL BE DOWN FIVE MINUTE   *
   COL-1---->                S'          (This line was continued)
```

```
 MEMBER=@1659      (4:59 pm)
    COL-1--->        MTWTF   SE '**** TSO IS GOING DOWN IN ONE MINUTE      X
    COL-1--->                ****'       (This line was continued)
---COL-1---->         MTWTF   SE '****   YOU WILL RECEIVE NO WARNING        X
    COL-1---->                ****'        (This line was continued)


 MEMBER=@1700      (5 pm)
    COL-1---->        MTWTF   F TCAM,TS=STOP
    COL-1---->        MTWTF   $HQ,T
    COL-1---->        MTWTF   $AQ,P
    COL-1----> 05/17 ******* F A,TESTJOB  SCHEDULE ONLY ON MAY 17
    COL-1----> 05/16 *****   F A,TESTWKDY SCHEDULE ONLY ON MAY 16 Weekdays
    COL-1----> 05/16     **  F A,TESTWKND SCHEDULE ONLY ON MAY 16 Weekends


 MEMBER=@1702      (5:02 pm)
    COL-1---->        MTWTF   Z TP



        EXAMPLES OF OPERATOR 'DEMAND' COMMANDS:

    COL-1---->*MM/DD MTWTFSS COMMAND---------------------------------------
---->72

 MEMBER=IPL                           (  S AUTO.A,PARM=IPL  )
    COL-1---->        XXXXXXX S RMF.RMF
    COL-1---->        XXXXXXX S OPSWTR.OPSWTR
    COL-1---->        XXXXXXX F A,-VARYCONS,VARYALL,SETJES

 MEMBER=SETJES                        (  F A,-SETJES )
    COL-1---->        XXXXXXX $TPRT1,Q=A
    COL-1---->        XXXXXXX $TI9,Z
    COL-1---->        XXXXXXX $SI9
    COL-1---->        XXXXXXX $SPRT1

 MEMBER=SHUTDOWN                       (  F A,-SHUTDOWN  )
    COL-1---->        XXXXXXX $P
    COL-1---->        XXXXXXX $IPRT1
    COL-1---->        XXXXXXX $PLNE1
    COL-1---->        XXXXXXX $ELNE1
    COL-1---->        XXXXXXX P OPSWTR
    COL-1---->        XXXXXXX P RMF
    COL-1---->        XXXXXXX P A

 MEMBER=UNLOAD                         (  F A,-UNLOAD  )
    COL-1---->        XXXXXXX U 180
    COL-1---->        XXXXXXX U 181
    COL-1---->        XXXXXXX U 182

 MEMBER=VARYALL                        (  F A,-VARYALL  )
    COL-1---->        XXXXXXX V 180-188,ONLINE
    COL-1---->        XXXXXXX V 150-158,OFFLINE
```

```
   COL-1---->        XXXXXXX V 160-16F,ONLINE

 MEMBER=VARYCONS                           (   F A,-VARYCONS   )
   COL-1---->        XXXXXXX V 120,CONSOLE,ROUT=(1,2,8,9,10,12,13,14,15)
   COL-1---->        XXXXXXX V 121,CONSOLE,ROUT=(3,4,5,6)
   COL-1---->        XXXXXXX V 122,CONSOLE,ROUT=(7)
   COL-1---->        XXXXXXX V O-123,CONSOLE,ROUT=ALL

 MEMBER=VARYOFF                            (   F A,-VARYOFF   )
   COL-1----> 09/**      XX V 150-158,OFFLINE   ONLY FOR SEPT WKEND
   COL-1---->        XXXXXXX V 160-168,OFFLINE
   COL-1---->        XXXXXXX V 170-178,OFFLINE
```

RACF changes that may be necessary for some sites:

Depending on your sites configuration, you may or may not have to perform RACF changes to create these procedures. If so, then the following may be necessary, the group you choose should be capable of issuing operator commands as well as submitting any JOB's that you might want SyzAUTO/z to take care of, we choose the SYS1 group for simplicity:

ADDUSER SYZAUTO DFLTGRP(STCGROUP) NOPASSWORD NOOIDCARD

CONNECT SYZAUTO GROUP(STCGROUP) AUTH(USE)

CONNECT SYZAUTO GROUP(SYS1) AUTH(USE)

RDEFINE STARTED SYZAUTO.* STDATA(USER(SYZAUTO) +
   PRIVLEDGED(YES) TRUSTED(YES) GROUP(STCGROUP))

SETROPTS RACLIST(STARTED) REFRESH

Also, some subsystems (like HSM or CICS and DB2) may require that you authorize SyzAUTO/z in order to be able to issue commands (such as HSM recycle commands), so you may have to perform other RACF updates. Please contact Syzygy technical support for more complete information. You should treat SyzAUTO/z like any other member of your systems programming staff, normally, if you authorize SyzAUTO/z with access tot he same RACF resources as your systems group, you should not run into any problems. Typically, connecting SyzAUTO/z to the SYS1 group will suffice, but there may be other changes that are necessary, and are dependent upon how complex your RACF rules are set up.

SyzEmail: (Send Job Condition Codes via Email)

This utility has been superceeded by SyzMAIL/z for sites running SyzMPF/z Version 4 and up.

This program is a utility provided as part of the SyzAUTO/z product package, which can be used to send all or some of the condition codes of the job stream in which this program is added, to as E-Mail via SMTP to any individual or number of individuals that they wish. Additionally the user may elect to also (or instead of) having SyzEmail send the step condition codes to a sequential file (new or existing) so that a log of the execution for a particular JOB stream can be monitored over a period of time.

The SyzEmail program interfaces with the sites SMTP server under z/OS, and can (if the site allows) send email directly to any user in the sites local network, or to any user with an email address, (if the site allows the transmission).

The SyzEmail program, by default, sends the condition codes of all steps which precede the SyzEmail step within the JOB, but the user can select to send only those steps which are of interest to them (up to 16 within a single JOB). For instance, if a JOB has 500 steps, but the user only cares about the condition codes of STEP031 and STEP499, then they can elect to have SyzEmail only send information on those two steps.

The SyzEmail program will send the condition code EMAIL to all users that have been identified to receive the email. There is no limit to the number of users that can be notified from a single execution of SyzEmail.

SyzEmail Installation

The SyzEmail program was delivered in the same XMIT package as your SyzAUTO/z program and for best results it should be placed into a load library that is in the systems LINKLST concatenation pointed to by the sites "SYS1.PARMLIB(PROGxx)' member. The library that SyzEmail is placed into need not be authorized, and also need not be in LINKLIST. If not placed in LINKLIST, then a //STEPLIB DD must be added to the execution JCL for the SyzEmail program.

SyzEmail execution JCL overview

To be used, SyzEmail needs to be added as a step to an existing BATCH JOB. Ideally, it should be added as the last step of the JOB, because only SyzEmail can only report on the steps that have executed **before** it does. There are several parameter options and DD cards that can be used by SyzEmail, some of them are required, but most are not. SyzEmail does not report on it's own condition code from the step that it executes in.

SyzEmail Parameters

There are currently 3 execution parameters that can be used with SyzEmail, and the site may elect to use any of them as it may desire. Some of the parameters can be "pre-set" by zapping the SyzEmail module via supplied AMASPZAP control cards. The Zaps will make it easier for users to utilize the SyzEmail program, and do not keep the site from overriding the settings with a PARM override, even after the Zaps have been installed.

NODE parameter

> *Usage*
> NODE(nodename)
> Where "nonename" is the 1 to 8 character JES NJE node name.

This is the JES node which the job is executed on, and it's used to identify the output to SMTP when the email is generated within the HELO message. In most cases, SMTP doesn't really care what the node name is, but for some sites, it's a requirement that the node name be valid. The default within the product is "UNKNOWN".

Zap Parms

        NAME SYZEMAIL SYZEMAIL
        VER 1658 E4D5D2D5,D6E6D540   verify "UNKNOWN "
        REP 1658 XXXXXXXX,XXXXXXXX   replace with your node name pad
                            with X'40'
                            blanks to 8 characters total.

FROM parameter

Usage

          FROM(USER@DOMAIN.COM)
        Where "USER@DOMAIN.COM" is a VALID email address that your site will

                        Accept as allows to send email from your

                        mainframe to your network.

This is the VALID email address that you wish to use to appear to send the condition code email from. Some sites require a specific name to send email from the mainframe, such as the nodename or system ID of the system followed by the valid URL of the network you are part of. For us at Syzygy, we use "Notify@SyzygyInc.com", but only because we think it looks nice, not because of any special requirement. The default within the program is "EMAIL@DOMAIN", and is not normally valid for most sites.

Zap Parms

        NAME SYZEMAIL SYZEMAIL
        VER 0D9D C5948189937CC496948189956E4040
                            verify   "EMAIL@DOMAIN> "
        VER 0DAC 40404040,40404040   Verify 8 more blanks
        VER 0DB4 40404040,40404040   '
        VER 0DBC 40404040,40404040   '
        VER 0DC4 40404040,40404040   '
        VER 0DCC 40404040   Verify last 4 blanks

        REP 0D9D XXXXXXXX,XXXXXXXX   replace with your desired EMAIL ID
                            up to 50 characters, you must follow the
                            ID with a greater than sign ">" to denote
                            the end of the email address.

*Note: the ZAP requires a ">" to immediately follow the email address, but the override in the parm field of the execution does not require it, it is placed there programmatically if you use the execution parm override "FROM(email@address.com) but if you use the ZAP, you have to place it at the end of the email address or the program doesn't know when to stop reading the address.

STEP parameter

Usage

STEP(Step#,STEP#,...up to 16 separate steps)   for standard program execution

or

STEP(PROCSTEP.STEP,PROCSTEP.STEP, etc.)   for individual PROCSTEP execution

or

STEP(PROCSTEP.*,etc)   All steps of a PROCSTEP execution

Where "STEP" and "PROCSTEP.STEP" identifies the STEPs or PROCSTEPS that you wish to limit the notifications to within the JOB.

This identifies which JOB STEPs or PROCSTEP.STEPs that you wish to limit the STEP condition code notifications to. By default, all steps are included. There is no ZAP for this parameter.

DD requirements

NOTIFY DD

This DD statement is used to identify what Email Id or Ids are to be sent the same condition code notification messages. This DD card is required. Without it, how would we know who to send the Email to? The DD may be specified as inline "DD *" or a sequential data set or a member of a PDS. The cards that make up this file can only specify ONE Email address per line, but you can have any number of lines in the file. The Email address(es) must begin in column 1 of the card. The DCB of the data set must be LRECL(80) and may have any valid BLOCKSIZE. An "*" in column 1 denotes a comment and can be used for your informational purposes within the data set, they will be skipped by the SyzEMail program.

Example 1

This will send the condition codes to 3 separate people using "DD *" format.
..... all the other steps of the JOB go here
```
//*
//EMAIL        EXEC PGM=SYZEMAIL,COND=EVEN,   (Use CODE=EVEN execute
                                                      no matter what)
//  PARM='NODE(JESSYS1),FROM(Notfy@SyzygyInc.com)'
//NOTIFY      DD      *
* comment card
Brian_Westerman@SyzygyInc.com
* another comment card
Pepper_Kosch@SyzygyInc.com
* comment
Fred@Flintstone.net
/*
```

Example 2

This will send the condition codes to 3 separate people using a NOTIFY PDS member
..... all the other steps of the JOB go here
```
//*
//EMAIL        EXEC PGM=SYZEMAIL,COND=EVEN,   (Use CODE=EVEN execute
                                                      no matter what)
//  PARM='NODE(JESSYS1),FROM(Notfy@SyzygyInc.com)'
//NOTIFY       DD      DISP=SHR,DSN=NOTIFY.PDS(THISJOB)
```

contents of the PDS member (THISJOB)

* comment card
  Brian_Westerman@SyzygyInc.com
  * another comment card
  Pepper_Kosch@SyzygyInc.com
  * comment
  Fred@Flintstone.net

SYSPRINT DD

   If a SYSPRINT DD statement is used, not only will the step condition codes be sent to the Email addresses that were specified in the NOTIFY DD, but they will also be written to the sequential data set or PDS member pointed to by the SYSPRINT DD. If the data set already exists, it's DCB attributes will be used. If no DCB attributes are specified the data is written with a RECFM=FB, LRECL=133, BLKSIZE=6118.

Example 1

   This will send the condition codes to 3 separate people and logs the same messages to SYSOUT=* so that they can be viewed as the last output for this JOB.

```
..... all the other steps of the JOB go here
//*
//EMAIL      EXEC PGM=SYZEMAIL,COND=EVEN,   (Use CODE=EVEN execute
                                                 no matter what)
//  PARM='NODE(JESSYS1),FROM(Notfy@SyzygyInc.com)'
//NOTIFY     DD     *
* comment card
Brian_Westerman@SyzygyInc.com
* another comment card
Pepper_Kosch@SyzygyInc.com
* comment
Fred@Flintstone.net
/*
//SYSPRINT  DD     SYSOUT=*
```

Example 2

This will send the condition codes to 3 separate people using a NOTIFY PDS member and will send the same messages to a sequential data set that is used by this job every time it runs.

```
..... all the other steps of the JOB go here
//*
//EMAIL      EXEC PGM=SYZEMAIL,COND=EVEN,   (Use CODE=EVEN execute
                                            no matter what)
//  PARM='NODE(JESSYS1),FROM(Notfy@SyzygyInc.com)'
//NOTIFY     DD     DISP=SHR,DSN=NOTIFY.PDS(THISJOB)
//SYSPRINT   DD     DISP=SHR,DSN=PROD.CCODES.THISJOB
```

Sample SyzEmail Output

Sample JOB JCL

```
//SYZTEST JOB (SYZYGY),'WESTERMAN',CLASS=A,MSGCLASS=X,
// NOTIFY=BRIAN,TIME=1440,REGION=0M
//*
//STEP1    EXEC PGM=IEFBR14
//STEP2    EXEC PGM=IEFBR14
//STEP3    EXEC PGM=IEFBR14
//COMMAND  EXEC  PGM=SYZCMDZ3
//SYSABEND DD    SYSOUT=*
//IEFRDER  DD    *
EOSM=ALL
D A,L
PAUSE 1s
//STEP4    EXEC PGM=IEFBR14
//STEP5    EXEC PGM=IEFBR14
//STEP6    EXEC PGM=IEFBR14
//COMPCODE EXEC PGM=SYZEMAIL,COND=EVEN,
// PARM='NODE=(TEST),FROM=(Westerman@SyzygyInc.com)'
//NOTIFY   DD *
Brian_Westerman@SyzygyInc.com
/*
```

Sample Email Output

```
Date: MON, 01 FEB 2008 23:50:09 -0700
From: SYZTEST Job Status <Notify@SyzygyInc.com>
Subject: SYZTEST (J04508)- JOB/STEP Notification - Syzygy Computer Center - NJE Network Node: TEST SYSID: SYS1


SYZTEST (J04508)- STEP1 IEFBR14 ENDED, CODE 0000, 01 FEB 2008 23:50:09

SYZTEST (J04508)- STEP2 IEFBR14 ENDED, CODE 0000, 01 FEB 2008 23:50:09

SYZTEST (J04508)- STEP3 IEFBR14 ENDED, CODE 0000, 01 FEB 2008 23:50:09

SYZTEST (J04508)- COMMAND SYZCMDZ3 ENDED, CODE 0000, 01 FEB 2008 23:50:09

SYZTEST (J04508)- STEP4 IEFBR14 ENDED, CODE 0000, 01 FEB 2008 23:50:09

SYZTEST (J04508)- STEP5 IEFBR14 ENDED, CODE 0000, 01 FEB 2008 23:50:09

SYZTEST (J04508)- STEP6 IEFBR14 ENDED, CODE 0000, 01 FEB 2008 23:50:09
```

Cross references to other products

SyzAUTO/z can operate in conjunction with the SyzCMD/z product. SyzCMD/z can "Script" any series of commands and provide interactive feedback to the script logic so that intelligent choices can be made "on the fly".