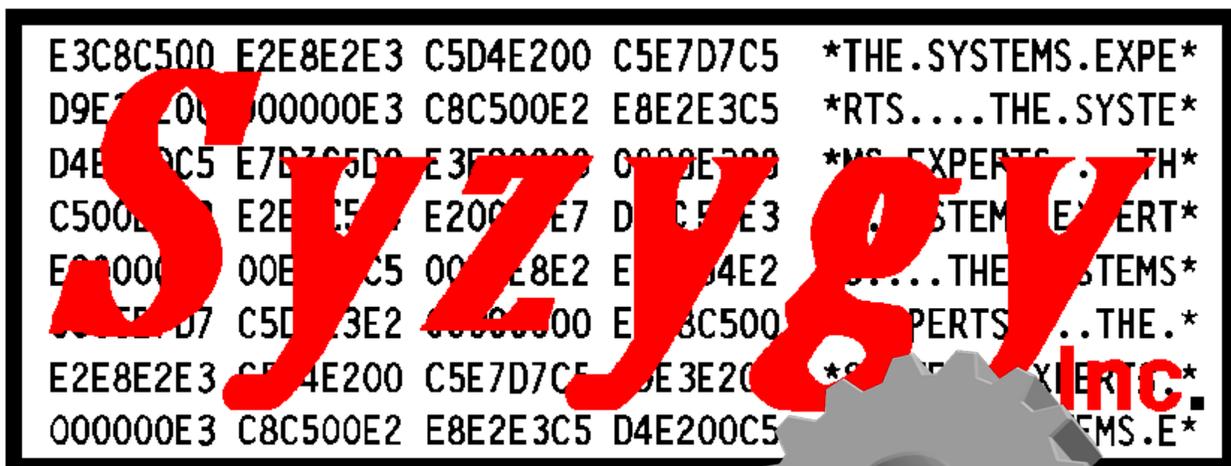


[Version 2.]
November 23, 2014

SyzMAIL/z



[Installation and User's Guide]

z/OS based eMail and SMS/Text automation and general API

Revision History

Version	Date	Revision Description
2.0	11/23/2014	<p>New support for HTML email</p> <p>New support for wider range of information inside Email and SMS packages</p> <p>New command DBUGPKG which controls console message display of package location information</p> <p>New Command SENTMSG which controls console message display of "email sent to:xxxxxx" messages</p> <p>New MAXLINES command to control the maximum size of an email package to be sent (default 1000 lines)</p> <p>General code changes for speed and reduction of memory usage and CPU resources consumed by SyzMAIL/z</p>
1.1a	2/28/2014	<p>Added SMTPCLASS and SMTPNAME parameters for sites that might not use the IBM defaults (class=B and SMTP=SMTP)</p> <p>Altered expiration checking code to be not quite so intrusive.</p>
1.1	1/23/2014	<p>Initial GA release:</p> <ul style="list-style-type: none"> • Global defaults for all incoming Syzygy products To:, From:, Cc:, Bcc:, Subject:, ReplyTo: Message support • Automatic cleanup of incomplete or invalid email packages • Installation Defaults supported via Parmlib member • Global Nickname support for both Email and SMS text • Debugging mode to file and console • Remove constraints of maximum email length, now only dependent on the package sent via the API. • The NICKNAME DSN is now a variable that can be set without JCL (it can be identified within the new parmliib member)
1.0	7/12/2013	Version 1 Beta Program Release

Table of Contents

Contents

Revision History	ii
Table of Contents.....	3
1. Introduction.....	5
Purpose	6
Related Documents	6
Conventions	7
Problem Reporting Instructions	7
2. Overview.....	8
Functional Description.....	9
3. Installation Instructions.....	10
Procedures.....	10
STEP 1:	10
STEP 2:	10
STEP 3:	10
STEP 4:	11
STEP 5:	11
STEP6:	13
STEP7:	14
STEP8:	14
Required RACF changes (that may be necessary for some sites):	15
STEP9:	15
4. Reference	18
Global Settings (system parmlib)	19
Sample SYZEML00 “SYSTEM PARMLIB” DEFAULTS member	20
NICKNAME	22
DEBUG= Y/N	23
DEBUGWTO= Y/N	24
ECHO.....	26
INTERVAL.....	27
LOGCLASS=.....	28

LOGDSN=	29
SMTPCLASS=	32
SMTPNAME=	33
TimeZone or TZ =.....	34
STATistics Stats.....	35
WARN= Y/N.....	36
Operator Control Commands	37
DEBUG= Yes No	38
NODEBUG	38
DEBUGWTO= Yes No	40
NODEBUGWTO	40
QUERY ALL.....	41
STOP.....	43
WARN= Yes No.....	44
NOWARN.....	44
5. Available Cellular gateways.....	45

[The rest of this page has been intentionally left blank]

1. Introduction

SyzMAIL/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 several main products;

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

SyzCMD/z, the Command Scripting facility, provides the z/OS site with a comprehensive means 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 z/OS site to respond to any Console-type message with predefined scripts allowing complete interactive control of any console situation or message. SyzMPF/z provides an automated means for 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.

SyzMail/z, the system eMail facility. SyzMail/z supports the monitoring and gathering of eMail messages from all Syzygy products and Syzygy supported Customer API's, and some other vendor products. This facility allows the use of Nicknames for the To:, Cc:, and Bcc: commands, so that multiple recipients can be supported with a single nickname entry. The facility supports storage of the email messages until they are successfully passed off to the sites SMTP server for distribution. Full statistics are provided for all functions.

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
SyzSPOOL/z Installation and Operations Guide

SyzMPF/z	Installation and Operations Guide (this guide)
SyzNotify/z	Installation and Operations Guide
SyzMail/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

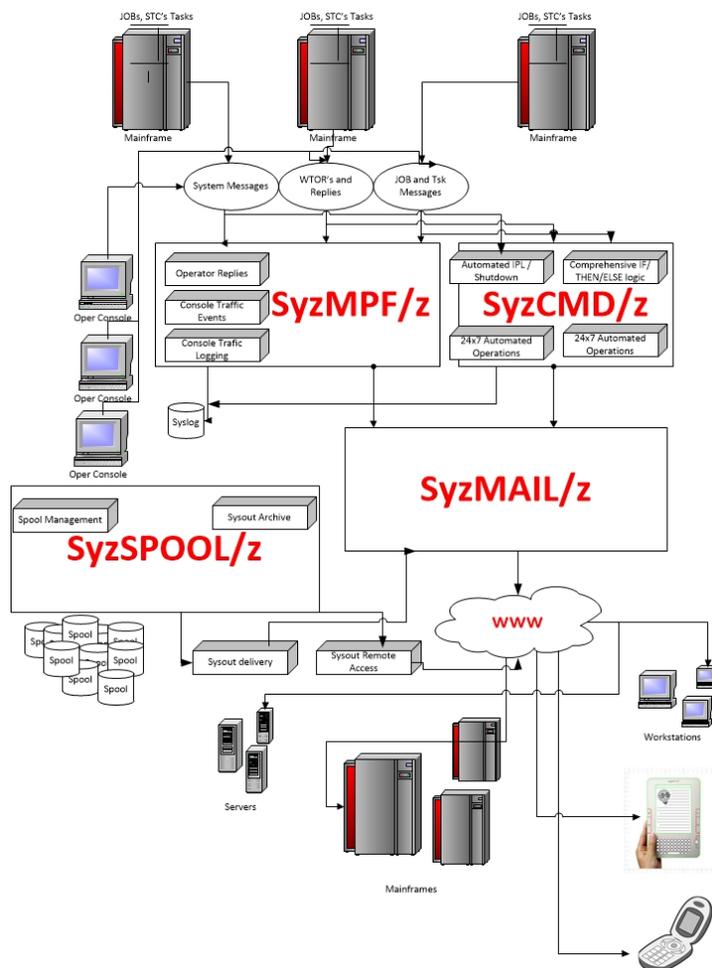
2. Overview

The SyzMAIL/z eMail and SMS/text processing utility provides a z/OS data center with the capability of sending eMail or SMS text messages from most Syzygy Automation products. Currently, SyzMPF/z, SyzCMD/z, and SyzSPOOL/z support the use of this facility, full support for SyzAUTO/z and SyzMON/z is planned for 2014/2015.

The actual eMail and SMS/Text packages are generated within other Syzygy Automation products, and an API is available to allow other products or even Client site software (currently in beta) to create and send email. Those products create the mail package, and it is the responsibility of SyzMAIL/z to locate those packages and format them and resolve nicknames and other information for use by SMTP to be sent to the end recipient(s) via eMail or SMS/Text.

Functional Description

SyzMAIL/z is implemented as a separate Started Task (or, if absolutely necessary, a BATCH job) and will use the startup global parameters from the SyzEMLxx member of the system parmlib concatenation. SyzMAIL/z is composed of several 31bit and 64bit subroutines (CSECTS) generated into a single load module, which must be executed from a “APF authorized” library in Linklist. Further, many of the functions of SyzMAIL/z can be protected by calls to the sites Security system (RACF, ACF/2 Top Secret, etc).



3. Installation Instructions

Procedures

STEP 1:

The software is sent to the authorized site representative's email address. It will be attached to this E-Mail communication as a .ZIP file. You should save or copy this file to a suitable directory on your PC.

STEP 2:

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

software.XMI where "software" is the name of the name of the Licensed Software Product that you have ordered.

STEP 3:

PRE-ALLOCATE THE RECEIVING XMIT FILE(S) ON YOUR MAINFRAME

Pre-allocate a container file to hold the product's load module library transmit format library n MVS (e.g. 'userid.software.[linklib](#).date.xmit') with the following DCB.

Space: 1 Cylinder
Organization : PS
Record format : FB
Record length : 80
Block size : 3120

Pre-allocate a container file to hold the Installation Source/samples transmit format library on MVS (e.g. 'userid.software.instlib.date.xmit') with the following DCB.

Space: 1 Cylinder
Organization : PS
Record format : FB
Record length : 80
Block size : 3120

STEP 4:

UPLOAD THE software.XMI FILE(S) 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.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 LINKLIB and SAMPLIB PDS(s)

On your MAINFRAME perform a RECEIVE command on the uploaded XMIT files from step 4 above as follows:

NOTE: If operating under ISPF use panel option 6, or you should exit ISPF to TSO READY state.

COMMAND:

```
RECEIVE INDATASET('userid.software.LINKLIB.date.xmit')
```

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

After completing 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 can name the dataset something other than the "default" name that the file was shipped as, which may not meet your site's Dataset Naming requirements.

You can enter:

```
DSNAME('userid.software.linklib')
```

... 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 output dataset name.

NOTE: PLEASE BE SURE to choose a dataset 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.linklib') 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).

Repeat this step (STEP 5) specifying the name of the (Installation Library) INSTLIB dataset:

```
RECEIVE INDATASET('userid.software.INSTLIB.date.xmit')
```

***where 'userid.software.INSTLIB.date.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 can enter:

```
DSNAME('userid.software.INSTLIB')
```

... where "userid.software.INSTLIB" will be a new Source/Samples PDS that you have selected which will contain all of the Installation related samples required to help with the correct operation of the licensed software that you have ordered. Do not pre-allocate this output dataset name.

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. It is strongly suggested that you place the provided module(s) in a system LinkList library. 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:**Add the SYZEMLnn parameter to your System Parmlib Concatenation.**

In order to support greater control over the capabilities of SyzMAILz, a new feature was added which allows the site to control overall operation of the SyzMAILz product. This member, (which can exist in ANY level of the SYSTEM PARMLIB concatenation), will hold all of the “defaults” for global SyzMAIL/z operational capabilities. If this member does not exist, then SyzMAIL/z operates with programmed defaults. In addition, some features may not work correctly.



***NOTE: if the parmlib defaults member is not found, some features of SyzMAIL/z may not function or may function differently.

The default member name is SYZEMLD00 (zero-zero). You can override that default at run time, (see JCL PARM option), 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.

STEP8:

Add the Startup JCL for SyzMAIL/z to a local Proclib (assuming you are going to run SyzMAIL/z as a started task). Example JCL is located in the supplied SAMPLIB, but is shown here as well:

```
//SYZEMAIL EXEC PGM=SYZMAIL1,TIME=1440      V1.00
//* * * * *
//* THIS TASK IS THE EMAIL PROGRAM          * * *
//* * * * *
//* * * * *
//STEPLIB DD DISP=SHR,DSN=SYS1.SYZMAIL1.LINKLIB <--AUTH LINKLIB
```



The started task should have RACF authority to read the data set(s) identified in the parameter or in the SYZEMLxx parmlib member as well as read the parmlib itself.

Required 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 will be necessary:

```
ADDUSER SYZEMAIL DFLTGRP(STCGROUP) NOPASSWORD NOOIDCARD
```

```
CONNECT SYZEMAIL GROUP(STCGROUP) AUTH(USE)
```

```
CONNECT SYZEMAIL GROUP(SYS1) AUTH(USE)
```

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

```
SETROPTS RACLIST(STARTED) REFRESH
```

STEP9:

Create the Nicknames PDS. Add the name to SYZEMLnn in your System Parmlib Concatenation.

Nicknames

This dataset may be allocated via DD statement in the SyzMAIL/z startup or via a system parmlib member SYZEMLxx. It is suggested that the site use the parmlib method since it provides greater flexibility. The Nicknames PDS is used to identify what Email Id or Ids, or SMS text addresses are to be sent the email packages. This DD card is required or must be allocated dynamically via the SYZEMLxx parmlib member if NICKNAMES are desired to be used. Without it, SyzMAIL/z could not resolve the nickname requests from other products.

The individual member names within this PDS are the “nicknames” that will be used, and they contain from 0 to 256 email addresses or SMS text addresses to be used by that nickname. The cards that make up this file can only specify ONE Email address or SMS text address per line, but you can have any number of lines in the file. The Email/SMS 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 SyzMail/z program.

Example

SyzMPF/z uses the following for TASK PAYROLL1 which has a jobcard NOTIFY=PAYROLL:

EMAIL MAXCC STEPCC	← send the maxCC and step details
To: &NOTIFY	← to whoever has the current NOTIFY on the JOB card
SENDMAIL	← generate the email package

contents of the PDS member (PAYROLL)

* comment card

John.Doe@thissite.com

* another comment card

Fred@Flintstone.net

1234567890@txt.att.net

← SMS text message address at AT&T

1234567890@messaging.sprintpcs.com

← SMS text message address at Sprint

The result is that the email package created by SyzMPF/z (which only specified that the email be sent to the NOTIFY=user on the JOBCARD, will be resolved to be sent to John Doe, Fred, and to two separate SMS text addresses by SyzMAIL/z.

4. Reference

Global Settings (system parmlib)

A feature of Version 1.0 of SyzMAIL/z is the inclusion of GLOBAL settings which are now contained within a member of the system parmlib concatenation. The default parmlib member name is SYZEML00, and the suffix of this member can be controlled via the startup execution parm of the SyzMAIL/z STC of "ID=nn". i.e.:

```
//SYZMAIL EXEC PGM=SYZMAILZ,PARM='ID=A7'
```

will result in the parmlib member [SYZEMLA7](#) being used instead of SYZEML00.

There are several new GLOBAL parameters that can be contained within the parmlib member that SyzMAIL/z uses and they will be described in this section. These settings will be used for all invocations of the SyzMAIL/z message processor. These settings are analyzed by the SUZMAIL/z program and while every effort is made to edit the provided GLOBAL settings programmatically, care should be taken to view the output from SYZMAIL started task when the settings are changed to make sure they are used in the way the site actually wanted them to be used. These settings are loaded into a memory area that is immediately accessible by the SyzMAIL/z without any physical i/o or memory access delays. The method used is extremely efficient, and the overhead of using the GLOBAL settings in this way are so low as to be considered trivial.

Sample SYZEML00 "SYSTEM PARMLIB" DEFAULTS member

```

*
* This is the V2.x SYZMAIL startup parm member
*
* CONFIRM=Y | n          Confirm delivery of Command packages
*                        Default is Y
*
* DEBUG=y | N           Detail DEBUG MODE to the COMMAND log
*                        Default is N
*
* DEBUGWTO=y | n        Detail DEBUG MODE to the CONSOLE
*                        Default is N
*
* DEBUGPKG=y | n        Detail DEBUG MODE of packages to CONSOLE
*                        Default is Y
*
* SENTMSG=y | n          Detail Messages about package delivery
*                        Default is Y
*
* ECHO=y | N,            ECHO commands to the console
*                        Default is N
*
* LOGCLASS=A            JOB class for LOG dataset (default A)
*
* LOGDSN=SYS1.SYZMAIL.LOG DSN to use for LOG (if not SYSOUT)
*                        Required if SyzMAIL/z is SUB=MSTR
*                        RECFM=FBA LRECL=133 BLKSIZE=any
*
* MAXLINES=nnnn | 1000  Max lines allowed per email package
*                        Default=1000
*
* RDRCLASS=A            JES JOB Class that all submitted JOBs
*                        Default is A
*
* WARN=Y | n             Inform of not critical messages
*                        Default is Y
*
* INTERVAL=9999s | m | h up to 9999 hours Max allowed
*
* TIMEZONE=+/-nnnn or USzone Time zone of this system (default is PST)
*
* SMTPNAME=SMTP | Site SMTP name (Default is SMTP)

```

*

*

*

DEBUG=No
LOGCLASS=A
RDRCLASS=C
ECHO=Yes
WARN=N
INTERVAL=10S
TimeZone=-1000
NICKNAME=Syzygy.Notify
STATS=Yes

NICKNAME

= Nickname.data.set.name

The Nickname GLOBAL setting parameter is used to tell the SyzMAIL/z Subsystem which NICKNAME dataset to use for processing all email packages.

Parameters:

Any supported PDS data.set.name



***note:** This dataset MUST exist, and MUST be cataloged

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Syzygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

DEBUG= **Y/N**

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

Parameters:

Yes	=	YES
No	=	NO

Default No

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

DEBUGWTO= **Y/N**

The DEBUG parameter entry is used to control the ability for SyzMAIL/z to send detailed debugging messages about the email packages it is sending to the console dataset.

Parameters:

Yes	=	YES
No	=	NO

Default No

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

DEBUGPKG= **Y/N**

The DEBUGPKG parameter entry is used to control the messages which SyzMAIL/z normally sends to the OS console which detail the location and disposition of the eMail packages that it processes. These messages tell which product or API created the package and its size and location.

Parameters:

Yes	=	YES
No	=	NO

Default No

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGPKG=No	← debug email PKG location to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

ECHO

Yes | **No** (no is the default)

The ECHO GLOBAL setting parameter is used to tell the SyzMAIL/z Subsystem what degree of command ECHOing is to take place.

The settings are cumulative, so you can add or subtract them in any order or in separate command requests (one per line)

Parameters:

Yes Echo commands to the console
No Do not echo. (the default)

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Syzygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics
.....	

INTERVAL

nnnn[S|M|H]

Description:

SyzMAIL/z constantly scans for email packages to work with based on an interval programmed into the software. The default "interval" time of these waits is 15 seconds. Some sites might have a higher volume of email that they wish to be processed on a smaller interval and would rather not wait 15 seconds. Reducing the wait interval to even 1 second uses a trivial amount of extra resources, but 15 seconds was determined to be adequate for most sites. You can, with this command set that interval to any value from 1 second up to 9999

Parameters:

nnnn [S|M|H]

nnnn - required - Up to a 4 digit number of seconds (the default), minutes or hours to wait to allow syzMAIL/z to re-check the environment. The maximum allowed value of digits is 9999. "S" is the default and sets the preceding number to be that many seconds. You can also specify "M" for minutes or H for hours. If not specified, the default is 15 seconds (the SyzMAIL/z default)

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

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 SyzMAIL/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:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

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 SyzMAIL/z. This log is used to implement enhanced DEBUG features to SyzMAIL/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 SyzMAIL/z startup JCL. In instances where you would like to execute SyzMAIL/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 SyzMAIL/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:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

MAXLINES= **nnnnnn/1000**

The MAXLINES parameter entry is used to control the maximum number of email lines that will be sent of the email package. The packages are created outside of SyzMAIL/z, but in some cases it may be desirable to limit the number of lines that can be sent. This is especially important when files are being attached for which the size may not be known at the time they are created and the site may not wish to send 4 million lines of SMS text that contains SYSOUT of a task. When the MAXLINES number is reached, the EMAIL package data is truncated. The default is 1000 lines, That includes the actual email information lines, which typically are 25 lines. As such, this number should never be reduced to less than 25 unless the site desires that NO email or text be sent, or that very small contents are used.

Parameters:

999999 = any number up to 999,999 lines

Default 1000 lines maximum in email package

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
MAXLINES=2500	← Limit size of email to 2500 lines
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
Timezone=-0700	← Time zone if not provided
NICKNAME=Syzygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

SENTMSG= **Y/N**

The SENTMSG parameter entry is used to control the messages which SyzMAIL/z normally sends to the OS console which detail the email address or SMS text address that the eMail/SMS package is being sent to. These messages are informational, but some sites do not wish them displayed on the OS console or SYSLOG.

Parameters:

Yes	=	YES
No	=	NO

Default No

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
SENTMSG=No	← Display Who gets package on os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

SMTPCLASS=

'B | local SMTP output class'

The SMTPCLASS parameter entry is used to provide an alternate jes2 class for the SMTP writer supplied by IBM. Most sites will use class "B" (the default) for the SMTP output class used for this writer, but there are some sites that already are using class "B" or for some other reason elect to use another JES2 class for SMTP. This parameter allows for that alternate name.

Parameters:

B | JES2 class used by SMTP (1 character max)

Default "B"

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics
SMTPCLASS=M	← Alternate SMTP output JES2 class
SMTPNAME=SMTP2	← Alternate SMTP addressspace name

.....

SMTPNAME=

'SMTP | local SMTP name'

The SMTPNAME parameter entry is used to provide an alternate name for the SMTP writer supplied by IBM. Most sites will use SMTP for the name of this writer, but there are some sites that name SMTP something OTHER than "SMTP". This parameter allows for that alternate name.

Parameters:

SMTP | any alternate SMTP name (8 character max)

Default SMTP

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics
SMTPNAME=SMTP2	← Alternate SMTP addressspace name

.....

TimeZone or TZ =

Time zone offset to use (default = PST)

The "TimeZone or TZ" GLOBAL setting parameter is used to tell SyzMAIL/z what default Time Zone to use for all outgoing eMail packages. This setting is not required but is suggested because the default is PST. Any valid offset may be used, including the standard US time zone offsets, PST,PDT,EDT,EST,CST,CDT,etc.) The site can also specify the offsets to the west or (- negative) or east (+positive) of GMT. i.e. -0700 is the pacific time zone (PST) west 7 hrs.

Parameters:

Timezone or offset (+/-) Default is PST (TZ=-0700)

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

STATistics | Stats

= Yes| No

The STATs startup paramter will allow or disallow the creation of the SyzMAIL/z shutdown statistics

Parameters:

Yes	Produce shutdown statistics (the default)
No	No shutdown statistics will be produced

Example:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Syzygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

WARN= **Y/N**

The WARN parameter entry is used to control the ability for SyzMAIL/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:

.....SYS1.PARMLIB(SYZEML00) member contents...

DEBUG=No	← debug to LOG dataset
DEBBUGWTO=No	← debug email cards to os-console/syslog
LOGCLASS=A	← SYSOUT class of LOG dataset
LOGDSN=LOG.DATASET.NAME	← LOG dataset name if not SYSOUT
RDRCLASS=C	← Class that all jobs submitted by SyzMAIL/z go to
ECHO=Yes	← Echo commands to the console
WARN=N	← Generate warning messages as well as ERRORS
INTERVAL=10S	← interval to check for MAIL packages
TimeZone=-0700	← Time zone if not provided
NICKNAME=Szygy.Notify	← Nickname dataset to use for all mail packages
STATS=Yes	← produce shutdown statistics

.....

Operator Control Commands

SyzMAIL/z allows the operator, using the MVS Modify command, to query SyzMAIL/z to get information on the status of the system, the settings, and to perform several control functions. These commands can be issued from any valid operator console or from SDSF.

Example:

```
F SYZMAIL,QUERY ALL
```

The following Operator Commands are supported:

DEBUG	-	Turn on (or off) detail tracing of SyzMAIL/z
DEBUGWTO	-	Turn on/off email content tracing.
Interval	-	Change the scan interval
NODEBUG	-	Turn off detail tracing
WARN	-	Turn on (or off) warning messages
NOWARN	-	Turn off warning messages
QUERY	-	Display settings and resources used
STOP	-	Terminate SyzMAIL/z

DEBUG= **Yes|No**

NODEBUG

The DEBUG operator command can be used from the operator console (or from within a SyzMAIL/z command member) to turn on (or off) detailed debugging. The debugging information is logged in the SyzMAIL/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 SyzMAIL:

.....

Command:

F SYZMAIL,Debug=Y Modify to turn on detailed DEBUGging messages

.....

DEBUGPKG= **Y/N**

The DEBUGPKG parameter entry is used to control the messages which SyzMAIL/z normally sends to the OS console which detail the location and disposition of the eMail packages that it processes. These messages tell which product or API created the package and its size and location.

Parameters:

Yes	=	YES
No	=	NO

Default No

Example:

Example of command issued from console for SyzMAIL:

.....

Command:

F SYZMAIL,DebugPKG=Yes Modify to turn on detailed DEBUG Package messages

.....

DEBUGWTO= Yes | No

NODEBUGWTO

The DEBUGWTO operator command can be used from the operator console (or from within a SyzMAIL/z command member) to turn on (or off) detailed EMAIL debugging to the operator console. It may be advantageous at some point to see what lines are contained within the email package that SyzMAIL/z has received. This command will display those upon the console. The debugging information is also logged in the SyzMAIL/z LOG dataset or SYSOUT (depending on which is being used). NODEBUGWTO is the same as DEBUGWTO=No. The default is "Yes", i.e DEBUGWTO with no operands is the same as DEBUGWTO=Yes.

Parameters:

 Yes | No = Yes or No

Default Yes

Example of command issued from console for SyzMAIL:

.....

Command:

F SYZMAIL,DebugWTO=Y Modify to turn on detailed Email DEBUG messages

.....

QUERY ALL

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

Parameters:

 All = Display both the Settings and the counters

Default None

Example of command issued from console:

.....

Command:

F SYZMAIL,Query All to display the settings and counters

Response:

The settings and details of current operation are displayed

.....

SENTMSG= **Y/N**

The SENTMSG parameter entry is used to control the messages which SyzMAIL/z normally sends to the OS console which detail the email address or SMS text address that the eMail/SMS package is being sent to. These messages are informational, but some sites do not wish them displayed on the OS console or SYSLOG.

Parameters:

Yes	=	YES
No	=	NO

Default No

Example:

Example of command issued from console for SyzMAIL:

.....

Command:

F SYZMAIL,SENTmsg=Yes Modify to turn on Package delivery messages

.....

STOP

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

Parameters:

None

Default n/a

Example of command issued from console:

.....

Command:

F SYZMAIL,STOP Shutdown SyzMAIL/z.

Response:

.....

WARN= **Yes | No**

NOWARN

The WARN operator command can be used from the operator console (or from within a SyzMAIL/z command member) to turn on (or off) detailed Warning WTO's that SyzMAIL/z can issue under some circumstances. 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:

.....

Command:

F SYZMAIL,Warn=No to turn off Warning WTO messages

Response:

.....

5. Available Cellular gateways

Most carriers have “SMS gateways” which take email messages from the Internet and deliver them to their customers’ cell phones as SMS text messages. The trick is that you need to know what carrier the recipient’s phone is on — it’s not enough to know their phone number. That’s because the carrier determines what the email address of the receiving phone is going to be. For example, Cingular phones’ address are all “something@cingularme.com” while Verizon phones are “something@vtext.com.”

Sound complicated? It’s not. All you really need to do is find your carrier in the list below, and then use the pattern shown there to figure out your email address. If you want to send email to another persons’ s phone, just ask them which carrier they use, and off you go!

For the major carriers, we have included a link to a page with more detailed information on how SMS works with that carrier, how much it costs, and where you can find more information. In the list below, just click any carrier’s highlighted name to find out more.

[Alltel](#)

[10-digit phone number]@message.alltel.com

Example: 1234567890@message.alltel.com

[AT&T](#) (formerly Cingular)

[10-digit phone number]@txt.att.net

Example: 1234567890@txt.att.net

For multimedia messages, use [10-digit-number]@mms.att.net

Example: 1234567890@mms.att.net

[Boost Mobile](#)

[10-digit phone number]@myboostmobile.com

Example: 1234567890@myboostmobile.com

[Cricket Wireless](#)

[10-digit phone number]@sms.mycricket.com

Example: 1234567890@sms.mycricket.com

For multimedia messages: [10-digit phone number]@mms.mycricket.com

Example: 1234567890@mms.mycricket.com

Nextel (now part of Sprint Nextel)

[10-digit telephone number]@messaging.nextel.com

Example: 1234567890@messaging.nextel.com

Sprint (now Sprint Nextel)

[10-digit phone number]@messaging.sprintpcs.com

Example: 1234567890@messaging.sprintpcs.com

T-Mobile

[10-digit phone number]@tmomail.net

Example: 1234567890@tmomail.net

Verizon

[10-digit phone number]@vtext.com

Example: 1234567890@vtext.com

Virgin Mobile USA

[10-digit phone number]@vmobl.com

Example: 1234567890@vmobl.com

Other U.S. and Canadian carriers:

Bell Canada: [10-digit-phone-number]@txt.bellmobility.ca

Centennial Wireless: [10-digit-phone-number]@cwemail.com

Cellular South: [10-digit-phone-number]@csouth1.com

Cincinnati Bell: [10-digit-phone-number]@gocbw.com

Metro PCS: [10-digit-phone-number]@mymetropcs.com or [10-digit-phone-number]@metropcs.sms.us

Qwest: [10-digit-phone-number]@qwestmp.com

Rogers: [10-digit-phone-number]@pcs.rogers.com

Suncom: [10-digit-phone-number]@tms.suncom.com

Telus: [10-digit-phone-number]@msg.telus.com

U.S. Cellular: [10-digit-phone-number]@email.uscc.net

