ENQWATCH

Program Information

C O P Y R I G H T

Disclaimer

Kevin E. Ferguson of **Abbydale Systems LLC.**, has developed this computer program and associated materials.

This material has been used successfully by **Abbydale Systems LLC.** and to the best of our knowledge this material and any system(s) of which it is a part are operational as of the service level or date stated in the body of this material (if so stated). However, *no warranty* is given or implied as to the accuracy of this material or any related material or systems, and *no responsibility* is assumed for any effect or modification directly or indirectly caused by the use of this material.

It is the responsibility of any user of this material to evaluate its usefulness to the user's environment.

Abbydale Systems LLC. does not guarantee to keep this nor any related material current, nor does it guarantee to provide any corrections or extensions described by any users of this material or any corrections or extensions made in the future by **Abbydale Systems LLC.** itself.

Acknowledgements

This document refers to several software products that are produced by other companies. In most cases the names of these products are trademarks and/or copyright of those companies. It is not our intention to claim either the name of the trademark, nor the product itself, these remain solely the right of the owning companies.

CONTENTS

1 Overview

The program, **ENQWATCH**, is based around the IBM supplied program ISGECMON with the features of the David Alcock's ISGECMOM added and massively re-written and some 'bells and whistles' added.

ENQWATCH will only work for dataset conflicts between batch jobs and TSO users. Conflicts between two or more batch jobs are not handled by this program.

It is designed to be used as a long running program, in fact it won't end at all until you issue a stop for, or cancel, it. With this in mind remember that you will need to code TIME=NOLIMIT or TIME=1440 on the EXEC statement.

The purpose of the program is the monitor the system for any dataset contention and if the holder of the resource is a TSO user, it will send them a message requesting that they free the resource.

The added features of **ENQWATCH** include:

- The ability to force an iteration of the loop thereby allowing automation to be written to immediately send a message to a TSO user if they are holding up an important batch run.
- More statistics are available including how many times the iteration was forced by the above feature.
- The program has been modified to allow commands to take effect immediately rather than wait until the loop time limit is met.
- Dynamic adjustment of the wait time limit is now supported.
- The ability to cut SMF records. These will be cut even if the maximum number of notifies has been sent.

ENQWATCH must be run from an authorized library because a STIMER exit is used within it

1.1 Passed Parameters

An optional wait time can be passed to the program. It must be between one and four numbers long. If omitted a default value will be used (60 seconds).

1.2 Called Programs: *None* (Indirectly calls) *None*

1.3 IBM macros used:

CVT	IEZCOM	LTORG	STORAGE
EXTRACT	IHAASCB	MODESET	TIME
GQSCAN	IRAOUCB	QEDIT	TPUT
IEZCIB	ISGRIB	SMFWTM	WAIT
	LOCASCB	STIMER	WTO

1.4 Assembled Program Values

There are six values that currently are only set when the program is assembled. If you need to change any of the first three values, then the program will only pick them up after a restart. These values are located at the top part of the program and they are:

1.4.1 MAXMSGS

This is the maximum number of 'Get the heck out of the dataset' messages that will be sent to single individual TSO users for any given ENQ conflict. The current value for this is 5. This value will not affect the SMF record being produced.

1.4.2 MAXOWNER

This is the value for the maximum number of ENQ conflicts that can be tracked. This value is currently set at 50. Note: only conflicts between TSO users and batch jobs are included.

1.4.3 AREASIZE

This value specifies the size of the area to be used to hold the RIBS and RIBES and should only be changed by a system programmer. This value is multiplied by 1000 in the program. The current value for AREASIZE is 10.

1.4.4 SMFON

This is a flag that denotes whether SMF records are cut when an ENQ notify is issued. 1 means cut SMF records, any other value and no SMF records are cut.

The status can be changed dynamically using the SMF command.

1.4.5 SMFREC

This is the numeric value of the SMF record type to be cut if SMF recording is switched on. The value must be between 128 and 255.

Please be sure to check with your site SMF administrators before setting this value to ascertain which SMF record number to use. Some third-party vendors produce SMF records in the range 128-255 notably Syncsort which cuts SMF type 208 records by default.

The default value for the SMF record number is 223.

1.4.6 WAITTIME

This is the value, in seconds, which the program will wait for between normal iterations of the loop cycle. The current value for WAITTIME is 60 seconds.

Do not exceed 9999 seconds or the program may fail.

The value for WAITTIME can be changed dynamically while the program is running, but obviously if a permanent change is desired this value needs to be changed, the program re-assembled and then restarted.

The current settings for these values can be displayed using the INFO command (F sysname,INFO).

2 Installation Procedure

2.1 *From XMI File*

The XMI (or XMIT) file is in IBM TSO TRANSMIT format and **must** be transferred to z/OS™ as a fixed blocked 80 byte BINARY file. The disk space requirement for the file is 2 tracks of 3390 disk.

The FTP process (if performed in a 3270 emulator) must be performed in TSO READY mode or in option 6 of ISPF™.

The dataset name used as input for the TSO TRANSMIT was ABBYDALE.ENQWATCH.LOADLIB. Unless this is changed by the TSO RECEIVE command it will be the name of the dataset created by the RECEIVE command.

2.2 *By Assembling the Source Code*

To assemble **ENQWATCH** the source code must be saved in a partitioned dataset and assembled using High Level Assembler. (ASMA90).

It should be linked using LIST,LET,NCAL,MAP LOADER options.

2.2.1 Sample Assembly Job

The following is a sample assemble and link JCL the can be used to assemble **ENQWATCH**

```
//STEP1     EXEC PGM=ASMA90,PARM='OBJECT,NODECK',REGION=1024K
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//         DD  DSN=SYS1.AMODGEN,DISP=SHR
//SYSUT1   DD  SPACE=(CYL,(2,1)),UNIT=SYSDA,DSN=&&WRK1
//SYSLIN   DD  DSN=&&WRK5,UNIT=SYSDA,SPACE=(CYL,(2,1)),
//             DCB=(RECFM=FB,BLKSIZE=3200,LRECL=80),DISP=(,PASS)
//SYSIN    DD  DSN=your.input.pds(ENQWATCH),DISP=SHR
//LKED1    EXEC PGM=IEWL,PARM='LIST,LET,NCAL,MAP',COND=(4,LT)
//SYSLMOD  DD  DSN=your.output.load.library(ENQWATCH),DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DSN=&&WRK5,DISP=(OLD,DELETE)
```

2.3 Sample JCL for ENQWATCH

The following JCL will run **ENQWATCH**

Example 1

```
//stepname EXEC PGM=BSLBR14
```

This JCL will assume all of the **BSLBR14** defaults so it will take 20 seconds to run and will return COND CODE 0

Example 2

```
//stepname EXEC PGM=BSLBR14,PARM='RC=16,WAIT=45'
```

This JCL will take 45 seconds to run and will return COND CODE 16.

Example 3

```
//stepname EXEC PGM=BSLBR14,PARM='ABEND=S806,WAIT=30'
```

This JCL will take 30 seconds to run and will crash with a S806 ABEND.

Example 4

```
//stepname EXEC PGM=BSLBR14,PARM='WAIT=5,RC=4'
//SYSIN       DD *
IEF009Z Please ignore this message
```

This JCL will issue the message 'IEF009Z PLEASE IGNORE THIS MESSAGE' then wait 5 seconds before ending with a COND CODE 4.

Obviously as BSLBR14 can issue **any** messages you could issue messages that **BSLBR14** may itself issue so a careful check of the job stream is recommended.

Example 5

```
//stepname EXEC PGM=BSLBR14,PARM='ABEND=U0001,WAIT=2'
//SYSIN          DD *
ABC message
WAIT=20     wait for 20 seconds
XYZ message
```

This JCL will issue the message 'ABC message'. It will then wait 20 wall clock seconds and issue the message 'XYZ message'. It will then wait a further 2 wall clock seconds and then abend with a U0001 abend.

3   Messages

ENQW001I

ENQW001I : Starting - ENQWATCH last assembled on *mm/dd/yy* **at** *hh.mm*

Meaning

This message is issued when **ENQWATCH** is started. The date and time values will be filled in with the actual date and time that **ENQWATCH** was last assembled.

Corrective Action

This is an informational message and no action need be taken.

ENQW002I

ENQW002I - Enqueue monitor active

Meaning

This message is an indication that **ENQWATCH** has obtained all of the storage it needs and successfully established the operator interface and is now waiting on the ECB list. In other words…it is up and running.

Corrective Action

This is an informational message and no action need be taken.

ENQW003I

ENQW003I -- Free of start CIB unsuccessful

Meaning

This message is issued to indicate that **ENQWATCH** had trouble freeing the control interface block for a start command. This is no big deal as we don't use a subsystem start command.

You should never see this message, but it is included for completeness.

Corrective Action

This is an informational message and no action need be taken.

ENQW004I

ENQW004I - Enqueue monitor shutdown in progress.

Meaning

This message is issued to indicate that and stop command was issued for **ENQWATCH** and it is being closed down.

Corrective Action

This is an informational message and no action need be taken.

ENQW005E

ENQW005E - GQSCAN Failed. Retrying.

Meaning

This message is issued to indicate that **ENQWATCH** had a problem issuing the GQSCAN macro to list dataset contentions. The program will attempt a retry at the next wait time expiration.

If the message continues to keep on being issued, then notify the System Programmer.

Corrective Action

This message is an indication that the GQSCAN area is too small to hold all the contentions. If this message keeps on being issued, consider increasing the value for ARAEASIZE and re-assembling the program.

ENQW006E

ENQW006E – Invalid command.

Meaning

This message is issued to indicate that the operator command entered for **ENQWATCH** was invalid or not supported.

Corrective Action

This message is an indication that the GQSCAN area is too small to hold all the contentions. If this message keeps on coming out consider increasing the value for ARAEASIZE and re-assembling the program.

ENQW007I

ENQW007I - ENQWATCH was assembled using these values:.

Meaning

The ENQW007I messages are issued to display the current value of the MAXMSGS, MAXOWNER, AREASIZE and WAIT time.

In the case of WAIT time there is also an indication of how the wait time limit was set (operator command, default or passed parameter).

These messages are issued and startup and as the result of the INFO operator command being issued.

An example of the ENQW007I messages follows:

```
ENQW007I - ENQWATCH was assembled using these values:
ENQW007I - Maxmsgs=5, Maxowner=50, Areasize=10,000
ENQW007I - Wait time=232 Seconds (Set via Parm).
ENQW007I - SMF Recording is ON. Record Number=223
```

Corrective Action

These are informational messages and no action need be taken,

ENQW008I

ENQW008I - Status: Notify=n, Cycles=c, Loops=l

Meaning

The ENQW008I message is issued either as a response to a STATS command or at **ENQWATCH** termination time.

The message is a count of :

- How many times a message was sent to TSO users (The notify value)
- How many CYCLE commands were issued to the program (The cycle value)
- How many times the wait interval triggered a GQSCAN (The loops value)

Corrective Action

These are informational messages and no action need be taken.

ENQW009E

ENQW009E - Invalid WAIT command. No action taken.

Meaning

The ENQW009E message is issued as the result of an invalid wait value being requested by an operator command.

The prior wait value will still be in effect until a successful WAIT command is processed.

Corrective Action

Correct the WAIT command and retry.

ENQW010E

ENQW010E - Invalid PARM passed. Default wait taken

Meaning

The ENQW010E message is issued when the parameter being passed to **ENQWATCH** on the EXEC JCL card is detected to be invalid.

The parameter will be ignored, and the default wait time value used (currently 60 seconds).

Corrective Action

The wait value can be changed dynamically via the WAIT command, but the JCL will need to be corrected in order to make this a permanent change.

ENQW011I

ENQW011I - SMF recording has been switched OFF | ON

Meaning

The ENQW011I message is issued in response to a F ENQWATCH,SMF OFF or SMF ON command.

The status of SMF recording can be displayed by using the INFO command.

Corrective Action

This is an informational message and no action need be taken.

ENQW012E

ENQW012E - Invalid SMF command. No action taken.

Meaning

This message is issued when the program detects that an invalid format of the
F ENQWATCH,SMF command has been issued.

The only options for the SMF command are SMF OFF or SMF ON

Corrective Action

This is an informational message and no action need be taken, however you will need to
correct the modify with the valid format and re-issue it before the desired action is complete.

4 Summary of Amendments

Date	Version	Fix Id.	Comment
15th December 2023	6.0	n/a	Correct messages.
30th May 2017	6.0	n/a	Made program 31 bit addressable.
28th May 2017	5.1	n/a	Transferred to Abbydale Systems LLC.
20th February 2012	5.1	JOB02	Added Replace_Jobname subroutine
19th February 2012	5.0	JOB01	Commented out redundant code

Obtaining Support

Support for, comments about and suggestions for enhancements for this product can be obtained from our website :

www.abbydalesystems.com

or by emailing us at

support@abbydalesystems.com

In order to assist us in filtering support emails please specify in the heading of the email the name of the product that you require support on.

Spam will not be tolerated at this email address.

Where source code is provided for the product, support will be on a 'best efforts' basis. Where the user site has modified the source code, support may entail requesting copies of that sites source code and may result in support being withdrawn if this is not provided.

Abbydale Systems LLC. reserves the right to any code modifications that may have been undertaken at the user site.

Any alteration of the copyright information contained in the original source code is an infringement of the copyright of this and any other Abbydale Systems product and may result in legal action being taken against the perpetrator.