iSeries

Plan for controlled system shutdown

Version 5 Release 3

iSeries

Plan for controlled system shutdown

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 17.

Eighth Edition (August 2005)

This edition applies to version 5, release 3, modification 0 of IBM Operating System/400 (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

Control server shutdown using a power-handling program

A power-handling power program, when used with power protection devices, can minimize interruption during a power loss situation. Power protection (for example, UPS) devices help provide energy to the system when utility power is temporarily interrupted. The energy that is provided helps prevent system functions from ending abnormally. The controlled shutdown mechanisms help the system power down as smoothly as possible, minimizing adverse impacts on re-IPL time.

The following steps can help you get started:

- **Enabling a power-handling program**
  Enabling a power-handling program to control system activity during a power interruption is one way to power down as smoothly as possible.
- **Implementing a power-handling program**
  This example provides step-by-step instructions for you to use when implementing a power-handling program on an iSeries server .
- **Writing a power-handling program**
  Writing a power handling program can help you handle a brief power interruption without doing any unique processing or it can prepare for a normal power down if power is not restored after a brief time period.

You can tailor the CL program examples to your specific system requirements, as well as test the power-handling program that you created.

If you are looking for controlled shutdown concepts and definitions, see “Controlled shutdown concepts” on page 9.

Note: Read “Code disclaimer information” for important legal information.

Code disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Enabling a power-handling program to control system activity during a power interruption

System software support is essentially the same for both the battery feature and the uninterruptible power supply attachment. In some environments you may want to perform different actions when the uninterruptible power supply begins supplying power to the system or when power is fluctuating. A power-handling program can use any of the following methods to handle these situations:

- Sending specific messages to interactive users
- Ending batch jobs and subsystems in preparation for powering down
- Dynamically changing the system values that control uninterruptible power supply processing
- Issuing the PWRDWNSYS command to power down the system

To specify that you have power handling programs, change the QUPSMSGQ system value to the name of a queue you have created. The system will send the same messages to both QSYSOPR and the queue you specified. Change the QUPSDLYTIM system value to *NOMAX.

The program you use to handle the message queue must be active and must allocate the queue. If a program has not allocated the queue that is specified in QUPSMSGQ, the system will assume that no power handling program exists.

Implementing a power-handling program

The following is an example of implementing a power-handling program on an iSeries server when a full uninterruptible power supply is attached. This example assumes that QCTL is the controlling subsystem. For a sample program with a full uninterruptible power supply, go to “Example: Power-handling CL program” on page 5.

1. Because of the critical nature of a power-handling program, you should isolate the objects used by the power-handling program in their own library and secure them from other users, as follows:

   ```
   CRTLIB LIB(UPSLIB) AUT(*EXCLUDE) CRTAUT(*EXCLUDE)
   ```

2. A power-handling program requires exclusive use of a message queue. For this reason, you should create a unique message queue and exclude its use from all other users and general system use, as follows:

   ```
   CRTMSGQ MSGQ(UPSLIB/UPSMSGQ) AUT(*EXCLUDE)
   ```

3. Create the CL power-handling program and exclude its use from all other users, as follows:

   ```
   CRTCLPGM PGM(UPSLIB/UPSPGM) AUT(*EXCLUDE)
   ```

4. Create the job description for the power-handling program you want started automatically whenever the controlling subsystem is started.

   ```
   CRTJOBD JOBD(UPSLIB/UPSJOBD) JOBQ(QSYS/QCTL2)
          JOBPTY(1) RQSDTA('CALL UPSLIB/UPSPGM')
          AUT(*EXCLUDE) USER(xxxxx)
   ```

   Note: You must provide a user profile to use the job description as an auto-start job.

5. Create an alternative controlling subsystem description by making a copy of the current controlling subsystem description, as follows:

   ```
   CRTDUPOBJ OBJ(QCTL) FROMLIB(QSYS)
   OBJTYPE(*SBSD) TOLIB(QSYS)  NEWOBJ(QCTL2)
   ```

6. Modify your startup program to start all subsystems. You will need to include a check to see if system value QCTLSBSD is equal to QCTL2. See system value QSTRUPPGM for the name and library. If you do not modify the startup program it will not check for QCTL2 in QSYS or QGPL and the startup program will end without starting the rest of your subsystems.
7. Add the autostart job entry to the alternative controlling subsystem description, as follows:

```
ADDAJE SBSD(QSYS/QCTL2) JOB(QSYS/QCTL2)
JOBD(UPSLIB/UPSJOBD)
```

8. Change the controlling subsystem system value to use the alternative controlling subsystem description, as follows:

   ```
   CHGSYSVAL SYSVAL(QCTLSBSD) VALUE('QCTL2')
   ```

9. Change the system values to allow the program to handle a power outage, as follows:

   ```
   CHGSYSVAL SYSVAL(QUPSMSGQ) VALUE('UPSMSGQ UPSLIB')
   CHGSYSVAL SYSVAL(QUPSDLYTIM) VALUE(*NOMAX)
   ```

10. Perform an IPL of the system to have the new controlling subsystem description take effect, as follows:

    ```
    PWRDWNSYS OPTION(*IMMED) RESTART(*YES)
    ```

Writing a power-handling program

A power-handling program should be activated at each IPL and remain active at all times. It should be accounted for in the activity level available in work management subsystem specifications.

The message queue that is specified in QUPSMSGQ is used for uninterruptible power supply message processing. The program normally allocates the queue by specifying the command:

```
ALCOBJ OBJ(xxx/yyy *MSGQ *EXCL)
```

When a message arrives, the critical messages to process are:

- **CPF1816: System utility power failed at &1;** (this message applies to the battery feature and full power supply)
- **CPF1817: System power restored at &1;** (this message applies to the battery feature and full power supply)
- **CPI0994: System power is restored** (this message applies to the limited uninterruptible power supply)
- **CPI0963: System on auxiliary power** (this message applies if system power fails during the IPL)

You can choose to ignore the other messages.

Your program can handle a brief power interruption without doing any unique processing. For example, when the CPF1816 message arrives, you can set a switch in your program that indicates that the message occurred. The program could then perform a RCVMSG with WAIT(10) to cause a time-out in 10 seconds. If the CPF1817 message is received before the time-out occurs, you can reset the switch and perform no other action.

Your program can prepare for a normal power down if power is not restored after a brief time period. For example, if you have remote work stations that are still active, you may want to send them a message requesting they sign off quickly. You may want to issue ENDSBS OPTION(*CNTRLD) to prevent new work stations from signing on or new batch work from beginning. If you have batch jobs running, you may want to end them with the following command:

```
ENDJOB OPTION(*CNTRLD)
```

This sets an indicator to end the job. Some higher level languages and the control language allow you to test within a program to see if a controlled ENDJOB was specified. If the program does not end itself, the default on ENDJOB (30 seconds) is used.

You can set a second timer in your program, such as RCVMSG WAIT(120). If utility power has not been restored, you can issue the PWRDWNSYS OPTION(*IMMED) command. The wait time should be specified based on your battery time and the time that is required for a power-down.

If you name a message queue for the QUPSMSGQ system value and *NOMAX for QUPSDLYTIM, the following conditions apply:

- The message queue you specify must be allocated by a program when the CPF1816 message occurs.
- If the message queue you specify is a work station message queue, it must be in a break or notify mode.

If not, the system assumes that no power handling program exists, and the system will be powered down.

Note: When the system has been placed in a restricted state (for example, ENDSBS *ALL), your uninterruptible power supply handling program will no longer be active. For this reason, it is necessary to prepare an alternate method of dealing with your uninterruptible power supply and any possible power interruptions that may occur while your system is in a restricted state.

For example, when performing a SAVSYS (Save System) or RCLSTG (Reclaim Storage), your uninterruptible power supply program will no longer be active once all subsystems have been terminated. Only a single workstation job will be active. You can perform one of the following actions as an alternative:

1. After all subsystems have been ended, from the command line change the mode for the message queue specified in system value QUPSMSGQ to *BREAK. This will cause all uninterruptible power supply messages to be sent as break messages to the user signed on to that work station. With this method the user will manually decide what to do should a power failure occur.
2. Change the system value QUPSDLYTIM to some value other than *NOMAX (for example, the number of minutes you wish the uninterruptible power supply to ride out the power failure). This method will prevent the system from performing an immediate quick power down. However, if a power failure occurs, a quick power down will be performed if the power failure lasts longer than the value specified for the system value QUPSDLYTIM.
3. Modify your existing uninterruptible power supply handling program for use as a BREAK HANDLING program which may be used while the system is in a restricted state. This can be done by creating a second version of your uninterruptible power supply program that does not allocate the message queue specified in system value QUPSMSGQ. (In other words, do not use the ALCOBJ command.) To utilize this program while in a restricted state, prior to starting a dedicated function such as SAVSYS, enter the command:

   ```
   CHGMSGQ MSGQ(LIB/MSGQ) DLVRY(*BREAK)
   PGM(LIB/PGM)
   ```

   where (LIB/MSGQ) is the name the message queue specified in system value QUPSMSGQ, and (PGM/LIB) is the name of your modified uninterruptible power supply handling program. Now, should a power failure occur, the power failure message will be handled by the break handling program, even while a function such as SAVSYS is running. To deactivate the break handling program either have the user sign off or enter:

   ```
   CHGMSGQ MSGQ(LIB/MSGQ) DLVRY(*HOLD)
   PGM(*DSPMSG)
   ```

   Once you have deactivated the break handling program, you should immediately start your subsystems and your normal uninterruptible power supply handling program.

Examples

“Example: Power-handling CL program” on page 5
“Example: Testing a power-handling CL program” on page 8

Example: Power-handling programs

You can tailor the sample program to your specific system requirements, as well as test the power-handling program that you created.

- “Example: Power-handling CL program” on page 5
- “Example: Testing a power-handling CL program” on page 8

Note: Read “Code disclaimer information” on page 1 for important legal information.

Example: Power-handling CL program

You can use the sample program with a full uninterruptible power supply. Although this sample CL program will work correctly as written, you should tailor it to your specific system requirements. For example, add additional recovery to the program by monitoring for error conditions specific to your system. You will also need to supply a user-written program that performs the steps necessary to prepare for a normal shutdown of the system. These steps may include holding job queues, sending messages, and ending subsystems. The program should restart normal operations should the power outage end before the system is powered down.

Note: Read “Code disclaimer information” on page 1 for important legal information.

The program performs the following:

1. The power-handling program retrieves the system value QUPSMSGQ into the variables &LIB and &MSGQ. Although this is not absolutely necessary, it does help to ensure that the correct message queue is allocated each time the program is started. The program then deletes the message queue (if it already exists) and then creates it again. This step helps eliminate clearing the message queue or any problems that might occur if the message queue is damaged.
2. After the message queue has been created, the program must allocate (ALCOBJ command) the message queue exclusively.

   Note: When the system value QUPSDLYTIM is set to *NOMAX, use one of the following methods to allocate the message queue that is specified for system value QUPSMSGQ:

   - Use the command `CHGMSGQ MSGQ(UPSLIB/UPSMSGQ) MODE(*BREAK)`
   - Include the ALCOBJ command within the power-handling program

   You may only use one of the methods that are listed.

   If a user or a program has not allocated the message queue, and a power outage occurs, the system performs an immediate quick power down.
3. At label `A` in the example power handling program, the Receive Message (RCVMSG) command is used to determine what message has been sent to the message queue. The RCVMSG command is also used to determine the amount of wait time (WAIT parameter) throughout the program.

   On line `27.00` of the example power-handling program, the value of the WAIT parameter on the RCVMSG command causes the program to wait 600 seconds (ten minutes). After ten minutes, the program checks to see if a controlled end to the job has occurred (using the ENDSBS or ENDJOB command). This prevents the never-ending program from delaying the ENDJOB or ENDSBS command.

   If you use ENDSBS *IMMED or ENDJOB *IMMED, then this part of the program can be removed. You can change the value for the WAIT parameter on the RCVMSG command to *MAX. The RCVMSG command runs immediately if the system sends a message to the message queue that is specified on the RCVMSG command. This occurs regardless of the value that is specified for the WAIT parameter.
4. If the message that is received by the RCVMSG command is CPF1816 (system utility power failed), the program checks to see if this is simply a short power failure. The program runs a second RCVMSG command with a value of ten seconds for the WAIT parameter (you must decide how many seconds is adequate for your site).

   If the message that is received by the RCVMSG command within the specified ten seconds is CPF1817 (system utility power restored), then power was restored. The program returns to label `A` and starts the cycle again.

   If the ten second limit is reached and no message is received, then the power failure is longer than ten seconds and additional steps are necessary. At this point, you can call a user-written program that performs the following actions:

- The program uses the HLDJOBQ command to hold certain long running batch jobs
- The program notifies unaffected remote users
- The program begins ending jobs and subsystems in an orderly manner

5. At label B, in the example power-handling program, the program attempts to wait-out the power failure. The program retrieves the present time and places this information into a CL variable that is named &START. The present time is used to determine how much uninterruptible power supply time is left.

   A third RCVMSG command runs, and a CL variable that is named &WAIT (that was changed earlier in the program) determines the value for the WAIT parameter. The CL variable &WAIT is the amount of reserve power the uninterruptible power supply can provide. The value for the &WAIT variable at label A should be adjusted to the amount of reserve power that the uninterruptible power supply can provide. (The amount of reserve power is measured in seconds.)

   In the example program, the value of the &WAIT variable is set to 1200 seconds (20 minutes). If message CPF1817 (System utility power restored) is received during that time, then power *has* been restored, and another program can be called to restart normal system operations. The program then returns to label A and starts the cycle again. If message CPF1817 is not sent after 1200 seconds, then RCVMSG returns a blank message ID (not equal to CPF1817). This indicates that power has *not* been restored and an immediate system power down is started. If a message other than CPF1817 is received during this 1200 second wait, the following actions occur:

   a. The program retrieves the present time, and calculates how much of the 1200 second wait period has elapsed.

   b. The program subtracts the difference, and changes the CL variable &WAIT to reflect that amount.

   c. The program returns to label B to use the remaining power that is provided by the uninterruptible power supply.

   This part of the program checks to see if a date change occurs, which is necessary should the power outage occur on a different date.

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...
      PGM
      DCL      VAR(&UPSMSGQ)  TYPE(*CHAR)  LEN(20)
      DCL      VAR(&LIB)      TYPE(*CHAR)  LEN(20)
      DCL      VAR(&MSGQ)     TYPE(*CHAR)  LEN(20)
      DCL      VAR(&MSGID)    TYPE(*CHAR)  LEN(7)
      DCL      VAR(&ENDSTS)   TYPE(*CHAR)  LEN(1)
      DCL      VAR(&WAIT)     TYPE(*DEC)   LEN(6)
      DCL      VAR(&HOUR)     TYPE(*DEC)   LEN(6)
      DCL      VAR(&MIN)      TYPE(*DEC)   LEN(6)
      DCL      VAR(&SEC)      TYPE(*DEC)   LEN(6)
      DCL      VAR(&TIME)     TYPE(*CHAR)  LEN(6)
      DCL      VAR(&START)    TYPE(*DEC)   LEN(6)
      DCL      VAR(&END)      TYPE(*DEC)   LEN(6)
      DCL      VAR(&RESULT)   TYPE(*DEC)   LEN(6)

      RTVSYSVAL  SYSVAL(QUPSMSGQ) RTNVAR(&UPSMSGQ)
      CHGVAR     VAR(&MSGQ) VALUE(%SST(&UPSMSGQ 1 10))
      CHGVAR     VAR(&LIB)  VALUE(%SST(&UPSMSGQ 11 10))
      DLTMSGQ    MSGQ(&LIB/&MSGQ)
      MONMSG     MSGID(CPF2105) /* Message queue not found. */
      CRTMSGQ    MSGQ(&LIB/&MSGQ) TEXT('UPS Power handling +
                        program message queue') AUT(*EXCLUDE)
      ALCOBJ     OBJ((&LIB/&MSGQ *MSGQ *EXCL))

A:    RCVMSG     MSGQ(&LIB/&MSGQ) WAIT(600) RMV(*YES) +
                        MSGID(&MSGID)
      IF         COND(&MSGID *NE CPF1816) THEN(DO)
      RTVJOBA    ENDSTS(&ENDSTS)
      IF         COND(&ENDSTS *EQ '1') THEN(GOTO CMDLBL(ENDPGM))
      GOTO       CMDLBL(A)
      ENDDO

     /* Check to see if this is a short power outage. */
     IF         COND(&MSGID *EQ CPF1816) THEN(DO)
     RCVMSG     MSGQ(&LIB/&MSGQ); WAIT(10) RMV(*YES) +
                   MSGID(&MSGID); /* Wait ten seconds)
     IF         COND(&MSGID *EQ CPF1817) THEN(GOTO CMDLBL(A))
     ENDDO

     /* Power outage was longer than 10 seconds.  */
     CALL       PGM(LIB/PGM) /* User program that prepares +
                        system for possible shutdown. */

```

```
    /* Check to see if this is a long power outage. */
     CHGVAR     VAR(&WAIT); VALUE(01200) /* 20 minutes. */
B:   RTVSYSVAL  SYSVAL(QTIME) RTNVAR(&TIME);
     CHGVAR     VAR(&HOUR);  VALUE(%SST(&TIME 1 2))
     CHGVAR     VAR(&MIN);   VALUE(%SST(&TIME 3 2))
     CHGVAR     VAR(&SEC);   VALUE(%SST(&TIME 5 2))
     CHGVAR     VAR(&START); VALUE((&SEC); + (&MIN * 60) + +
                   (&HOUR * 3600))
     RCVMSG     MSGQ(&LIB/&MSGQ); WAIT(&WAIT); RMV(*YES) +
                        MSGID(&MSGID);
     IF         COND(&MSGID *EQ CPF1817) THEN(DO)
     CALL       PGM(LIB/PGM) /* User program that restarts +
                   system operations. */
     GOTO       CMDLBL(A)
     ENDDO

     IF         COND(&MSGID *NE CPF1817) THEN(DO)
     RTVSYSVAL  SYSVAL(QTIME) RTNVAR(&TIME)
     CHGVAR     VAR(&HOUR)  VALUE(%SST(&TIME 1 2))
     CHGVAR     VAR(&MIN)   VALUE(%SST(&TIME 3 2))
     CHGVAR     VAR(&SEC)   VALUE(%SST(&TIME 5 2))
     CHGVAR     VAR(&END)   VALUE((&SEC) + (&MIN * 60) + +
                 (&HOUR * 3600))
     CHGVAR     VAR(&RESULT); VALUE(&END - &START)
     IF         COND(&RESULT < 0) THEN(CHGVAR VAR(&RESULT) +
                   VALUE(86400 + &RESULT)) /* Check for +
                        change of day.  86400 = 24 hours. */
     IF         COND(&RESULT *GE &WAIT) THEN(PWRDWNSYS +
                 OPTION(*IMMED) /* UPS battery reserve has +
                  expired. */
     CHGVAR     VAR(&WAIT) VALUE(&WAIT - &RESULT) /* UPS +
                        battery reserve has not expired. */
     GOTO       CMDLBL(B)
     ENDDO

ENDPGM:    DLCOBJ     OBJ((&LIB/&MSGQ *MSGQ *EXCL))
           ENDPGM
```

Related examples

“Example: Testing a power-handling CL program”

Example: Testing a power-handling CL program

Once a power-handling program has been created, you can test it by creating a simple CL program that uses the Send Program Message (SNDPGMMSG) command and the Delay Job (DLYJOB) command. Simply set the DLY parameter value on the DLYJOB command to meet your testing needs.

Notes:

1. Read “Code disclaimer information” on page 1 for important legal information.
2. When testing an uninterruptible power supply program, commands such as PWRDWNSYS, ENDJOB, and ENDSBS should be replaced with the SNDMSG command to indicate that the command has run.

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7....
 1.00             PGM
 2.00             DLYJOB     DLY(120) /* Wait for 2 minutes. */
 3.00             SNDPGMMSG  MSGID(CPF1816) MSGF(QCPFMSG) +
 4.00                          TOMSGQ(UPSLIB/UPSMSGQ) /* Power failure
 5.00                          message. */
 6.00             DLYJOB     DLY(5) /* Wait for 5 seconds. */
 7.00             SNDPGMMSG  MSGID(CPF1817) MSGF(QCPFMSG) +
 8.00                          TOMSGQ(UPSLIB/UPSMSGQ) /* Power restored
 9.00                          message. */
10.00             ENDPGM
```

Figure 1. Testing a Power-Handling Program Example

Related examples

“Example: Power-handling CL program” on page 5

Controlled shutdown concepts

If you are looking for controlled shutdown concepts and definitions, start here:

- “Battery backup unit”
- “Handling uninterruptible power supply conditions when no power-handling program exists”
- “IPL considerations for uninterruptible power supply” on page 10
- “Power loss controlled shutdown” on page 10
- “Power restore IPL system value (QPWRRSTIPL)” on page 10
- “Uninterruptible power supply” on page 11
- “Uninterruptible power supply delay time system value (QUPSDLYTIM)” on page 11
- “Uninterruptible Power Supply Message queue system value (QUPSMSGQ)” on page 14
- “Uninterruptible power supply messages” on page 14
- “Weak battery condition signal from uninterruptible power supply” on page 15

Battery backup unit

A Battery Backup Unit (BBU) provides a safety net for users who do not have an uninterruptible power supply or those with a failed uninterruptible power supply. The BBU provides 30 seconds of runtime. If AC power is not restored within 30 seconds, the system immediately goes into a controlled shutdown.

Systems 620, 640, 650, 720, 730, 740, 830, SB1, SB2, SB3 are equipped with a BBU.

Handling uninterruptible power supply conditions when no power-handling program exists

You can specify that you do not have a power handling program by using the default for the QUPSMSGQ, which is QSYSOPR. When you use the default, the system sends all power-related messages to QSYSOPR. You should set the QUPSDLYTIM to something other than *NOMAX.

Normally, you do not supply power to the work station devices. When utility power is interrupted, the system remains active, but the work station jobs usually end abnormally. If utility power is restored while the system is operating on the uninterruptible power supply, the system remains active and the work station jobs can be restarted.

If the QUPSDLYTIM timer ends or the weak battery signal occurs, the system saves main storage and powers down. Select a value for QUPSDLYTIM that is appropriate for your uninterruptible power supply and your system size.

If the system is powered down while on uninterruptible power supply, the QPWRRSTIPL determines whether an IPL is performed when utility power is restored. The default is to not perform the IPL.

IPL considerations for uninterruptible power supply

When the system performs an IPL, the Licensed Internal Code verifies various internal switches to see if the system was correctly powered down. Only the successful completion of the Power Down System (PWRDWNSYS) command causes the iSeries server to be correctly powered down. For any other type of shutdown, the OS/400 program considers the next IPL to be abnormal. The Licensed Internal Code considers the IPL to be normal if the system saves main storage and completes the power-down sequence successfully. If neither power-down technique completes normally, the Licensed Internal Code runs various recovery functions on the next IPL.

When an abnormal IPL occurs, the OS/400 program performs additional recovery functions. In an attended IPL, you can control some of these functions. In an unattended IPL that is caused by the "Power restore IPL system value (QPWRRSTIPL)" or a timed IPL, the system can only use the values that are currently set.

If the Power Down System (PWRDWNSYS) command is run while power is supplied by the Battery Backup Unit (BBU) or uninterruptible power supply, the system delays writing any job logs until the next IPL. The system handles this type of PWRDWNSYS so that the amount of processing is minimized. The system does not perform an IPL while operating on the BBU.

You can perform an IPL on the system if utility power is off and the system is operating on an uninterruptible power supply. This does not apply for a timed or remote IPL. Only a manually initiated IPL is allowed when utility power is interrupted.

For information about uninterruptible power supply messages, see "Uninterruptible power supply messages" on page 14.

Power loss controlled shutdown

The power loss-controlled shutdown mechanism allows the system to power down in an orderly fashion following a loss of utility power. The power loss-controlled shutdown mechanism is available only if you connected the iSeries server to an uninterruptible power supply.

If utility power is not restored within the time that is specified by the QUPSDLYTIM system value, the Licensed Internal Code signals each job to end at the next instruction boundary. Usually a job is at the next instruction boundary or will be shortly. However, some long running instructions, such as those that build access paths or create programs, may not complete in the time that is allowed. After a fixed internal time to reach an instruction boundary, the changed pages in main storage are written to auxiliary storage, the system shuts down abnormally, and then powers off.

Successfully completing a power loss controlled shutdown causes the next initial program load (IPL) of the system programs to be considered normal by the Licensed Internal Code, but not by the operating system. If the uninterruptible power supply batteries do not hold the system long enough for the power loss-controlled shutdown to complete, the next IPL will be abnormal from the Licensed Internal Code viewpoint.

For more information about the QUPSDLYTIM system value, see "Uninterruptible power supply delay time system value (QUPSDLYTIM)" on page 11.

Power restore IPL system value (QPWRRSTIPL)

This value controls what happens if the system ends when utility power is interrupted and then restored at a later time. The default is 0 (Not allowed), which prevents the system from performing an IPL when utility power is restored.

Normally, you would only leave this value set to 0 under one of the following conditions:

- You prefer to manually start the system again
- You have a power handling program that determines whether or not the batteries are recharged enough to allow another IPL.

Uninterruptible power supply

An uninterruptible power supply provides a source of AC power if utility power should fail. Typically, uninterruptible power supplies have a finite backup time.

An uninterruptible power supply supplies power to the iSeries server and all associated DASD controllers and devices during a utility power failure. The runtime of the uninterruptible power supply should be sized appropriately for a power loss controlled shutdown.

Three system values affect an uninterruptible power supply. These system values define the action the system takes in response to a change in the power supply signal when an uninterruptible power supply is attached:

QUPSMSGO
QUPSDLYTIM
QPSRRSTIPL

Uninterruptible power supply delay time system value (QUPSDLYTIM)

QUPSDLYTIM is the system value for the “Uninterruptible power supply” delay timer. The value of QUPSDLYTIM controls the length of time that the system waits before saving main storage and powering down the system. If utility power is restored before the wait time ends, the system resets the timer. If the wait time is exceeded, the system begins to save main storage and goes into a controlled shutdown.

The three choices for the QUPSDLYTIM value are as follows:

- *BASIC or *CALC
- A numeric value
- *NOMAX.

The time that is specified by the QUPSDLYTIM system may be one of the following:

- Infinite if QUPSDLYTIM is set to *NOMAX (must have a power-handling program).
- A fixed internal delay if QUPSDLYTIM is set to *BASIC or *CALC.
- An alternate value that is specified by QUPSDLYTIM. The value must be a positive number, specifying seconds (zero means no delay).

If you set QUPSDLYTIM to *NOMAX, the following conditions must be met or the system begins an immediate power down:

- The message queue specified in the QUPSMSGQ system value must exist.
- If the message queue is a workstation message queue (or QSYSOPR), it must be in break or notify mode.
- If the message queue is not a workstation message queue, it must be allocated by a job.

The figure displays these choices and their implications.

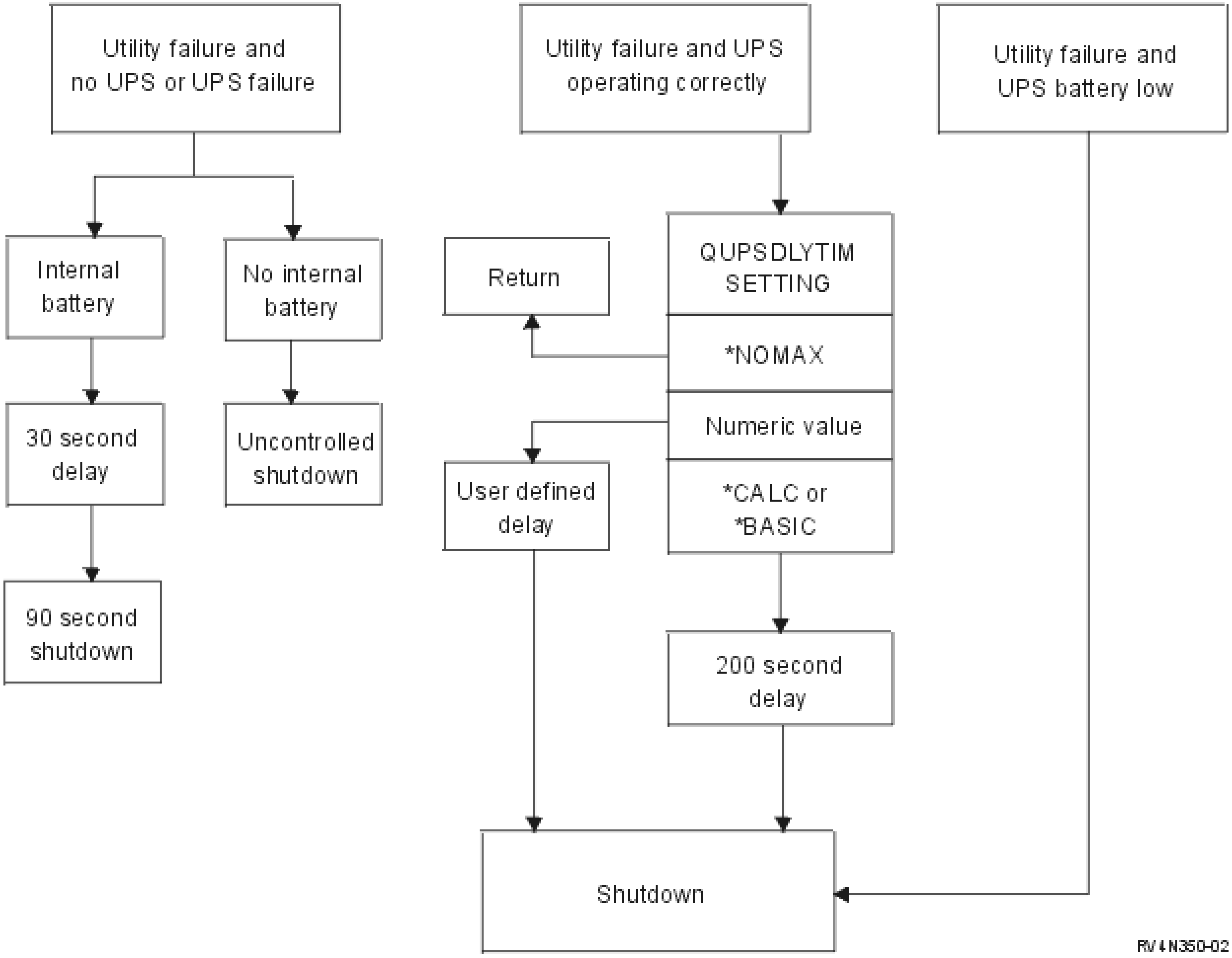

Figure 2. QUPSDLYTIM Values and Actions

The default value for QUPSDLYTIM is *CALC. Leaving QUPSDLYTIM set to *CALC may defeat the purpose of having an uninterruptible power supply. *BASIC and *CALC provide the same function in systems that are running V3R6 or later releases of OS/400.

If QUPSDLYTIM is set to *BASIC or *CALC, the system will perform a controlled shutdown after a fixed interval delay of 200 seconds. If you have an uninterruptible power supply, you can specify a numeric value. See **Determining the Value of QUPSDLYTIM**.

Determining the Value of QUPSDLYTIM

To determine the value of QUPSDLYTIM you need to know the following:

- The hold-up time of your uninterruptible power supply at the given load (the battery-rated time).
- The amount of time it takes your system to save main storage and power down.

Available Battery Runtime

The available battery runtime is a dynamic variable. Evaluate your system's available battery with these questions:

- Has the power been fluctuating?

- Has the battery been previously discharged?
- Is the battery fully charged?

The actual battery runtime is a function of capacity. Even if the battery is fully charged, it may not have 100% capacity. A typical battery will lose 20% to 50% of its rated capacity in 4 to 5 years, depending on ambient room temperatures. Elevated operating temperatures tend to increase the loss of capacity. The actual battery runtime is also a function of the discharge load. The more loads the UPS serves the less time it can sustain them. When the battery on the system falls below a specific level of charge, the UPS will issue a weak battery condition signal. The “Weak battery condition signal from uninterruptible power supply” on page 15 affects the shutdown mechanisms.

The time it takes to save main storage and power-down is not an exact number. Save time depends on the number of changes in main store that have not been written to disk. The number of disk arms available is also a factor; the more disk arms, the faster the system can write main storage to disk. The system power-down will also depend on the number of jobs and the average amount of time it takes to end them. Usually a job will be close to an instruction boundary; however, some instructions are long running. The following formula shows a worst case scenario in estimating the number of minutes it will take to shut down a system:

```
((0.0554*(MS size in MB))/(# of disk arms)) + 1.6 = # of minutes
```

Refer to the following table for an estimate of time required to write main storage to disk on your particular system. The typical amount of time required to shut down a system will usually be a small percentage of this value.

Table 1. Time needed in minutes to write main storage to disk

Arms	32MB	64MB	128MB	256MB	512MB	1GB	2GB	4GB	8GB	16GB	32GB	64GB	128GB
2	2.5	3.4	5.1	8.7	15.8	30.0	58.4	115.1	228.7				
4	2.0	2.5	3.4	5.1	8.7	15.8	30.0	58.4	115.1	228.7			
8	1.8	2.0	2.5	3.4	5.1	8.7	15.8	30.0	58.4	115.1	228.7		
16	1.7	1.8	2.0	2.5	3.4	5.1	8.7	15.8	30.0	58.4	115.1	228.7	
32	1.6	1.7	1.8	2.0	2.5	3.4	5.1	8.7	15.8	30.0	58.4	115.1	228.7
64		1.6	1.7	1.8	2.0	2.5	3.4	5.1	8.7	15.8	30.0	58.4	115.1
128			1.6	1.7	1.8	2.0	2.5	3.4	5.1	8.7	15.8	30.0	58.4
256				1.6	1.7	1.8	2.0	2.5	3.4	5.1	8.7	15.8	30.0
512					1.6	1.7	1.8	2.0	2.5	3.4	5.1	8.7	15.8
1024						1.6	1.7	1.8	2.0	2.5	3.4	5.1	8.7
Assumes all pages in main storage have been modified and changes have not yet been written to disk. Proportionately less time is required if fewer pages need to be written to disk.													

Explanation for QUPSDLYTIM values and actions flow chart

The flow chart shows the values and actions of the QUPSDLYTIM system values. The flow chart starts with three boxes across the top of the chart.

The first box on the top is labeled ″Utility failure and No UPS or UPS failure.″ Two arrows proceed from this box. One arrow points to a box labeled ″Internal battery.″

The other arrow points to a box labeled ″No internal battery.″ The box that is labeled ″Internal battery points to another box labeled ″30 second delay.″ The box that is labeled ″30 second delay″ points to a box that is labeled ″90 second shutdown.″

The box that is labeled ″No internal battery″ points to a box that is labeled ″Uncontrolled shutdown.″

The second box on the top is labeled ″Utility failure and UPS operating correctly.″ An arrow from this box points to another box that is labeled ″QUPSDLYTIM SETTING.″ This box shows the three settings for the QUPSDLYTIM setting, which are *NOMAX, numeric value, and *CALC or *BASIC, respectively.

If *NOMAX is selected, an arrow points to another box that is labeled ″Return.″ If numeric value is selected, an arrow from the QUPSDLYTIM SETTING box points to another box that is labeled ″User defined delay.″ Another arrow from the ″User defined delay box points to a box that is labeled ″Shutdown.″ If *CALC or *BASIC is selected, an arrow from the QUPSDLYTIM SETTING box points to another box that is labeled ″200 second delay.″ The 200 second delay box points to the box that is labeled ″Shutdown.″

The third box is labeled ″Utility failure and UPS battery low.″ This box points to the box that is labeled ″Shutdown.″

Uninterruptible Power Supply Message queue system value (QUPSMSGQ)

The Uninterruptible Power Supply Message Queue (QUPSMSGQ) system value determines which message queues the power supply messages are sent to.

The system sends messages about the power supply to the system operator (QSYSOPR) message queue regardless of the value that is specified in the system value. If you specify a different message queue, that message queue also receives the same power supply messages. Specify a different message queue if you have one of the following:

- Another message queue you want to receive the power supply messages (for example, the data processing manager's message queue)
- A program that handles events that are related to the uninterruptible power supply

Uninterruptible power supply messages

Message ID	Message Text	Additional Information
CPF1816	System utility power failed at &1;	
CPF1817	System power restored at &1;	The system power switched to the utility source.
CPF1819	System ending. Power failure message not monitored. System ending for reason &3. Note 1: Message queue &1 in library &2 specified by QUPSMSGQ system value is not allocated to a user program or workstation. Note 2: Message queue &1 in library &2 is allocated to a workstation or system operator message queue (QSYSOPR in library QSYS), but the message queue is not in *BREAK or *NOTIFY mode.	
CPI0961	Uninterruptible power supply (UPS) no longer attached.	
CPI0962	The uninterruptible power supply (UPS) is now attached.	
CPI0963	System on auxiliary power.	System is currently running on auxiliary power.

Message ID	Message Text	Additional Information
CPI0964	Weak-battery condition exists.	The external uninterruptible power supply (UPS) or the internal battery indicates a weak-battery condition. If utility power fails during this condition, the system may begin an immediate power down. See your uninterruptible power supply manual for more information.
CPI0965	Failure of battery backup feature in system unit.	There may be a failure of the battery or the battery charger for the battery backup feature in the system unit. Contact your service representative.
CPI0966	Failure of battery backup feature in expansion unit.	There may be a failure of the battery or the battery charger for the battery backup feature in the expansion unit. Contact your service representative.
CPI0973	Weak battery condition no longer exists.	The weak-battery condition for the external uninterruptible power supply or the internal battery no longer exists. See your UPS manual.
CPI0974	UPS has been bypassed.	If a utility power failure occurs, the uninterruptible power supply cannot supply system power. The system will end abnormally.
CPI0975	UPS no longer bypassed.	The uninterruptible power supply (UPS) is no longer bypassed.
CPI0976	Notification of message &1; failed.	Unable to send &1; message to message queue &2; in library &3; specified in QUPSMSGQ system value.
CPI0981	Automatic IPL disabled.	Automatic IPL after utility power restored, specified by system value QPWRRSTIPL, was disabled for one of the following reasons: • Utility power failed and the battery weak condition was detected during the IPL. • Utility power failed during the IPL and the uninterruptible power supply delay time specified in system value QUPSDLYTIM was exceeded before the IPL complete.
CPI0994	System power is restored.	The system power switched to the utility source at &1; The utility power failed for &2; seconds. During this time the system was not doing any application processing. If the utility power continues to fail, power down the system (PWRDWNSYS command).

Weak battery condition signal from uninterruptible power supply

When the battery on the system falls below a specific level of charge, the “Uninterruptible power supply” on page 11 will issue a weak battery condition signal. The weak battery signal from the uninterruptible power supply will cause the system to perform a power loss controlled shutdown if the following conditions are true:

- You have chosen to use the four wire communication between the iSeries server and a UPS. See Planning for hardware and software for more information.

- The utility fail signal is active.

The controlled shutdown occurs immediately. If the system is running on utility power and the uninterruptible power supply sends a weak battery signal, the system remains up and posts a CPIO964 message. For more information about this message, see “Uninterruptible power supply messages” on page 14. However, the system initiates a shutdown mechanism immediately under this condition if utility power is lost.

A typical factory-preset time for an UPS to send a weak battery signal is with approximately two minutes of runtime remaining. Some UPS models have an adjustable setting for this time. Ideally, you would want to set it for the amount of time it would take for your system to perform a power loss controlled shutdown. Do not assume that the factory preset time on the UPS is sufficient for a normal shutdown of your system. Calculate the amount of time necessary to perform a power loss controlled shutdown and use an appropriate value for QUPSDLYTIM and adjust the time for weak battery signal on the UPS (if adjustable).

The figure shows the time progression of the QUPSDLYTIM function, from left to right.

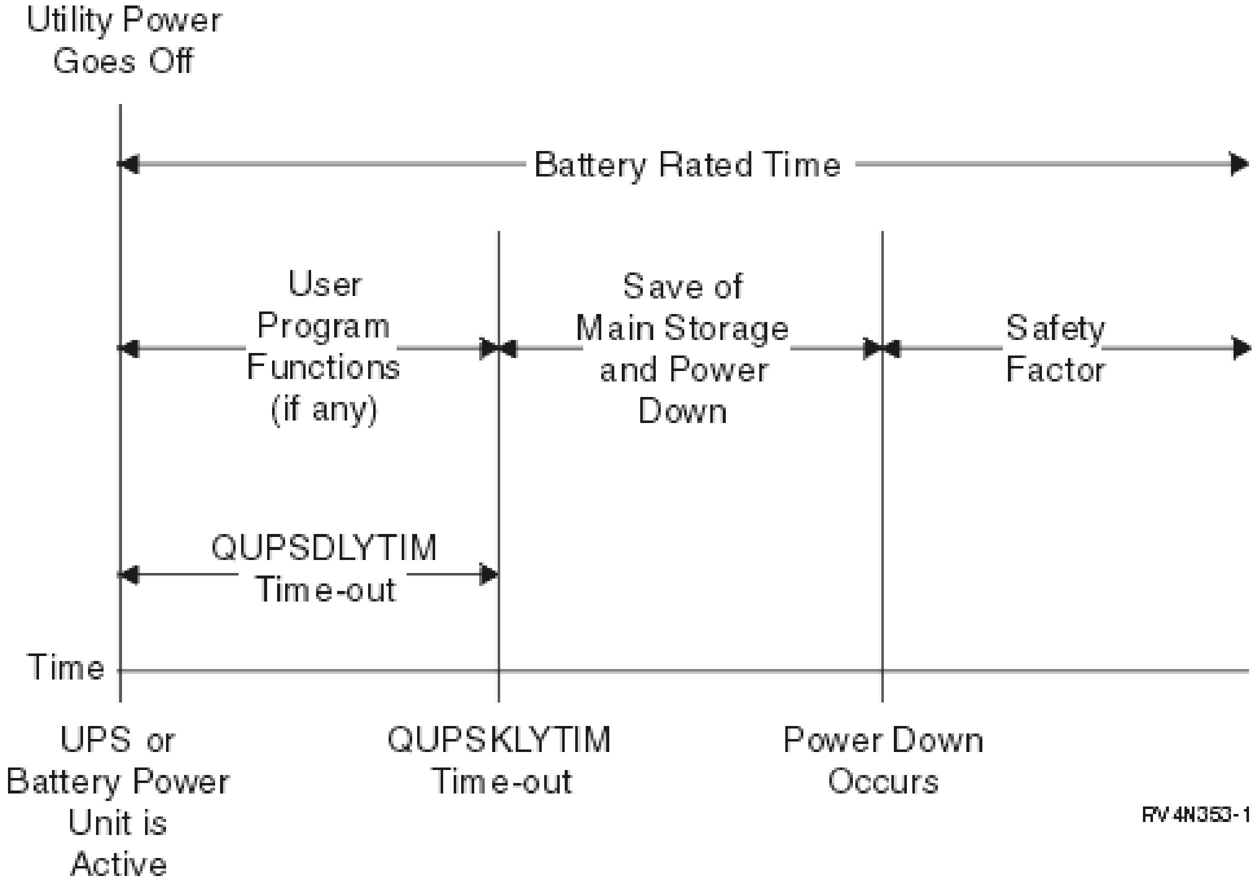

Figure 3. Time Line of QUPSDLYTIM Function

Explanation of illustration for time line of QUPSDLYTIM function

The time line shows what happens from the time the power goes out to the time the system shuts down. The duration of the time is the same as the rated life of the battery. The time line starts at the point that utility power goes off. At the same time the UPS or battery power unit is active. At this time user program functions (if any) and the QUPSDLYTIM value time-out is in effect. When that time expires, main storage is saved, then power down occurs. After that there should be some time left of battery life as a safety factor.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This (ADD NAME OF PUBLICATION HERE) documents intended Programming Interfaces that allow the customer to write programs to obtain the services of (ADD PRODUCT NAME HERE).

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Application System/400
AS/400
e(logo)server
eServer
IBM
iSeries

Operating System/400
OS/400
400

Lotus, Freelance, and WordPro are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing publications

Permissions for the use of the information you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

Personal Use: You may reproduce this information for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of this information, or any portion thereof, without the express consent of IBM®.

Commercial Use: You may reproduce, distribute and display this information solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of this information, or reproduce, distribute or display this information or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the information or any data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the information is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THIS INFORMATION. THE INFORMATION IS PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

All material copyrighted by IBM Corporation.

By downloading or printing information from this site, you have indicated your agreement with these terms and conditions.

Printed in USA