

Reduce Rerun Time with Efficient Restart

- Restart Db2 Batch Applications Near the Point of Failure
- Modify COMMIT Frequency in Real Time to Balance Performance and Risk
- Industry's Easiest Implementation

Business Challenges

Today's company is expected to provide 24/7 data availability to customers and business partners. This, in addition to managing ever growing Db2 databases, makes it essential for most businesses to execute and complete their batch processes each evening. In the event of a production job failure, companies often risk millions of dollars of revenue due to unavailable Db2 production applications. Quickly restarting and completing failed batch jobs is critical to maintaining normal business operations.

Restart and Complete Db2 Batch Jobs Fast

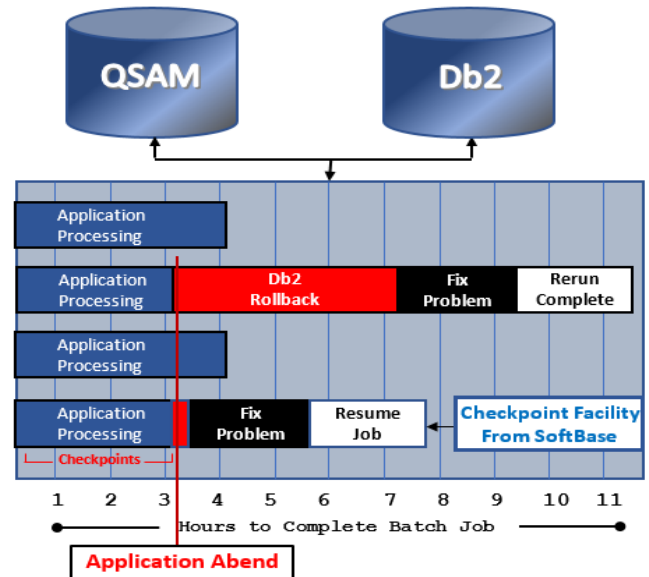
Batch jobs that do not regularly COMMIT their changes to Db2 risk excessive rollback times **should** they fail. Then, even after Db2 removes all work applied by the failing job, the original program must be rerun in its entirety after all problems have been corrected. Also, batch jobs that do not ever COMMIT database changes often use excessive Db2 resources and run longer than necessary.

Enhance Batch Job Performance

When Db2 batch applications perform COMMITs, it will often speed up those programs and also free up Db2 resources for other production jobs that are running at the same time.

SoftBase's Db2 Checkpoint will automatically add COMMITs to your Db2 batch programs, and allows you to quickly restart any abended production job from the point of failure. Even if there are no COMMITs in the application, checkpoints can be triggered by program actions such as number of records read or written to a file or by certain SQL statement executions.

SoftBase's Db2 Checkpoint will reestablish all positioning and resources within the problem program, including COBOL Working Storage and QSAM file access; then it completes any remaining application processing rather than rerunning the whole Db2 program.



An application with no COMMITs can impact the batch processing window. It can sometimes cost hours of valuable batch processing time, often resulting in outages for other applications or preventing the initialization of online systems for the daily start of business.

Balance Speed and Risk

Db2 Checkpoint also provides a powerful Variable COMMIT Frequency Option that enables you to dynamically tune Db2 batch jobs by changing their COMMIT frequency as they run. Database Administrators can now dramatically speed up any jobs that need to be completed as soon as possible prior to the online Db2 applications being restarted.

```

***** CANDESCENT SOFTWARE LLC ***** Row 1 to 10 of 10
*          CHECKPOINT RESTART MONITOR          *
*          SUBSYS - DBCG      CKPT TABLE - _default_      *
*****
Command ==>
Scroll ==> PAGE

  JOB      STEP  CHKPTS  TOTAL  VAR  SAVED  JOB  JOB
CMD NAME  NAME  TAKEN   LUW's  FREQ STORAGE RESTART ACTIVE
-----
S  CSB6FDAX  FILVRDW  0000009  0000009  000000320
  CSB6TLTH  CKP6BASE  0000463  0000463  000001184  Y
  CSB2AS2   CKP2BASE  0000001  0000500  000001168
  CSBCCASE  CKPCBASE  0000246  0000246  000001184
  CSBCREAD  STEP1     0000025  0000025  000000568  Y
  CSBCTLN   CKPCBASE  0000621  0000621  000001184
  CSBCTLO   CKPCBASE  0000619  0000619  000001184
  CSBCTLR   CKPCBASE  0000744  0000744  000001184
  CSBCTLT3  CKP6BASE  0000017  0000017  000001184
  CSBCTLTH  CKPCBASE  0000122  0000122  000001184
***** Bottom of data *****

```

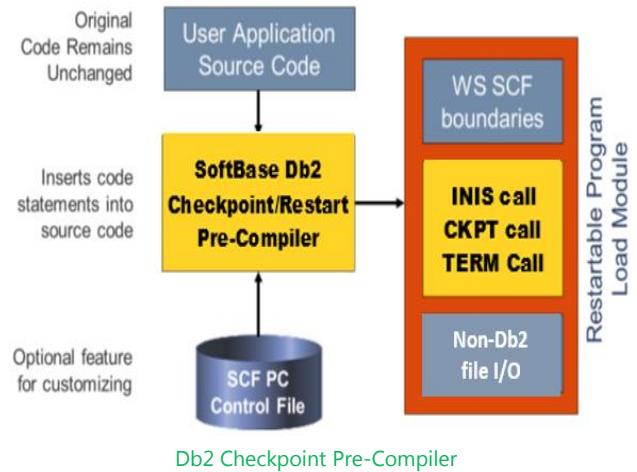
The Db2 Checkpoint Monitor is an ISPF dialog that allows users to monitor all Db2 Checkpoint jobs that are either executing or awaiting restart. Working storage areas and file buffers may be viewed using this monitor.

Seamless Integration

SoftBase's Db2 Checkpoint incorporates a PL1 and a COBOL Pre-Compiler which can insert the appropriate function calls into your PL1 or COBOL source code to implement Db2 Checkpoint. A Pre-Compiler control file can place the function calls at the appropriate LUW locations you specify.

The COBOL **Db2 Checkpoint Pre-Compiler** works equally well for applications which have already been modified for legacy checkpoint restart solutions such as third party or homegrown solutions. The Pre-Compiler easily converts existing applications for use with Db2 Checkpoint.

Db2 Checkpoint's transparent interface allows those applications coded to use IMS or other checkpoint products to use the Db2 Checkpoint with no coding changes.



Db2 Checkpoint Pre-Compiler

Features and Benefits

- Allows application to restart from the most recent checkpoint, rather than from the beginning
- Allows you to specify how frequently you want to create checkpoints and COMMITs while the application is running. Users can adjust settings to prioritize between checkpoint overhead, processing time, and restart time
- Allows you to suspend a batch job and free up resources for different applications, and resume batch processing at your convenience
- Can detect resource unavailable and deadlock timeout conditions and automatically restart, avoiding an abend
- Enables you to effectively save job progress prior to a scheduled outage and quickly resume afterward
- Provides applications with a single, unified restart solution as opposed to restart logic coded directly into applications
- Improves resource availability via commits
- Allows applications to be updated and compiled between restarts