

Restart Without Starting Over: Ensure DB2 on z/OS, IMS, and VSAM Availability

TABLE OF CONTENTS

- WHY CHECKPOINTS MATTER 1

- WHAT IS CHECKPOINT RESTART? 1

- BMC APPLICATION RESTART CONTROL 1
 - » BMC APPLICATION RESTART CONTROL for DB2 2
 - » BMC APPLICATION RESTART CONTROL for IMS 3
 - » BMC APPLICATION RESTART CONTROL for VSAM 3
 - » Restart Applications Easily 3
 - » Application checkpoint/restart 3
 - » How checkpoint/restart processing works when a batch job starts normally. 4
 - » How checkpoint/restart processing works when a batch job is restarted after a failure. 4
 - » *Checkpoints in action - Insurance Company (AMB Generali)* 5

- CHECKPOINT/RESTART AND VSAM FILE SHARING 5
 - » *Checkpoints and VSAM in action - Medical Testing Company* 6

- CHECKPOINT PACING. 6
 - » *Checkpoint Pacing in action - Telecommunications Company.* 6

- APPLICATION REATTACH 6
 - » *Application reattach in action - Automobile Club.* 6
 - » *Deadlock prevention in action - Insurance company.* 7

- SUSPEND-AND-RESUME PROCESSING 7
 - » *Suspend-And-Resume processing in action - Manufacturer.* 7

- CONCLUSION 8

WHY CHECKPOINTS MATTER

You have worked diligently to ensure that your databases perform optimally and meet ever-increasing availability requirements. But what do you do when you have problems with batch jobs? How do you handle problems like application failures in the middle of a job, batch applications that use too many CPU resources, and jobs that do not take a quiesce or quiet point so that you can begin another process?

Application checkpoints can resolve all of these problems:

- » Application restart – Batch applications can lock portions of databases that they are using. If an application is restarted from the beginning after a failure, then the database is unavailable for the amount of time that the application originally ran, plus the amount of time that it takes the application to run again and complete. With checkpoint restart, the total amount of time that locks are held is significantly less, because the application does not need to reprocess data that it successfully completed.
- » Application performance/reduce CPU – Some older applications take too many checkpoints and use CPU resources unnecessarily. These applications were written to deal with older hardware limitations, and they could run more efficiently without all those checkpoints. Automated checkpoint pacing can improve CPU usage.
- » Suspend-and-resume processing – You may need to suspend a batch jobs to allow another process to take control of a database for some reason.

Taking a checkpoint releases the resource locks that the database management system (DBMS) or other subsystem has placed on the database or data set. Resource locks prevent other processes from accessing the data while an application program is updating it. When the program commits the changes by issuing a checkpoint request, it signals that other processes can safely access the data. Until the locks are released, the data is unavailable, resulting in performance problems. Releasing resource locks is required for most online processes and recommended for many batch application programs. Timely release of resource locks greatly improves data availability.

While checkpoints are valuable, it is not easy to implement them. You must change application code, and you must account for the overhead that checkpoints will bring.

WHAT IS CHECKPOINT RESTART?

When a batch job fails, it must be restarted at the point of failure or at the beginning of the job (after recovery of affected databases and files). It is not practical to back out everything and start from the beginning because backing out updates can take twice as long as running the application. Nor is it practical to restart from the wrong point, then back out errors and start over again.

If an application fails, will you have time to start the job from the beginning and still get it completed in the batch window? If you are using DB2, you may have batch jobs running continuously. If you are using IMS, you may have batch messaging programs that run all the time. If a batch job or batch messaging program fails, can you afford to back out changes and start the job from the beginning?

BMC APPLICATION RESTART CONTROL

BMC APPLICATION RESTART CONTROL (AR/CTL) provides automated checkpoint services that you can implement with no code changes to your batch applications. The checkpoint pacing features help you manage the overhead of checkpoints and eliminate those that require excess CPU resources.

BMC provides the following BMC APPLICATION RESTART CONTROL products:

- » BMC APPLICATION RESTART CONTROL for DB2
- » BMC APPLICATION RESTART CONTROL for IMS
- » BMC APPLICATION RESTART CONTROL for VSAM

BMC APPLICATION RESTART CONTROL enables you to resume failed or interrupted batch applications from the most recent checkpoint rather than from the beginning of the job step. Faster restarts minimize contention between batch jobs and online processing. You can often implement checkpoint/restart functionality with no changes to application code or jobs.

Ensuring data integrity is the primary reason for automating restarts. Manually searching for the last checkpoint is time-consuming and error prone. Restarting from the wrong checkpoint can result in errors, duplicate entries, omitted updates, or other problems. Recovering from an inaccurate restart can be even more complex and confusing than the original problem. BMC APPLICATION RESTART CONTROL products select the right checkpoint every time, eliminating errors and the need to restart your restart. BMC APPLICATION RESTART CONTROL products provide the following features:

- » Automatic restart checkpoint selection - ensures integrity and shortens restart time
- » Application working storage - can capture and restore an application program's working storage areas in main memory, which allows the program to resume processing at the last checkpoint. It can capture and restore saved areas of virtual storage for subprograms executing under the main program.
- » Application reattach - improves the operational stability of many batch application environments by providing automation to react to certain types of abend conditions. Abends often result from lock contention. Many times, application reattach makes it possible to schedule update processes to run in parallel rather than serially.
- » Checkpoint and restart coordination for DB2, IMS, and CICS/VSAM restarts – coordinates checkpoint restart for applications that span DBMSs
- » Automatic checkpoint – simplifies and speeds the process of implementing checkpoint/restart logic into application programs
- » Program exception handling – automatically redirects “bad” input data that causes S0C7 abends into a reject file and lets the application continue. You can clean up the redirected records later.
- » Flat files – automatically manages flat files and ensures that the contents of the files are synchronized with database activity when checkpoints are issued. During restart processing, the files are automatically repositioned to their state as of the latest checkpoint.
- » Suspend and resume processing – enables you to reorganize an IMS or DB2 database online without canceling or interrupting BMPs or DB2 batch jobs, take image copies and establish quiesced recovery points without canceling BMPs. Suspend and resume processing works with the following BMC solutions:
 - BMC Backup and Recovery Solution for IMS, BMC IMAGE COPY PLUS, and BMC RECOVERY MANAGER for IMS
 - BMC MAXM Reorg/Online for IMS
 - BMC Fast Path Online Restructure/EP
 - BMC Database Performance for DB2, BMC REORG PLUS for DB2

Let take a closer look at the BMC APPLICATION RESTART CONTROL solutions.

BMC APPLICATION RESTART CONTROL FOR DB2

BMC APPLICATION RESTART CONTROL for DB2 provides the following features:

- » SQL return code handling - can intercept a defined SQL return code received during application program processing and issue a user-defined user abend code and reason code. This can be used to standardize -911 processing throughout an entire application environment.
- » Cursor repositioning - any checkpoint restart solution can effectively save working storage, but only BMC APPLICATION RESTART CONTROL for DB2 can return the application to the proper position within the cursor. You no longer need to add logic to your DB2 applications to track and store the cursor position for use in checkpoint restart.
- » Batch Attachment facility - Performs the attachment to DB2 on behalf of the application; it can run in an “Attach Only” mode to provide the DB2 attach for programs not using Checkpoint/Restart services. This makes it easier to migrate batch applications from IMS to DB2.

- » IMS Compatible Mode – Removes the need to run under a full IMS batch environment just to use the call attach facility (CAF) to DB2 under IMS. It substitutes all IMS process work and thus is cost effective. The CAF is a substitute for batch TSO (IKJEFT01) when the purpose is to attach to DB2. BMC APPLICATION RESTART CONTROL provides a way to use this attach to DB2 even if no checkpoint/restart services are used. Both ways help to achieve up to 30% savings in elapsed time.

BMC APPLICATION RESTART CONTROL FOR IMS

- » Restart with no code changes –fully supports and enhances the IMS Extended Restart Facility, Requires no application code or JCL changes and eliminates the need to change application code to call a third-party restart program.
- » Flat file management –supports and manages IMS GSAM files and native file techniques; there is no need to convert flat files to GSAM.
- » Checkpoint management – externally filters excessive checkpoint activity to provide significant savings in elapsed time and CPU consumption. Many legacy applications were developed to run on slower processors and the checkpoint intervals were never recalibrated for hardware upgrades. This will save CPU resources.
- » DBRC conversion aid –can automatically provide a logging environment to avoid having to retrofit DL/I JCL when converting an application to run under DBRC.

BMC APPLICATION RESTART CONTROL FOR VSAM

- » Local VSAM access services for VSAM data sets that are accessed exclusively by a batch VSAM application program – provides checkpoint support and automatic backout support for VSAM files, normally not available.
- » DBMS synchronization – Automatically synchronizes VSAM Checkpoint/Restart activity with DB2 or IMS checkpoint processing. This removes the complexity of manual synchronization between different data sources, and increases the availability in restart situations.
- » VSAM file sharing - Supports remote VSAM file sharing between batch applications and CICS regions executing on the same or different z/OS images. This allows batch application programs to update VSAM files while they are online to CICS and in full update mode and makes it possible to avoid converting a VSAM file to DB2 or IMS to provide 24x7 type access to the file.

RESTART APPLICATIONS EASILY

BMC APPLICATION RESTART CONTROL allows applications to “restart without starting over” after a failure. With shrinking batch windows and a high emphasis on quickly resuming business after an outage, BMC APPLICATION RESTART CONTROL provides higher availability and significant cost savings.

If anything stops a batch application before it completes, you must start the job from the beginning. All the work that was done before the failure is lost, and all the time it took to do that work is wasted. BMC APPLICATION RESTART CONTROL solves this problem by taking checkpoints that save information the application uses and coordinating with database sync point processing. With BMC APPLICATION RESTART CONTROL, the application can restart from a checkpoint instead of starting over from the beginning of the job.

APPLICATION CHECKPOINT/RESTART

One strategy you can use to prepare for problems is to take application program checkpoints in preparation for possible restart. BMC APPLICATION RESTART CONTROL provides a framework for building a restartable application environment, with a wealth of supporting functions and features to ensure data integrity, data availability, and wise use of resources. BMC APPLICATION RESTART CONTROL is the ultimate tool for implementing application checkpoint/restart in a variety of application program environments.

Checkpoint/restart support allows a failed application program to restart at the last checkpoint rather than the beginning of the job step. Restarting from a checkpoint reduces the amount of processing that must be done if an application program or system failure occurs. This processing not only includes the amount of work that must be done over again to ensure that it is correct; it also includes the work required to reverse (back out) the changes that were not committed when the failure occurred. Depending on the environment, backing out the changes can take almost twice as long as making the changes originally. If backout is not available to the application program, the databases that the program changed must be recovered to a point before the start of application program execution.

HOW CHECKPOINT/RESTART PROCESSING WORKS WHEN A BATCH JOB STARTS NORMALLY

The following steps explain the checkpoint/restart process:

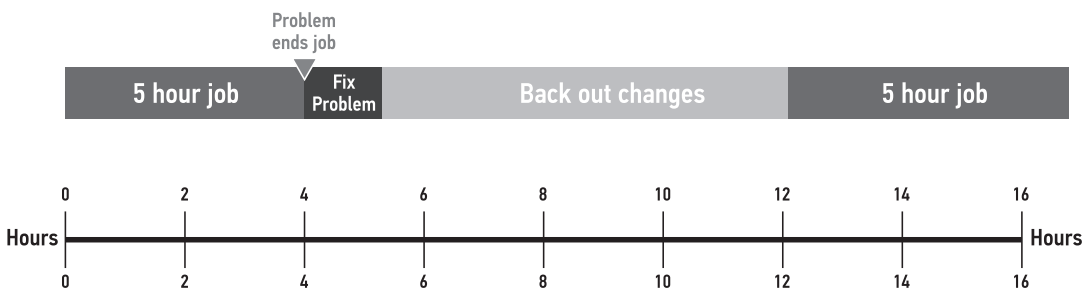
1. When you submit a job step to execute an application program, the program initializes its working storage areas.
2. The program issues a request for restart services. In response to the restart request, BMC APPLICATION RESTART CONTROL sets up the restart environment. BMC APPLICATION RESTART CONTROL automatically determines whether the job step is a normal start or a restart.
3. The application program issues the first checkpoint request. In response to the checkpoint call, BMC APPLICATION RESTART CONTROL writes a checkpoint record to the dynamically allocated checkpoint data set. The checkpoint record contains the information needed to restore application program working storage areas and to reestablish position within the databases and the BMC APPLICATION RESTART CONTROL-supported data sets (VSAM and sequential data sets that the program accesses through standard calls).
4. BMC APPLICATION RESTART CONTROL invokes database management systems (DBMSs) for commit processing.
5. The application program continues processing and making calls to work with databases and BMC APPLICATION RESTART CONTROL supported data sets.
6. At the end of a unit of work (UOW), and before normal termination, the program issues a checkpoint request to save working storage and repositioning information and invoke the DBMS for commit processing.
7. If an abend occurs, the DBMS backs out (reverses) uncommitted changes. If a system failure occurs, the DBMS backs out uncommitted changes during a warm start or an emergency start.

HOW CHECKPOINT/RESTART PROCESSING WORKS WHEN A BATCH JOB IS RESTARTED AFTER A FAILURE

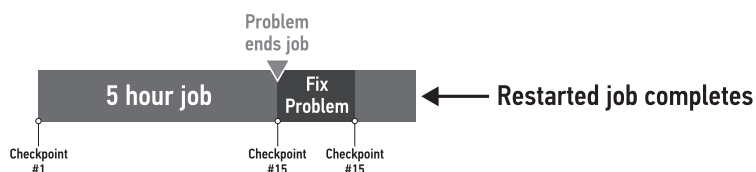
With BMC APPLICATION RESTART CONTROL, an application program can be resubmitted after a failure and will start processing again at the last completed checkpoint.

8. The application program is submitted at the proper step for restart and the program performs its initialization processing.
9. The program issues the restart request. In response to the restart request, BMC APPLICATION RESTART CONTROL automatically detects that the program requires restart, restores the working storage areas and information needed to reestablish database position, and reestablishes the program's position within the BMC APPLICATION RESTART CONTROL-supported data sets.
10. Application program processing resumes at the last checkpoint.

Application Without Restartability



Application With Restartability



CHECKPOINTS IN ACTION - INSURANCE COMPANY (AMB GENERALI)

An insurance company (AMB Generali, Germany) had a problem with their year-end batch runs. In December they needed to send a letter to each customer stating the price of the annual premium beginning in January. Many customers were set up to pay the premium automatically, but the payments reflected the price of the old policy.

For several years, the batch job that generated the letters with the new premium price took more than 30 hours to complete, and the jobs had some space-related abends. Therefore, the letters were not sent in time for customers to change their automatic payment settings. The delay in processing time is expensive and cost the company approximately 300,000€ per day in lost interest.

With BMC APPLICATION RESTART CONTROL, the company was able to split the job into more manageable parts, and they were able to handle upcoming abends and avoid outages. In the first year, the company saved 400,000€ in lost interest by using BMC APPLICATION RESTART CONTROL.

CHECKPOINT/RESTART AND VSAM FILE SHARING

Because BMC APPLICATION RESTART CONTROL is already participating in the access to VSAM data sets, BMC APPLICATION RESTART CONTROL can meet another critical need in a CICS/VSAM environment—the need to share VSAM files between CICS and batch application programs. File sharing allows a non-CICS application program to access and update a VSAM file while the file remains available in the CICS region. BMC APPLICATION RESTART CONTROL dynamically intercepts the native VSAM requests issued by the application program, transforms the requests to online CICS commands, and ships the commands to the CICS region for service. The CICS region logs changes made by those transactions and automatically backs out changes in case of failure.

With the intelligent checkpoint/restart services of BMC APPLICATION RESTART CONTROL, file sharing with BMC APPLICATION RESTART CONTROL is not bound by the limitations of other file sharing products. Batch programs often execute for a long time and update large quantities of data. With BMC APPLICATION RESTART CONTROL, a file-sharing batch application program can issue checkpoint requests at logical points during processing. Checkpoint processing saves data set repositioning information and commits updates, releasing the locks on the changed records and making them available to other CICS users. Without release of locks, CICS performance and data availability can be severely degraded.

Some file sharing products release locks arbitrarily, without concern for the consequences of a failure. If application program changes are made available to other users before a non-restartable program completes successfully, a program or system failure can cause serious data availability problems and possible data integrity exposures. Combining VSAM data set processing with IMS or DB2 database processing further complicates a failure situation. Through a lengthy and difficult process, you must find a point of consistency when all processing was complete and correct; then you must restore the environment to that point, if possible. After time-consuming and error-prone activities, the failed program and all other programs that were executing at the time of the failure must start processing again from the point of consistency.

With BMC APPLICATION RESTART CONTROL, the application program initiates checkpoint processing at the completion of logical units of work. Locks are released only when the program indicates that the changes it has made are complete and correct. If a failure occurs, CICS backs out uncommitted changes so that partial or erroneous changes are never made available to other users. No data recovery is necessary, and you restart the original failed program at the last checkpoint.

You can see significant savings in the application development area by using the VSAM file sharing functionality instead of converting VSAM files to DB2 or IMS. If you convert the files to DB2, you must also convert everything that touches the file, including all of the CICS transactions. File sharing enables you keep existing applications intact because it only converts only the access technique of the few programs that actually update a VSAM file. The rest of the application programs can remain untouched.

CHECKPOINTS AND VSAM IN ACTION - MEDICAL TESTING COMPANY

LabCorp, a medical testing company, has applications that span DB2 and VSAM files. They needed to coordinate checkpoint/restart synchronization issues between DB2 and VSAM. By using BMC APPLICATION RESTART CONTROL, their application program can issue a DB2 commit and BMC APPLICATION RESTART CONTROL will synchronize the VSAM files to that same point. If the application fails, the in-flight VSAM updates are backed out to the same point as the corresponding DB2 files.

CHECKPOINT PACING

The benefits of checkpoint/restart processing are tremendous, but taking checkpoints is costly, both in time and resources. Significant processing overhead is associated with every checkpoint. While an application program must take checkpoints frequently enough to prevent lengthy backouts and numerous resource locks, it must not take them so frequently that performance declines. A balance between overhead and benefits can be achieved, but most companies do not have time for the almost endless fine tuning of the checkpoint frequency to achieve this balance. Further, the optimum frequency can change depending on the time of day that the application program is executing.

BMC APPLICATION RESTART CONTROL provides complete control over checkpoints. Because it operates at the system level, rather than application program level, it can filter, or pace, checkpoints based on criteria defined externally to the application program. The program can issue checkpoints at every unit-of-work boundary without concern for performance.

CHECKPOINT PACING IN ACTION – TELECOMMUNICATIONS COMPANY

In some short BMPs, a large telecommunications company was seeing as many as 50 checkpoints per second. The checkpoint rates were application dependant, and the rates were set high to lessen system contention.

The company needed to reduce the level of checkpoint rates. They implemented BMC APPLICATION RESTART CONTROL for IMS without making any application code changes.

BMC APPLICATION RESTART CONTROL for IMS reduced the checkpoint frequency and yielded CPU savings of 8-18%.

APPLICATION REATTACH

To reduce the need for operator intervention in certain abend conditions (such as deadlocks), BMC APPLICATION RESTART CONTROL can reattach an application program instead of terminating when these abends occur. To z/OS, the job step is still executing; however, to the DBMS, the application program has abended. The DBMS backs out the uncommitted changes made by the application program. BMC APPLICATION RESTART CONTROL can reposition the supported data structures when BMC APPLICATION RESTART CONTROL reattaches the application program. This function increases the availability of critical batch services when running into DBMS-specific error conditions.

APPLICATION REATTACH IN ACTION - AUTOMOBILE CLUB

An automobile club in Germany tows cars that have broken down on the road. The price for the road service is much lower for club members than it is for non-members. To get the member price, most car owners sign a membership form when they need road service.

Road service companies send their invoices to the automobile club immediately (because they want to be paid immediately). However, membership applications are sent only once a week.

When the automobile club application processes the road service company invoice, the application rejects it because no membership record is in the database (the membership form has not been processed yet).

Without BMC APPLICATION RESTART CONTROL, the job abends and the technicians must manually edit the input data stream and the job to submit a next time.... for each missing membership record.

With BMC APPLICATION RESTART CONTROL, the process is handled automatically with the application reattach feature, saving time and preventing errors.

DEADLOCK PREVENTION IN ACTION - INSURANCE COMPANY

An insurance company has a very large, complex IMS environment. They implemented BMC APPLICATION RESTART CONTROL for IMS on 1,400 BMPs on an application that performs batch enrollments. While an individual can enroll themselves online, batch enrollments are used when the company is doing mass updates of the enrollment information for a company or large group of customers.

If something causes an outage, the outage affects the customers as well as customer service:

Most of the time, however, the costs caused by outages are not directly calculated and not highly visible. Without automated reattach or restart, it could take 10-60 minutes to resolve the problem. When many abends occur on a typical business day, the costs are considerable.

Immediately after the successful implementation of BMC APPLICATION RESTART CONTROL, the abends were eliminated. The average time to resolve the deadlock abends is about 2 hours, and the company averaged 400 abends a month.

SUSPEND-AND-RESUME PROCESSING

Suspend-and-Resume processing temporarily suspends execution of certain batch jobs at a checkpoint. The job step remains active, but other resources are released and allow other BMC products to obtain exclusive use of the database. BMC APPLICATION RESTART CONTROL then automatically restarts application processing from the completed checkpoint. Suspend-and-Resume processing is automatic and transparent to applications, DBAs, operators, and end users.

Suspend-and-Resume processing enables you to complete the following tasks when you use the appropriate BMC solutions:

- » Reorganize IMS databases and DB2 table spaces online with read/write access
- » Take snapshot copies of IMS databases online with read/write access
- » Ensure quiesced recovery points for IMS databases with no outage

If you do not use BMC APPLICATION RESTART CONTROL to temporarily suspend batch jobs that run concurrently with an online database reorganization, snapshot copy, or quiesce, any of the following can happen:

- » Batch jobs may abend or never start.
- » The reorganization or image copy job may abend or need to be canceled to allow the batch jobs to complete.
- » Batch jobs must be rescheduled so that they do not interfere with the reorganization, image copy, or quiesce.

SUSPEND-AND-RESUME PROCESSING IN ACTION - MANUFACTURER

One of Europe's leading commercial vehicle manufacturers, MAN Nutzfahrzeuge AG (MAN), is processing thousands of database transactions while maintaining continuous availability of its online systems. MAN uses r for IMS to maintain access to its online systems throughout Europe, South Africa and Australia while simultaneously reorganizing databases with BMC MAXM Reorg/Online for IMS.

With BMC APPLICATION RESTART CONTROL, MAN is able to process BMPs while reorganizing their databases. BMC APPLICATION RESTART CONTROL allows for suspend-and-resume processing, essentially allowing database administrators/ managers to run database reorganizations and batch applications at the same time. The benefits include better application performance, data availability, and data integrity; and decreased development time.

Typically, the execution of online reorganizations requires a short interruption to database access. If BMP applications are running at the time the interruption is required, the online reorganization will be forced to wait until the BMP has completed. This can take up a long time, and will reduce the ability to run reorganizations during BMP applications by a lot.

With BMC APPLICATION RESTART CONTROL in place, the BMP is automatically suspended for the duration of outage and then automatically resumed. BMC APPLICATION RESTART CONTROL eliminates the need for a manual intervention and eliminates any outage during the BMP. The database is reorganized when it is needed to.

"This automatic suspend-and-resume is great! It was very easy to integrate all of our BMC APPLICATION RESTART CONTROL jobs," stated Bernhard Miller, system administrator for MAN. "The quick, easy preparation and the successful run were terrific. Another big advantage of this solution is how it reduces the burden on our operational staff."

CONCLUSION

As these scenarios illustrate, the BMC APPLICATION RESTART CONTROL solutions enable continuous availability of applications and databases. As the batch windows shrink, BMC APPLICATION RESTART CONTROL — in conjunction with other BMC solutions — make it easy to execute batch applications safely and without costly interruptions.

Business runs on IT. IT runs on BMC Software.

Business thrives when IT runs smarter, faster, and stronger. That's why the most demanding IT organizations in the world rely on BMC Software across both distributed and mainframe environments. Recognized as the leader in Business Service Management, BMC provides a comprehensive and unified platform that helps IT organizations cut cost, reduce risk, and drive business profit. For the four fiscal quarters ended December 31, 2009, BMC revenue was approximately \$1.90 billion.

Visit www.bmc.com for more information.

