White Paper

Control-M

Application Continuity Version 9.0.18

Table of Contents

Introduction

Document Purpose

Application Continuity in the production environment is a critical issue that faces every modern organization. This document aims to provide information regarding what Application Continuity is and why it is important. It also reviews the available solutions for achieving Application Continuity in Control-M, as well as guidelines for their implementation.

The target audience for this document is software consultants, technical support staff, and customers. Since it is imperative that this document is kept as updated and as accurate as possible, any information or remarks are welcomed and can be sent directly to “CustomerSupport@bmc.com”.

This document includes examples of configuration and implementation of Application Continuity solutions. These examples are based on practical experience gathered in the lab and the field. Not all of the examples were tested according to the official standards, and these examples should therefore be treated as recommendations only, with limited liability.

What is Application Continuity?

Application Continuity is defined as the means taken by an organization to ensure that critical applications, such as Control-M, continuously function under all circumstances. When the failure is localized and limited to an application in a given location then we will refer to the term **High Availability** as a means to ensure Application Continuity. For Control-M, High Availability (HA) is defined as the means taken to minimize the frequency and duration of downtime in the production environment in a specific site with no data loss to scheduling and/or Workload Automation. Downtime can be caused by either unexpected hardware or software failures, or by planned events, such as scheduled maintenance like software version updates. The target of any organization that wants to establish High Availability levels is to maintain the lowest possible number of unexpected downtime incidents and the shortest possible planned or unplanned downtime.

With HA solutions in place, organizations can ensure Application Continuity against any disaster that occurs within a given location. However, if a disaster completely destroys a primary site, then the organization must have a contingency plan that replicates the primary site with a secondary site some distance away. The planning for a contingency site in a situation where the primary site has been completely disabled will be referred to as planning for **Disaster Recovery** (DR). With Version 9, in a DR scenario, the recovery of the Control-M infrastructure can be immediate.

Organizations do not always require both HA and DR solutions. If the distance between primary and secondary sites is within the technical limitation for High Availability solutions, then organizations can reduce the expense

of building both HA and DR within their organization. Technology has improved to a point where the ability to build an HA environment across hundreds of miles is more cost effective for most organizations.

Why is Application Continuity Important?

The importance of Application Continuity lies in the profound effect it has on modern business success. Modern regulatory and availability requirements make the need for Application Continuity an integral part of today's IT Infrastructure. The trend towards globalization (and the Internet in particular) and cloud services has brought availability issues to the forefront — even for industries that did not pay much attention to it in the past. Many organizations that not so long ago defined "acceptable downtime" as a few hours, today aim to reduce downtime to minutes or even seconds. Recurring and lengthy downtime has a direct effect on critical business aspects, including:

- Revenues are lost during downtime.
- Employee productivity is reduced.
- Customer satisfaction and confidence is sometimes lost.
- Technical support expenses increase.

Consequently, today's organizations are investing in solutions that ensure their applications' availability. However, implementing and maintaining High Availability is a major expense and must be included in the cost of the applications services IT provides. Usually captured in Service Level Agreements (SLAs), cost justification includes weighing the potential loss resulting from downtime and impact on any regulatory or audit requirements. Thus, each organization must determine the required level of availability for each of their applications and associated services and select the option best suited to their needs.

Control-M as a High Availability Application

Control-M is a best-of-breed workload automation solution that addresses business needs by focusing on applications, integration, and management of Workload Automation. It integrates automated workflows and batch processes running on diverse platforms and applications into a unified business process, ensuring service management requirements and smooth operations. Since Control-M triggers critical application jobs, its availability level must be extremely high — any downtime may affect major business services that involve numerous other critical applications. *Figure 1* (page 5) presents the basic architecture of the Control-M product, which is comprised of several components that can each be installed on different machines.

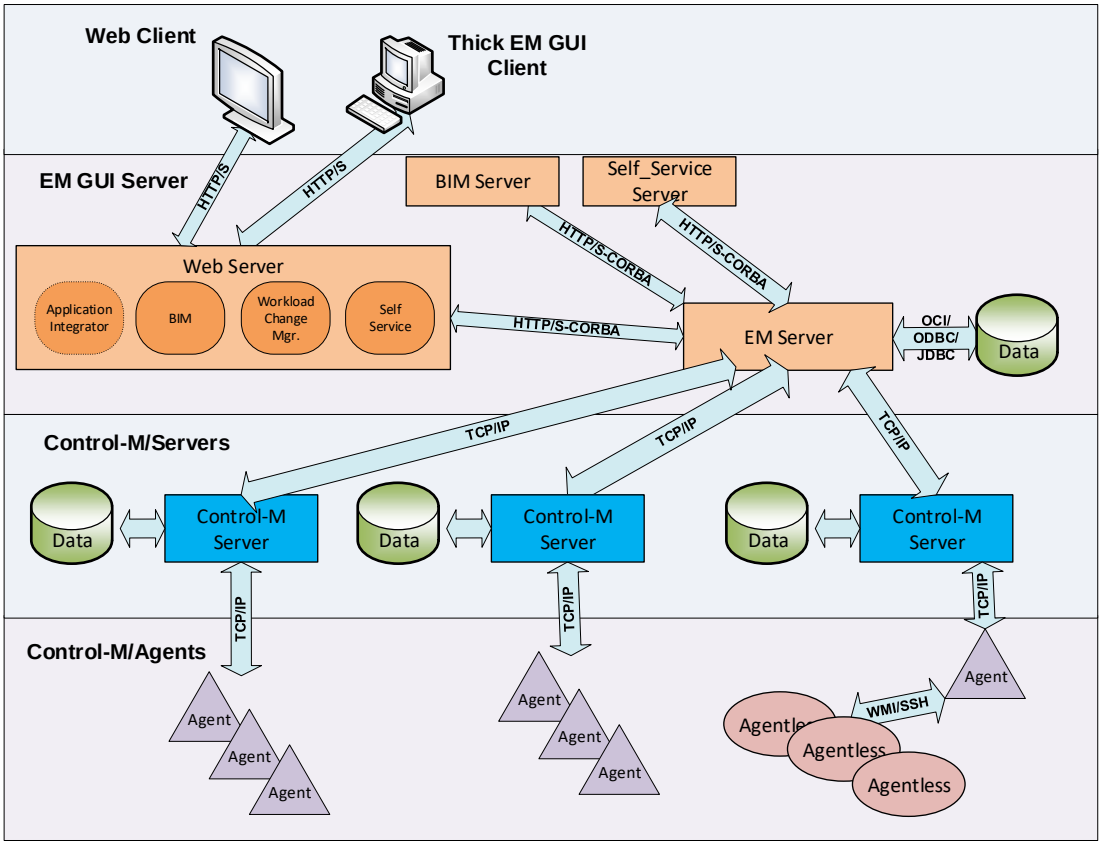

Figure 1: Basic Control-M Architecture

This flexible and scalable architecture is designed in a way that if one component on one machine fails, or the communication fails, the other components in the system continue to provide service based on the most recent available data, when possible.

In this document, Application Continuity refers to all possible means available to keep Control-M working on its vital task — Workload Automation and job scheduling. From a hardware perspective, Control-M makes extensive use of databases and, as such, is vulnerable to disk failures. Application Continuity can be approached from several angles, and each angle can be implemented autonomously. In most cases, the integration of these efforts will maximize the availability of the system. Every organization should choose the configuration that best suits its environment, needs, and resources.

Document Contents

This document reviews the available solutions in each of the following areas, and discusses and reviews the integration of each solution with Control-M:

- **Hardware Solutions**: Reviews common available fault tolerance systems, which overcome CPU, memory, network, and disk failures.
- **Operating System (OS):** Discusses Virtual machine and OS management, including OS clustering solutions for all supported platforms, which minimize the downtime after hardware or software failures.
- **Database Solutions**: Reviews database vendor solutions that are currently available in the market for maintaining database stability and their possible integration with Control-M.
- **Control-M HA Configuration** – Discusses the built-in Control-M High Availability solution and its integration with Control-M.
- **Control-M/Server Database Mirroring**: Discusses an inexpensive method to provide some level of High Availability for Control-M and its integration with other BMC solutions.
- **Proactive Notification Tools**: Presents the proactive notification solutions available from BMC and other vendors, which integrate with Control-M either to enable prevention of downtime or to rapidly identify a failure so that manual intervention can take place.
- **Best practices for Backing up Control-M**: Discusses the recommended way to backup the database irrespective of what solution you implement for High Availability or Disaster Recovery.

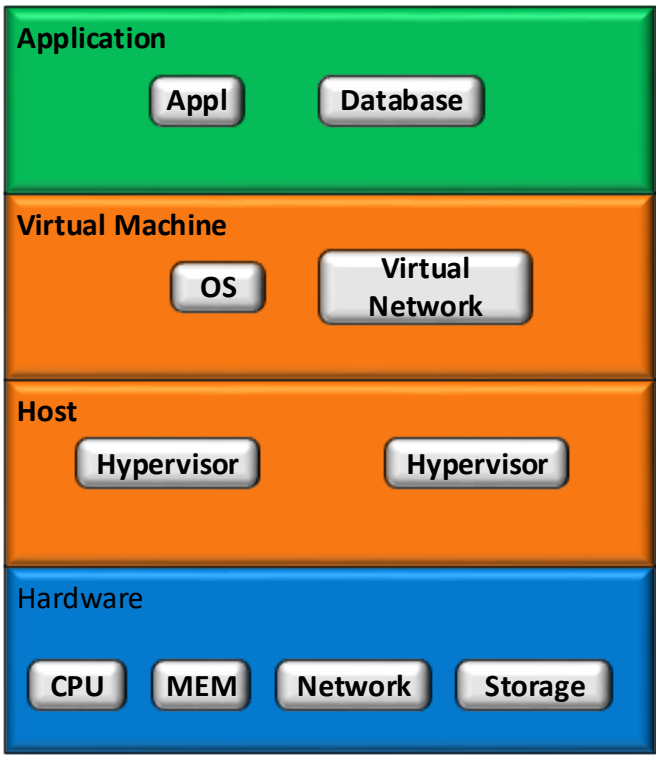

Figure 2: Typical Application Stack

The Diagram above shows the layers of a typical application "stack" in today's modern datacenter. This model uses four layers and shows the abstracted components in each layer.

From a Control-M perspective the main consideration is, just as with other applications, the types and effects of the layers below the application and database are virtualized and abstracted from the actual hardware. This is by design, and provides additional levels of availability. The physical or environmental requirements for the application and database are mapped to the virtual machine, instead of the physical hardware.

An often overlooked initial consideration is the need for bi-directional communication between Control-M components in both the hardware (physical) as well as virtual network.

Hardware Solutions

The traditional fault tolerant model relies on specialized hardware to detect a hardware fault and instantaneously switch to a redundant hardware component — whether the failed component is a processor, memory board, power supply, I/O subsystem, or storage subsystem.

Disk STORAGE Replication

Disk replication is a fault tolerance solution that is transparent to the user. It provides a reliable replication of the environment that can be used in specific failover scenarios. There are many vendors that provide real-time data replication, such as EMC2 with SRDF to Hitachi True Copy, DRDB for Linux®, and IBM® PPRC. These vendors support different operating systems and different levels of High Availability storage, such as RAID 1+0, RAID 0+1, and RAID 1=0+0. High Availability storage elevates the availability of applications that use it.

Control-M Implementation with Disk Storage Replication

Data replication can be used to replicate the entire Control-M environment from one machine to the other only in situations where the secondary machine assumes identically all machine attributes. These attributes include, but are not limited to:

- Platform and OS level must be identical between primary and secondary machines.
- Kernel parameters must be identical.
- OS user definition, group, and its internal IDs must be identical.
- Home directory of the OS user and its environment must identical.
- Hostname should be the same (or *see Knowledge Base for Control-M/Enterprise Manager and Control-M/Server for instructions on what changes are required when renaming the hostname*).

Data replication, when it meets the requirements listed above, are supported for both HA and DR configurations. The contents of the Control-M account can be replicated for either the application files only or it can be replicated for both application files and database files. The database file replication must follow the database vendor's guidelines and supported practices for real-time data replication. Specific database vendor documentation should be consulted on instructions on how to implement real-time disk replication. It is very

important to follow guidelines specified by the database vendor — especially since data files used by the database server are constantly updated and the data consistency must be adhered to (keeping in mind that Control-M's consistency is based on a committed transaction). In most cases, when the database file replication is set to synchronized mode, the data consistency is protected.

Because the guidelines for real-time data replication (RDR) vary greatly between the Control-M application and third-party databases, when using RDR for meeting failover demands, the application and database components should be documented and maintained separately in the overall strategy.

Host and Virtual Machine[1]

Virtualization technology provides a layer of abstraction between the computing, storage, and networking hardware, and the software that runs on it. This technology enables users to run additional operating systems in multiple sessions via a Hypervisor called virtual machines (VMs).

BMC welcomes the running of its products in the leading virtual environments. The VM environment can be configured to provide additional levels of availability independently from the Control-M configuration. In general BMC products are fully supported running under virtual environments. For additional information please see the Virtualization policy on our web site.

http://www.bmc.com/support/policy-for-virtualized-platforms.html

Virtual OS clustering

Operating System Clustering

General

Cluster computing architecture is used in many organizations to address expanding IT needs and requirements for High Availability. This architecture is based on deploying groups of computing resources, including CPUs, memory, disks, and network interfaces. Cluster computing addresses these current IT issues by providing increased service availability, increased scalability, better resource management, and improved manageability.

A cluster consists of two or more machines connected together with specialized hardware and software that work together to form a seamless single machine. Disk storage is usually external and accessible to both machines, although that is not necessarily the case. Each machine is referred to as a node. Machines do not

have their own unique users; adding users to the cluster makes the user login available everywhere. Neither does each machine require its own user licenses; these are shared between the nodes. Each node has its own IP address, and the cluster itself has a virtual IP address. In most cases, the servers in the cluster are interconnected using a faster network interface, such as Fiber Channel connection, for better performance. This network is called the Heartbeat Network.

The cluster software monitors the condition of the nodes, and whenever it identifies a failure in CPU, memory, network, disk, or application, it addresses another node to provide the required service. In addition to the failover feature, clustering provides for workload balancing across the computer. *(Please note that this feature is not discussed in this document).*

The cluster management software can vary between simple, custom, in-house programs and large products offered by the major vendors. The hardware can range from a PC to a Parallel Sysplex mainframe.

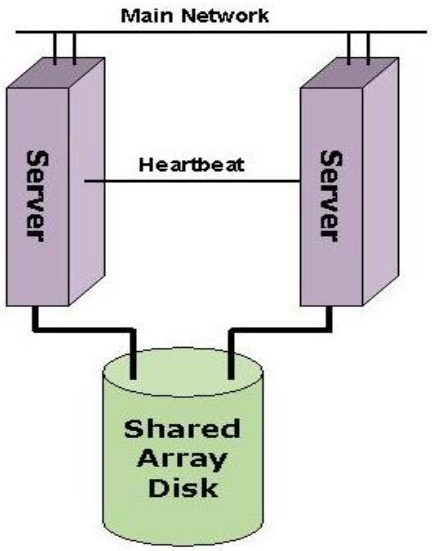

Figure 3: Basic Cluster Architecture

Failover Scenario in Clusters

The most common High Availability solution in cluster environments is a **hot standby server**. A hot standby server is a host that is dedicated to providing all of the services of another host. The host that is the primary provider of the services is called the primary server. Every server that is part of a cluster is called node. In case of a problem in the primary server, the hot standby server takes over control and provides the services of the primary server. The process of taking other node's services is called failover or takeover.

The failover process always starts with the detection and identification phase of the failure condition. The failure may occur at the hardware level, at the OS level, or even at the application level. An example of a failure can be a memory problem, system panic, or a database server crash. The detection and identification can be performed manually or automatically. Console automation products and frameworks can perform the automatic detection internally using cluster management software or externally using outboard automation products.

One of the most typical examples of a failover process is the recovery scenario from a complete cluster node crash. The heartbeat monitor of a console automation product detects the lack of response by one of the cluster's nodes. The console automation verifies that it is not just a network timeout, but also a hardware failure on this specific node, and starts the failover process.

The system administrator defines failover scenarios for each failure condition, using the vendor-provided cluster management software package. When the failure condition is identified, its corresponding failover scenario is triggered. Failover triggering can be performed manually by the operator, internally using the cluster built-in mechanisms, or by outboard automation products.

In this typical scenario of a complete node crash, the designated backup node receives a command that triggers the failover process. As a result, the following steps occur:

- The backup node turns on one of its additional network interfaces and assigns the failed node IP address to it. This enables the backup to respond to client service requests, which are addressed to the failed node IP.
- All of the previously available services and applications are started on the backup node.

In most cases, this transition is transparent to the end users and client applications, which typically experience the transition as an application timeout. In some cases, however, users may be required to restart their connections.

Furthermore, in most cases, the manual recovery process is performed during the least critical time period for the business.

Control-M/Server cluster support is limited to Active/Passive cluster configuration. For a typical Control-M failure scenario, the production resumes from the point where it stopped. The installation directory and the database reside on the shared drive and are accessible to the standby server. The database contains the most recent state of production at the moment of failure, and the standby server serves as a substitute for the primary server in all aspects. All jobs that were executing on the host that encountered a failure are aborted. Jobs that were running on the agents are not affected at all by the server’s failure; they continue to run and their connection to Control-M/Server is reestablished when the standby machine has reinitialized the Control-M application.

Each of the Control-M components can be configured as a separate cluster application group. The entire application group is treated as an entity, and if any resource in the application group fails, the entire application group is taken off line and restarted on the second node. In a Windows platform Control-M integrates with Windows Cluster API to install application and configure it as an out of the box clustered application. For Unix platforms Control-M provides ready script required for any Unix cluster to manage any standard application. When Control-M runs in a cluster the EM components cannot be managed by CMS and instead the Cluster manages the components through Control-M provided scripts.

For Control-M implementation and guidelines for installing in UNIX, Linux, or Windows environments, please see the Control-M Installation Guide chapter on configuring in a cluster environment.

Application Layer

Database Solutions

Database Replication

Database Replication is a software solution that enables the user to define tables and parts of tables to be replicated to another database. The term master is used to describe the instance of the database that is being updated. Replication can either be Multi-Master, where each database can be updated, or Master Snapshot, where there is one master database and the changes are propagated to read-only snapshot sites. In addition, the replication can be set up in two modes:

- **Synchronous mode**: Each transaction ends only after distributing it to the entire replicated database.
- **Asynchronous mode**: The updates to the replicated sites are distributed at certain intervals or upon request.

Synchronous mode is not suitable when the primary and second have a network latency greater than four millisecond (traceroute). Examples of vendor-specific database replication services are Oracle Data Guard, and MSSQL replication.

Replication servers secure the consistency of the application data only, Database binary files, log files, and other configuration data owned by the database server are not replicated with a give replication tools.

Standby Databases

Oracle® Standby Server

Oracle Standby Server (called DataGuard in Oracle 10/11) is a hot standby solution for the database server. This solution is software-based and, as such, is considered to be a less costly option than cluster solutions. In addition, there is no geographical limitation regarding the location of the standby database, so it can be used as part of a Disaster Recovery plan. On the other hand, to support High Availability there is a need to enable synchronous mode of database replication. Additionally, there are guidelines for maximum network latency for

replicating server working in a synchronous mode. In a DR scenario where, asynchronous replication mode can be enabled, the network latency considerations can be relaxed.

The configuration consists of two database servers; one is the primary database server and the other is in a standby mode. The servers can either run on different machines linked via a regular communication line, or they can run on the same machine. The primary server distributes its log files to the standby machine that can apply the changes immediately or not, based on the configuration. The detection of a failure and the failover procedure is a series of steps that can be operated manually or contained in a script that can be run automatically by the system that monitors the database. After the failover, which causes some inevitable downtime, the standby database will function as the main database server. Returning to the original database server is not a one-step process.

Installation of Control-M with Oracle Standby Server

No special guidelines are required for installation with Oracle Standby Server. However, it should be noted that in case of failover and a switch to the standby database, Control-M should be stopped and reactivated again, in order to reconnect to the database.

In a High Availability scenario, the Control-M application can either:

- Continue processing simply by re-pointing the **tnsnames.ora** file to the standby database server, or
- Manually start a standby Control-M environment that includes a preinstalled Control-M application that points to the standby database server.

For specific details on how to configure Control-M to point to Oracle Standby Server please refer to 'restore_host_config' utility documented in the *Control-M Utilities guide.*

In asynchronous mode, database replication is recommended for Control-M/Server in a Disaster Recovery scenario only. Control-M/Enterprise Manager in asynchronous mode is suitable for both HA and DR scenarios. In a DR scenario, a full version of Control-M is required to be installed on the secondary site. The reason why asynchronous replication is suitable for Control-M/Enterprise Manager in an HA scenario is because the active environment can be forced to resynchronize when the Control-M/Enterprise Manager comes up for the first time in a standby configuration and reconnects to the Control-M/Server. *To force synchronization for each data center, see troubleshooting menu of* ***root_menu*** *or see the Control-M/Server utility* ***reset_ecs****.*

Copying Control-M Oracle Database

When using the Oracle-provided utilities to copy an Oracle database from primary environment to secondary database server, it is important that the user definitions are not modified in the target database server. User roles should remain the same and one should not move user individual privileges to a group role. Stored procedures behave differently when a privilege is set at individual level or at the role level.

Installation of Control-M in Oracle Replication Database Environment

Database replication cannot be defined for the Control-M database since this requires certain database schema attributes that are not currently part of the product database schema.

PostgreSQL Replication

If not utilizing Control-M HA solution PostgreSQL server can also provide asynchronous database replication using hot backup with archiving enabled. The archiving creates an archive log at configured intervals or based on size limitation that is partitioned to a new achieved file. This replication process requires an automated script in place to ship the archived files over to the standby Control-M environment and either restore the shipped files immediately in the standby database server or leave them to be restored in case of disaster. This PostgreSQL solution is suitable only for disaster recovery scenario because replication can be done only in asynchronous mode.

One can utilize both Control-M HA and hotbackup at the same time to maintain both HA and DR environments. PostgreSQL manages the two features independently. Restoring a hotbackup on a machine with different hostname requires one first to localize the data before its restored.

When replication is done in a DR environment where DR Control-M is installed in a standby mode and the database is in read only mode, one needs to have a full plan when applying Fixpacks or patches in the DR environment. Applying Fixpack and patches requires that the database is both read and write mode. In DR environment the database is only in read mode. The two options available to apply a Fixpack or patches are stopping the replication and placing the database in read and write mode. Or to create a temporary database that can be pointed to using restore_host_config script and applying the fix against the temporary database. When DR environment is brought online then restore_host_config must be executed again to point the connection properties of database back to the read only database that contains the actual content of the replicated database.

Cluster Database Solutions

Oracle, and MSSQL offer solutions that use the benefits of the cluster environment to enable activation of multiple SQL servers over the cluster nodes. These solutions (Oracle RAC Parallel Server and MSSQL AlwaysON) provide transparency during the failover process by taking over the database session of the failed node in a way that is totally transparent to the application. The communication between the database servers on each node is very fast due to the FDDI communication between the cluster nodes, each node having its own database memory structures, and the database files residing on the shared disk of the cluster on several raw devices. A cluster database solution is preferred over disk replication because in addition to storage replication it also provides database server redundancy.

With the second release of Control-M for Cloud support, Control-M now support Postgres and Oracle RDS in a AWS cloud. RDS solution can provide similar level of database continuity as a clustered solution. Refer to AWS website for different levels of redundancy that are offered in RDS solutions.

Guidelines for Control-M installation with Parallel Server Cluster Solution

Starting with version 7 Fix Pack 3, Control-M fully supports installation with Oracle RAC server. There are no differences when installing Control-M with standard database server or with RAC server. The installation is transparent to Control-M — as long as one uses Service Instance Name and not SID Name.

Control-M Highly Available (HA) Configuration

Control-M High Availability enables you to maximize your production environment uptime and prevent data loss in the event of hardware maintenance or failure. Control-M supports the following high availability solutions:

- Control-M/EM and Control-M/Server high availability with Oracle or MSSQL: Enables you to set up a secondary host with Control-M/EM or Control-M/Server. If there is a hardware failure or if all Control-M/EM or Control-M/Server processes are down unexpectedly, the secondary automatically (by default) or manually assumes control and resumes production.
- Control-M/Server high availability with PostgreSQL: Enables you to set up a secondary Control-M/Server and a secondary PostgreSQL database server for database replication. If the primary Control-M/Server and database server are down, you can manually fail over to the secondary host.

Control-M High Availability is a software solution that monitors if the entire application is up or down. It is not a hardware level solution that monitors all aspects of hardware and conditional levels of the application health.

To set up your high availability environment please see the Admin and Install guide found on https://docs.bmc.com

Control-M/Server continues to support database mirroring on MSSQL and Oracle databases, but not in a Control-M High Availability environment. For more information, see Control-M/Server FAILOVER and Database Mirroring below.

Control-M Enterprise Manager High Availability Architecture using an Oracle or MSSQL database

The following diagrams show the configurations of both the Control-M Enterprise Manager (EM) and Control-M Server for high availability using Oracle or MSSQL databases.

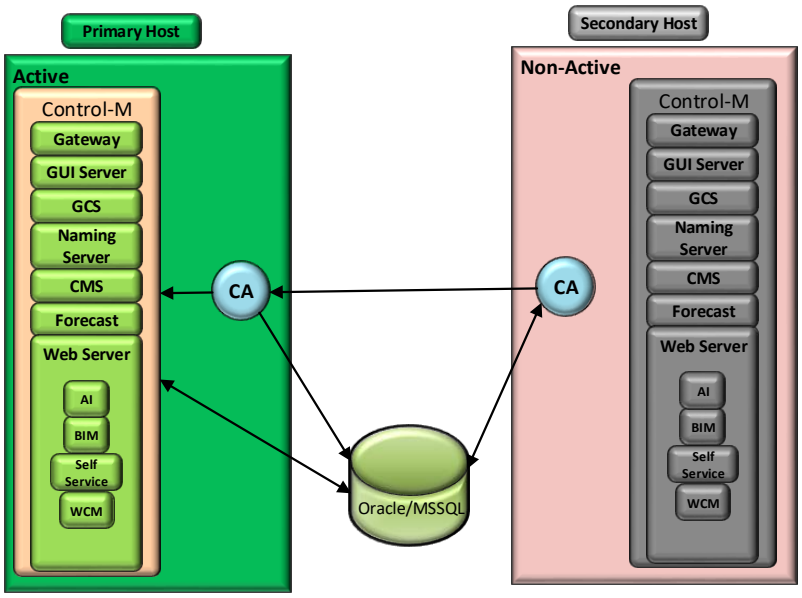

Figure 4: Control-M/EM with Oracle/MSSQL

The following diagram shows the configuration of the Control-M/EM after automatic failover when the primary components are no longer available.

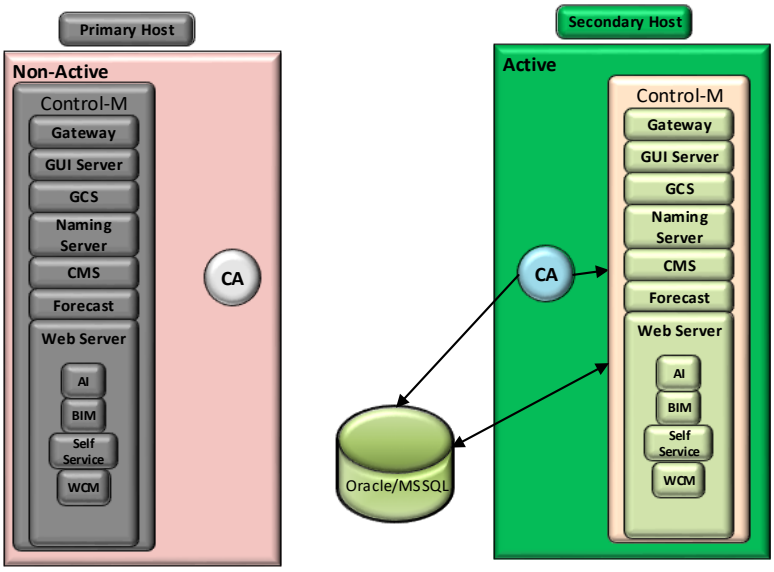

Figure 5: Control-M/EM High Availability Configuration

Control-M Server Manager High Availability Architecture

The following diagram shows the high-availability configuration of the Control-M Server using an Oracle or MSSQL database.

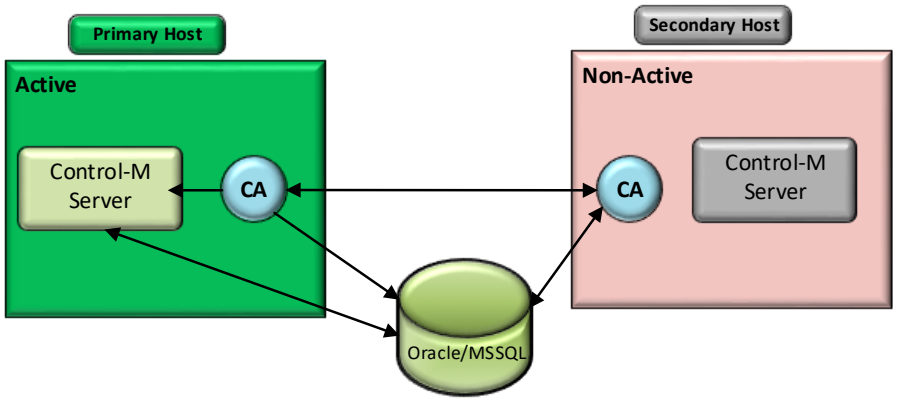

Figure 6: Control-M/Server High Availability Configuration

The following diagram shows the configuration of the Control-M Server using an Oracle or MSSQL database after automatic failover when the primary components are no longer available.

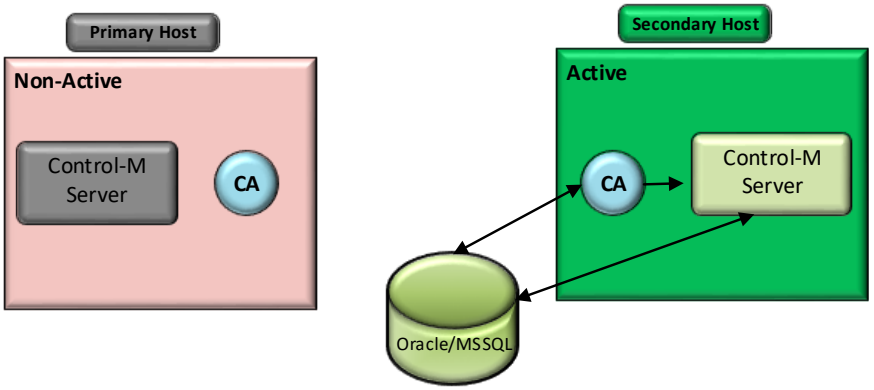

Figure 7: Control-M Server Automatic Failover

Control-M high availability on one host

To ensure that Control-M continues to run when it is installed on one host, the Control-M/EM and Control-M/Server Configuration Agent monitor and manage the following components in the CCM:

- Control-M/Server
  - Control-M/Server
  - PostgreSQL database server

Control-M/Enterprise Manager

- GUI Server (GSR)
- Gateway (GTW)
- Global Condition Server (GCS)
- Batch Impact Manager Server (BIM)
- Forecast Server
- Self Service Server
- Web Server
- Configuration Manager Server (CMS)
- Naming Service
- PostgreSQL database server
- Workload Archiving Server

If a Control-M/EM or Control-M/Server component goes down, the Configuration Agent if configured for automatic restart will attempt to start it up (if desired state is set to Up), based on defined intervals, as described in Maintenance parameters and High Availability parameters.

This configuration of one install is supported only with Oracle and MSSQL databases.

Control-M/EM and Control-M/Server high availability with Oracle/MSSQL

After you have installed a secondary Control-M, the Control-M/EM or Control-M/Server Configuration Agent on the secondary host monitors the primary Control-M/EM or Control-M/Server based on defined intervals. If there is no response from the primary, you can fail over to the secondary in one of the following modes:

Automatic failover: The secondary Configuration Agent automatically takes control and resumes production, when it detects that the primary Control-M/EM or Control-M/Server and its primary Configuration Agent has stopped unexpectedly.

Manual failover: You can perform a manual failover at any time from the CCM if the manual failover option is enabled. After the failover is complete, the production runs on the secondary.

NOTE: If you attempt to manually start up components on the secondary when the primary is active, the components shut down automatically. This prevents both the primary and secondary from running components simultaneously.

Control-M/Server high availability with PostgreSQL

The Control-M/Server high availability solution with a PostgreSQL database supports both a manual failover of the Control-M server along with PostgreSQL database replication. Control-M/Server is installed on the same host as the PostgreSQL database on the primary and on the secondary.

After the data replication is turned on and initialized successfully, the Control-M/Server database data is replicated, synchronously to the secondary database server. However, if there are network communication problems, the replication mode switches to Asynchronous. The data is replicated as well to a shared drive, which is used if the primary or secondary are temporarily unavailable.

The secondary Configuration Agent monitors the primary to verify life check responses from Control-M/Server and the primary Configuration Agent is working, based on defined intervals. You can perform a manual failover at any time from the CCM if the manual failover option is enabled, based on the conditions described in Manual failover.

If you are using a PostgreSQL database, the Configuration Agent manages the database component and sends a life check every defined interval (see Maintenance parameters and High Availability parameters). If there is no response after a defined number of attempts, the Configuration Agent restarts the database automatically. The restarting of PG is performed in the same environment. If you are using an Oracle or MSSQL database, you can view the database component in the CCM, but the Configuration Agent does not manage the component and cannot start it up or shut it down.

The following diagrams show the Control-M Server high availability architecture using the PostgreSQL database.

Control-M/Server High Availability Architecture with PostgreSQL

The following diagram shows Control-M/Server in a high availability environment using a PostgreSQL database.

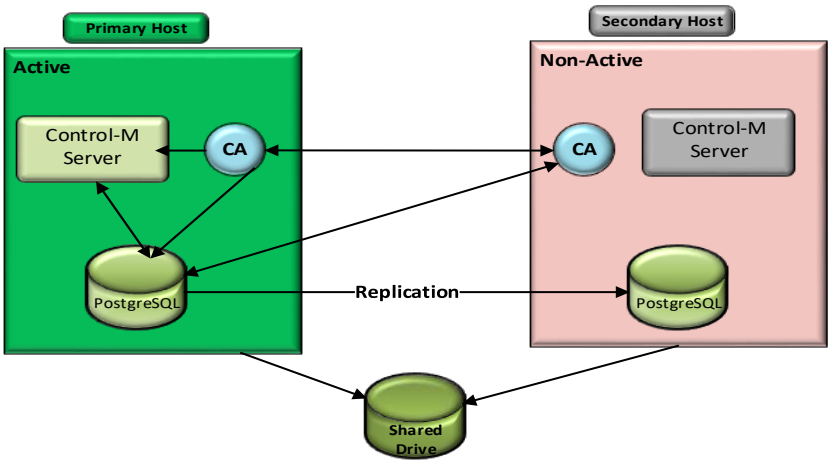

Figure 8: Control-M/Server HA with PostgreSQL

The following diagram shows Control-M/Server manual failover when the primary components are no longer available.

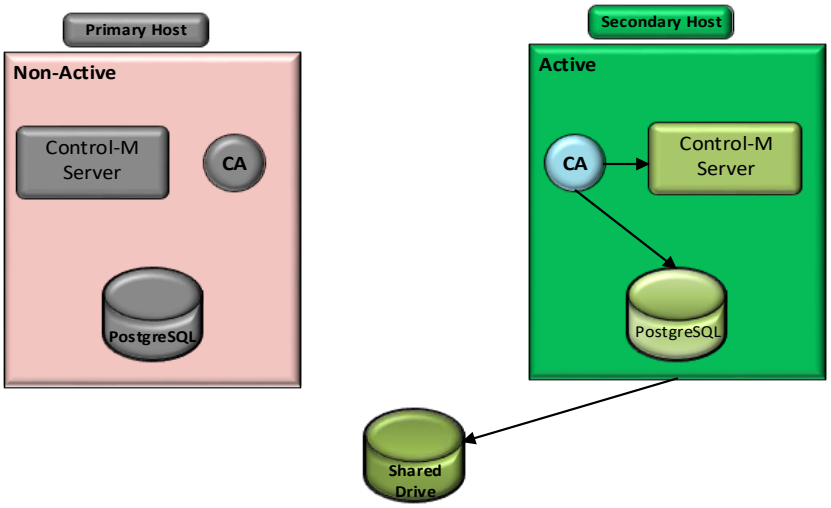

Figure 9: Control-M/Server with PostgreSQL Manual Failover

Control-M/Enterprise Manager High Availability with PostgreSQL is not available with the v9 GA release, This is being considered for future release.

Control-M/Server FAILOVER and Database Mirroring

Control-M/Server provides a built-in failover capability that includes database failure, network failure, and central Control-M/Server failure. In the event of a failure of the Control-M/Server, a backup or standby Control-M/Server can take over to maintain processing. The failover can be performed manually or automatically by a monitoring application, such as PATROL Knowledge Module for Control-M. As described previously mirroring is available with MSSQL and Oracle only. With PG the only option is the Control-M HA solution with database replication.

Any job that was submitted to an agent platform or to an agentless remote host continues executing. The Control-M/Server backup polls agent platforms or agentless remote host to determine the status of jobs listed in the Active Jobs file. The backup Control-M/Server performs all the duties of the primary Control-M/Server.

To ensure that every agent installed is authorized to receive requests from both the primary Control-M/Server and the secondary Control-M/Server, please refer to Control-M/Agent parameters in the Control-M Administrators Guide.

Unlike most cluster products, the distance limitation for this solution is guided by the network latency. Mirroring provides only synchronous replication and therefore is an HA solution. This solution is classified as Active/Passive replication. The backup Control-M/Server can reside on a machine that is located in a remote location with network latency not to exceed four milliseconds. Of course, this solution does not contradict any other cluster solution, and customers are encouraged to implement additional solutions in order to increase the availability of their systems.

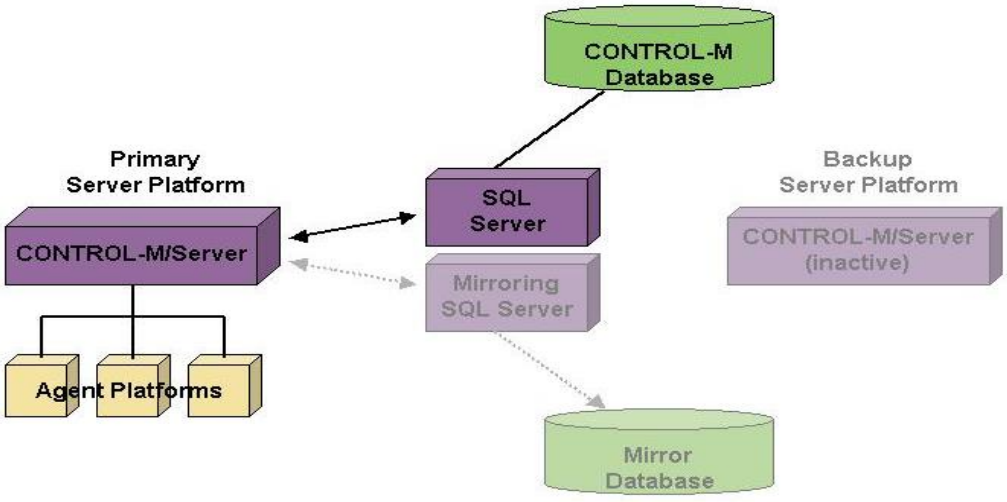

Figure 3: Control-M/Server in Normal Operation Mode

The following general guidelines should be followed:

- Install two full Control-M/Server installations, each with its own database server. The first installation (primary) is up and running and is used for the regular scheduling activities on the server. The second installation (backup) of Control-M/Server is inactive and is used as a warm standby for the second production server with hot standby database server.
- Ensure that the only component that is active on the backup installation is the database server that enables the replication of the database of another production Control-M/Server.

In case of a failure on the production server, the following occurs:

- The backup Control-M/Server is started. Since it has a mirror copy of the production Control-M/Server database, it contains the latest status of the jobs that where running. Jobs that were running on the agent platforms or agentless remote host at the time of failure continue to run. The failover can be performed manually or automatically by a monitoring application, such as PATROL-Knowledge Module for Control-M. *For further information, please refer to the Proactive Notification Tools section in this document.*
- The backup Control-M/Server reconnects to all of the agent platforms, and production resumes from the point at which it stopped.

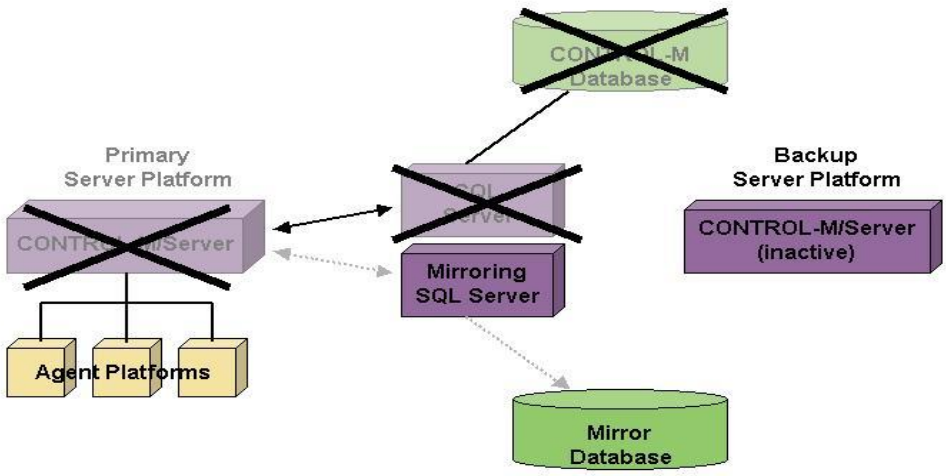

Figure 4: Backup Control-M/Server Taking Control Over Production

The recovery back-to-normal-operation mode is performed manually by rebuilding the primary server's database, copying the latest data from the mirrored environment and restarting Control-M/Server on it. *For details, please refer to Mirroring and Failover in the Control-M Administration Guide.*

Control-M Mirroring can also be used as a pure synchronous software database replication solution without the failover environment. Utilizing the mirroring feature allows Control-M/Server to connect to a standalone database server in worm mode and facilitate database replication synchronously.

When initializing mirroring to a standalone standby database server, one needs to select a "Full Build" over "Copy Mirror" data in order that the initialization will build a full schema for database replication. In a failover situation, when the primary database fails, then Control-M admin needs to select "Use Mirror" from the database mirroring menu, and all the connection properties will be updated to point the primary Control-M environment to the secondary database server. At some point, when the primary database server is restored, you will be able to go back to using the primary database by select "Copy from Mirror" option from the database mirror menu.

Control-M replication using file copy

Control-M currently **does not support** any form of file copy from a running and configured environment to a DR site or a standby secondary site. The only way to replicate an existing environment is to perform a full install of the secondary environment. The full install guarantees that all local configurations that are specified to the local environment are set properly. Control-M at installation time sets internal variables that point to the full path of Control-M environment, hostname, user account, and other local variables that cannot simply be reconfigured when an existing account is copied over to a new location.

If the database resided on the local machine, file copy of open files on a file database server is not supported by any database vendor. Each specific database vendor's documentation should be consulted for an acceptable means of how to copy the database files. Control-M includes database backup and restore utilities to facilitate an acceptable way of coping database data files. *Please see the Control-M Utilities Guide for more details.*

Proactive Notification Tools

This section reviews the tools that can be installed and integrated with different Control-M components in order to give proactive notifications about Control-M events. These notifications enable the user to prevent, or at the very least identify failures of the product components so that preventative or recovery actions (such as the activation of the failover mechanism) take place as soon as possible.

PATROL Knowledge Module for Control-M

The integration of Control-M with PATROL can increase the high availability of Control-M in several ways. This integration allows users to control and monitor the availability of Control-M/Server for Distributed Systems and Control-M/Enterprise Manager components using the PATROL console. It also ensures that potential failures are proactively brought to the attention of the appropriate personnel.

The PATROL console provides a view of the current status of all Control-M components and enables the users to control and modify their status as required. Warnings and alarms are issued when predefined thresholds are crossed. This allows fast detection and escalation of problems as they occur in any one of the components.

PATROL Knowledge Module (KM) for Control-M automates the failover procedure of Control-M (discussed in the previous section). This configuration provides the customer with a solid High Availability solution without investing in cluster hardware. *Additional information can be found in* Knowledge Article 000106519

By detecting problems, analyzing trends, and managing multiple hosts simultaneously, PATROL KM for Control-M helps to ensure that the Control-M installation continuously runs efficiently.

Frameworks Integration

The integration of Control-M with Frameworks allows better management and higher availability of Control-M by monitoring its different components.

This integration allows Frameworks to monitor different components of Control-M. Problems are detected immediately as they occur and then escalated to the framework application, where corrective action can be taken to keep continuous production flow. Control-M provides ability to send SNMP traps to any network management framework such as HP OpenView or IBM Tivoli software. Control-M/Enterprise Manager can send application Alerts and administrative XAlerts via SNMP.

Additional information can be found in Knowledge Article 000030024

Best Practices for full Control-M Databases backups

If you are using Control-M HA solution there is no need for hot backup therefore this section does not apply.

To protect Control-M against database failures, backup of the data base should be performed in such a way that one can recover to the point of failure. Control-M provides two ways to perform backups of the application database. The traditional way (or the **cold backup**) can be performed when the application is down. Cold backup is performed when no updates are being made to the database, thus protecting the integrity of the data.

Control-M provides a second mechanism for backing up a live database without having to shutdown the Control-M application. This is referred to as a **hot backup**. The integrity of the data is maintained in a hot backup because together with the running database snapshot, a hot backup includes a list of running transactions (more commonly referred to as the transaction log) that bridges the gap between the base snapshot and the last committed transaction. The database snapshot together with the transaction log combine to recreate the database with full integrity.

Cold backup can be inconvenient to a mission-critical environment that needs to be up 24x7. Additionally, cold backup will restore the data to the same point when the backup was performed. For Control-M/Server database this is not ideal point of recovery. Control-M/Server database should be restored to the point of failure to ensure that job flow processing continues from point of failure and not from few hours back.

Hot backup technology is dependent on the database feature called **archiving mode**. When archiving is enabled, the database incremental changes are written to the transaction log file, and then, after a configured amount of time or a configured size of transaction log, the log file is closed and made available for shipping to a safe designated storage. The transaction log files are sequentially labeled from the time the archiving is enabled. Every time a new hot backup is performed the archived logs are marked and can easily be identified as belonging a new backup snapshot image. The archived logs file should be stored on a safe storage location that is protected from machine and disk crashes.

When enabling archiving mode with PostgreSQL database, the user will be prompted for an empty folder where the archived log files will be written into when the transaction log files are closed. The active transaction logs file (also known as **Write Ahead Log,** or **WAL,** files) are kept in the default location **<pghome>/data/pg_xlog/.** They serve as the source files that feed the archive log files. The WAL file is managed by PostgreSQL and remains open at constant size for a specified duration. Only after the WAL file is closed does it get moved automatically to the archive log directory. Assigning a network drive for the archived log directory will ensure that files are safe from damage in case the host machine crashes.

Because the open WAL contains up-to-the-minute transactions, in case of a disk crash, it is in jeopardy of becoming damaged or corrupted. The transaction log files are simple flat files, and therefore, there is a risk that some of the data that was already committed is still buffered in the OS memory and not written out to the file. To manage the OS file buffering, a file synchronization mechanism exists with most database vendors. This solution ensures that for every database commit, the memory buffers are flushed and the data is written to the disk.

The PostgreSQL parameter **fsync**, when turned on, ensures that for every database commit, the memory buffers are flushed and the data is written safely to the disk. There is some performance penalty when the **fsync** parameter is turned to on. On databases with light activity, it is also recommended to add to the file

postgresql.conf the parameter **archive_timeout** with the value of 300. This parameter will force an archiving activity every five minutes. Even if the WAL files are not fully used, it will force a WAL file to close and be shipped. This also ensures that, in case of a disaster, the worst possible loss will be the past activities that occurred only in the last five minutes.

As the transaction log files fill up and are moved to the archive log directory, the number of archived log files can grow dramatically when there is a lot of activity. Therefore, in order to make the recovery process manageable, we recommend that a hot backup be made every 24 to 48 hours. The hot backup creates a new baseline, and thus makes all the previous archived log files irrelevant so they can be deleted. As soon as a hot backup is made, the baseline for a full restore is created and this baseline backup should be moved over to a safe location. Following the hot backup all the archived log files that will be create are the incremental changes and will be now associated with the new hot backup baseline.

For PostgreSQL database, when you initiate a new hot backup, you will be prompted for an empty directory to store the backup files. The backup files will include both a baseline database file and a full set of database binary files. The database data file is an export of the entire database instance; therefore, if both Enterprise Manager and Control-M/Server are serviced by the same instance then both tablespaces will be included in this export file. *Additional information can be found in* Knowledge Article 000094701

The following is a summary of backup strategy that is discussed above

1. **Enable archiving mode**
   - This option is necessary in order to run hot backups.
2. **Perform a hot backup**
   - Create a baseline of the database by performing a hot backup. A hot backup can be executed any time without bringing down the Control-M application.
   - Create a new baseline every 24 to 48 hours depending on the amount of activity and the size of the archive log files.
   - Purge logs archived prior to the hot backup after the new baseline is created.
   - Ship the baseline backup file to a safe storage location immediately.
3. **Ship archive log files**
   - As soon as transaction log file is closed, it should be shipped to a safe archived log location.

Both hot and cold backup operations perform full database backup using database vendor utilities. They also create a backup that is application independent. In order to restore this backup, Control-M needs to be down, and the entire system is restored to the point of the last archived log file.

When running hot restore in a primary or failover environment, ensure that archiving mode is off and that Control-M application is down. With PostgreSQL, you are also required to ensure that the PostgreSQL database is down prior to executing hot restore. The hot restore utility will prompt you for both the archive location of the archived log files and for the location of the hot backup baseline. The hot restore brings the database to last consistent position when the transaction log file was closed. The PostgreSQL restore also restores all the binary and database files to point of the last backup. *For additional limitations on restoring a hot back file with PostgreSQL please see Knowledge Database article* 000095042.

For PostgreSQL database, the restore operation will import both tablespaces automatically when one instance is used to service both Control-M/Server and Enterprise Manager components.

The full database backup for Enterprise Manager can also serve for safekeeping previous versions of application data. Performing a backup every 24 or 48 hours allows the user to restore the entire database to a previous date and time. In Control-M Enterprise Manager, this can be useful when a user mistakenly modifies application data, such as job definitions, and needs to restore to some point in the past. With a PostgreSQL database in situations where both Control-M/Server and Enterprise Manager reside in the same instance, the restore is not very practical. In order to restore application data from a full backup, one needs to build or have ready a test or a failover standby environment where the entire database can be restored and then selectively extract the application data needed to revert the changes in the Control-M data. *See XML utilities (e.g. exportdeftable) for extracting an Enterprise Manager specific set of application data from a given environment.*

Starting with Control-M/Enterprise Manager version 6.4.01, job versioning is supported in the product, which allows users to restore previous versions of job definitions using standard application functionality rather than going back to an older date backup.

In order to maintain a backup for the purpose of having the ability to restore application data from some past date, it is a best practice to use Enterprise Manager utilities to export the application data such as job/calendar definition, security, etc and then have it available for restore in the primary environment. These utilities can be executed periodically without having to bring the Control-M application down. *See Control-M Utility Guide, refer to the 'util' command and XML utilities.*

Best practice for performing backups with PG in Control-M HA mode:

When running backup utilities or for that matter any administrative utilities in a HA environment one needs to execute the utilities in the active environment or primary server and not on the secondary server that is not active. In a HA configuration the server that is not active does not have access to active process of the primary configuration and therefore is not able to run connect to the services it needs to complete the utilities.

Additional information can be found in knowledge base: 000094701

Best Practices for Disaster Recovery

In a Disaster Recovery (DR) scenario, the assumption is that the primary site, where Control-M components are installed, and a database server component that services Control-M components, will cease to exist and cannot be recovered. A failover Control-M environment with similar configuration is needed to take over all the processing from the point of failure. Some loss of data may be tolerated when manual recovery of data takes a short time. DR configuration has a much larger tolerance in most of the business continuity requirements than HA configuration.

The Control-M architecture is a scalable, enterprise-wide solution with distributed components residing all across a connective network. The distributed Control-M components can serve as redundant entities to support failover or they can serve for load balancing in capacity planning. Only Control-M/Enterprise Manager supports distributed architecture for majority of its add-ons.

When analyzing DR needs, each Control-M component should be considered individually — whether it provides redundancy functionality or mainstream functionality. Even if you chose to use the **One Install** (where all components are installed under the same user account), for DR purposes, the Enterprise Manager and Control-M/Server must be evaluated independently and treated as separate components.

The database server, which is an integral part of Control-M components, is a separate critical application. It is either internal to Control-M and thus integrated from the time of installation (as in case of PostgreSQL), or it is external to Control-M and thus installed in a different account or location to service the needs of the Control-M application (as in case of existing Oracle and MS SQL configurations). The management of external database applications is outside the scope of this document; we will focus only on the data that is application specific and on the connection properties that allow Control-M to utilize the services of an external database server.

The discussion of database availability was covered above in the *Database Solution* section. No matter what option you select to achieve database recovery, the procedure you select must follow guidelines of the specific database vendor; simply copying files of the database and restoring in a new location breaks all data consistency and integrity aspects of a database server and is not supported. Control-M simplifies the basic backup/restore of the database operation by providing utilities that wrap the vendor provided functionality to invoke backup/restore procedure for both internal installation and somewhat limited versions for external installation of database servers.

The primary function of Control-M/Server component is to manage enterprise-wide workload for execution of scheduled and event-based tasks that most of the time are time sensitive and mission critical. The sequences of events that occur in the Control-M/Server are transactional, and therefore, recovery of the database must be at

the point of failover to ensure that no job is executed twice after the recovery. Additionally, the integrity of the data is critical for the application to resume functioning from the time of recovery. As a result, our recommendation is to implement a standby database solution, such as Oracle DataGuard, Replication Servers, Control-M/Server mirroring, hot backups with immediate log shipping, and Control-M/Server HA solution to achieve recoverability with full data integrity that span distant network sites. *See Database Backup Guidelines and Database Solutions sections above on how to implement various database availability solutions.*

The rest of this section will focus on best practices for hot backup and restore as the sole solution for Disaster Recovery that does not involve any additional utilities or third-party components that are not packaged in the product. When using PostgreSQL, BMC recommend Control-M hot backup procedures provide asynchronous replication, which is encapsulated in the product for Oracle, PostgreSQL in all releases, and Microsoft S SQL Server databases.

In order to accomplish a failover environment, the following must be achieved

1. Select a failover platform that will facilitate the same database vendor's ability to connect and restore from the hot backup performed on the primary environment
2. Recover Control-M application on a failover environment
3. Recover database connection properties for external database servers
4. Recover and restore the database hot backup that was created on the primary site
5. Reconfigure local data that resides in the database

Selecting a failover platform is fully dependent on your ability to restore a hot backup created on the primary platform. The hot backup functionality is provided by the database vendor. Control-M does not implement the backup; rather it wraps the database vendor's own functionality for performing hot backup and restore. Therefore, the failover platform has to be a supported platform where Control-M and the database client are on the list of supported platforms. *For Product Compatibility and Availability, please see the Control-M Customer Support website or Release Notes for your specific product and version.*

As an example, where the primary environment is Solaris with an Oracle database, one can build a failover environment on AIX® platform because the AIX platform is compatible for Control-M with an Oracle database. In all cases, the database vendor must be same for primary and failover environments. In an environment where database is PostgreSQL, the primary and failover platforms must be the same because the backup also includes the PostgreSQL binary environment. Therefore, in order to restore a PostgreSQL, hot backup environment the failover must be the same platform as the primary.

Recovering a Control-M component on a failover environment can be accomplished only through a new Control-M install performed from scratch in the failover environment. As specified above, file-by-file copy or a basic file backup and restore of Control-M directory tree is not supported. A full install of Control-M, including the database, should be performed in a failover environment independent of any aspect of the primary environment. If the database is external, then it's up to the database services (DBA) to either provide a separate database within close proximity to the failover installation or to provide connection properties to a database server that will service the failover environment and will guarantee that the network latency will not impact the throughput of Control-M.

When installing the failover Control-M, ensure that the following:

- OS account owner for Control-M user is the same on both environments
- Path to Control-M account is the same from root directory on both environments

These parameters will ensure that jobs that need to run as Control-M owner will not require any modification on the failover environment. For PostgreSQL database configuration, the owner, owner ID, group, group ID DBO, DBA, DBO passwords, and path must be identical.

In a scenario where the continuity of primary database services is guaranteed — and when Control-M on the failover environment will be connecting back to the primary database services — then the connection properties in a failover environment will need to be configured to point to the database services that guarantee the database continuity. In this scenario, the following is needed to configure connection properties:

- The configuration includes changing the connection properties configuration files found in interface file **tnsnames.ora**, **interface**, and **postresq.conf** files.
- Modifying DBO user and password, database name, and instance name

Please refer to documentation for utility '*restore_host_config -reconf_db'* to facilitate these changes for both Enterprise Manager and Control-M/Server

When the failover Control-M application has access to the database server, the data restore operation can proceed. Prior to running hot restore, make sure that all the archived logs and the base backup file from the primary database location are safely copied to the failover machine or are accessible to the failover Control-M account prior to the restore. You will be prompted by the restore database command to specify the base backup file that was brought from the primary site as the source for database restore. You also will be prompted to specify the archived log directory that contains all the archived log files. The hot restore procedure will restore the base copy of the backup and then incorporate all the archived log files into the final configuration of the database. In an actual DR scenario where the last archived log file may not have been closed and shipped over to the failover site, there will be some discrepancy between the restored data and final state of some jobs in the primary Control-M environment. In this case, manual intervention may be required to reconcile the data of the

between the two sites. For a DR scenario that is controlled, one has the ability to ensure that archived logs are closed and shipped prior to bringing down the primary site. In such a controlled environment, the restore operation will ensure that the data is restored to the point of failover.

Recovery to the point of failure is critical for Control-M/Server because a loss of a single transaction can make a difference between knowing if the last job was submitted to the agent for execution or if it is still waiting to be sent. Therefore, recovery to point of failure on Control-M/Server is essential. However, recovery to point of failure on Enterprise Manager is much less critical because all the active data on the Enterprise Manager will automatically synchronize from its corresponding Control-M/Server or Control-M for z/OS®. The data that may be of possible concern is the definition data. As a result, the more convenient option for data recovery for Enterprise Manager may be to utilize the **util** export and import utility to extract all the data from primary Enterprise Manager on a daily basis and then import it to the secondary environment in case of disaster. As mentioned previously, the **util** export can be executed when Enterprise Manager is up and running and it does not require the overhead of handling log shipping and maintenance.

After the database is restored in a failover environment — and prior to starting the Control-M application — you need to ensure that database contains local environment information and is adjusted to reflect the current failover environment. More specifically:

- The local hostname that may be hardcoded in Enterprise Manager in the System Parameters and in other configuration files is updated to reflect the new environment. *See Utilities guide on "restore_host_config" for the steps needed to make the data imported reflect the new local environment.*
- Control-M/Server maintains full pathname to executables in the **CMS_SYSPRM** and **CMS_CMNPRM** table. **CMS_SYSPRM** and **CMS_CMNPRM** table needs to be updated to reflect the new paths for **LOGDIR**, **PROCLIB**, **EXE_PATH** fields. This step can be skipped if both primary and failover site have the same path from the root directory to the Control-M account. *See Utilities guide for restore_host_config utility.*

Step by step procedure for DR test and recover is provided in a Knowledge Base articles:

000156434 - Documents for Control-M/Server with Oracle / MSSQL

000157687 – Documents for Control-M/Server with Postgres

For More Information

For more information about application continuity, please visit www.bmc.com/control-m.

Business runs on IT. IT runs on BMC Software.
Business runs better when IT runs at its best. That's why more than 20,000 IT organizations – from the Global 100 to the smallest businesses – in over 120 countries rely on BMC Software (NASDAQ: BMC) to manage their business services and applications across distributed, mainframe, virtual and cloud environments. With the leading Business Service Management platform, Cloud Management, and the industry's broadest choice of IT management solutions, BMC helps customers cut costs, reduce risk and achieve business objectives. For the four fiscal quarters ended June 30, 2012, BMC revenue was approximately $2.2 billion.

Where to Get the Latest Product Information

To view the latest BMC Software documents, visit the BMC Customer Support page at http://www.bmc.com/support_home . BMC Software distributes printed copies of flashes, technical bulletins, and release notes with most product shipments, as indicated on your shipping list. In addition, all notices are available on the Customer Support page, including any notices that BMC Software issues after you receive your product shipment. You will not receive new notices by mail. However, by subscribing to proactive notification, you can receive e-mail messages that direct you to those notices. For more information about proactive notification, refer to the Customer Support page.

Iyar285777

BMC delivers software solutions that help IT transform digital enterprises for the ultimate competitive business advantage. We have worked with thousands of leading companies to create and deliver powerful IT management services. From mainframe to cloud to mobile, we pair high-speed digital innovation with robust IT industrialization—allowing our customers to provide amazing user experiences with optimized IT performance, cost, compliance, and productivity. We believe that technology is the heart of every business, and that IT drives business to the digital age.

BMC – Bring IT to Life

