

# Best Practices for Implementing Workload Automation with BMC Control-M



APRIL 3

Authored by: James J. Gingras

---

Table of Contents	
Executive Summary .....	3
Introduction .....	4
Overview .....	5
Best Practices and Standards .....	6
Best Practice: Control-M Implementation Architecture .....	7
Best Practice: Cloud Implementations .....	9
Best Practices: Containers .....	10
Best Practice: Workload Services .....	11
Best Practices and Standards for Workload Automation Components .....	14
Best Practice: Grouping the Workload for IT service alignment .....	15
BEST PRACTICES: SITE STANDARDS .....	16
BEST PRACTICES: SITE CUSTOMIZATIONS .....	17
Best Practice: Using Folders .....	18
Smart Folders .....	18
Regular Folders .....	19
Cyclic SMART Folders .....	19
Best Practices: Alerting .....	20
Best Practice: Scheduling .....	20
Job Editing Form.....	22
Templates .....	30
Workload Policies .....	30
Conclusion .....	30

---

# Executive Summary

---

This document provides high-level descriptions of essential issues and concerns for implementing Workload Automation using BMC Control-M.

The best practices presented are derived from years of experience in implementation Workload Automation with BMC Control-M.

Each organization has different best practices and standards based on how the processes of managing IT are designed, automated and supported.

It should be considered as a good starting point and used to ensure the best implementation workload automation implementation possible.

# Introduction

The purpose of this document is to provide guidelines for introducing best practices and standards for Workload Automation using BMC Control-M into your organization.

This document maintains an Enterprise Workload Automation perspective in that it will be platform independent and technology agnostic.

The document focuses on understanding the lifecycle of Workload Automation and the processes required to manage an enterprise workload. This includes coordinating automation between business and IT siloes (batch, event or user driven), automated service and support process integration, and addressing DevOps issues and concerns.

Relevant and practical recommendations are presented based on experience from all types of real-world implementations whether they are “greenfield” implementations with small and growing workloads to extremely large global workloads.

The diagram below describes the typical IT environments included in an IT Enterprise.



**FIGURE 1: TYPICAL TYPES OF IT ENVIRONMENTS**

---

# Overview

---

In his famous book, *The 7 Habits of Highly Effective People*, Stephen Covey's Habit #2 is "Begin with the End in Mind". This phrase captures the best kind of thinking in terms of architecting a solution and is essential when considering enterprise level systems like Control-M. We need an understanding of what type of environment(s) we are installing the three-tiered Control-M architecture into so we can design it according to the best practices and access controls that already exist within an organization.

Any best practice or standard begins with an understanding of the stakeholders and a naming convention of some type. Using a more modern vernacular, it should also include some type of service model that describes the types of services each IT group provides in terms that represent business value. This begins with an understanding of IT service management within an organization in relation to the enterprise workload IT supports.

Best practices include an understanding of the requirements for managing an IT enterprise that consists of both new and legacy technologies, applications, and systems, as well as, an understanding of differences between IT applications and the IT infrastructure required to support them.

From a workload automation perspective, the main benefits for implementing an enterprise workload automation strategy are:

- Centralized control of automation across all of Information Technology landscape
- Consolidation of tools and capabilities for better resource management and accountability
- Coordination of automated workflows between IT Silos and tools
- Visibility to the business value automation provides at all levels of the organization

The capabilities of Enterprise Workload Automation tools like BMC Control-M enable and automate the creation and governance of best practices and standards. The advantages to their use are universal. These benefits include:

- ease of implementation
- improved auditability, security and reporting
- proactive, exception-based systems management
- improved supportability and troubleshooting
- actionable intelligence for consumers of Business and IT services that enables better/faster business decisions.

Standards imply reusable components employed in a repeatable pattern. This is the essential requirement for workload automation to use jobs to automate any or all the tasks/steps of a well-defined process.

---

# Best Practices and Standards

---

Best Practices imply organizational methods of implementing automation that comply with standard repeatable patterns already exist. As such, standards don't usually change. Best practices vary within each organization according to how they support the business and IT.

The over-arching best practice is to identify and implement all the standards possible within the Workload Automation Domain that will ease the administration and management of an enterprise workload. Implementation best practices are to use the tools already supplied in Control-M to achieve this.

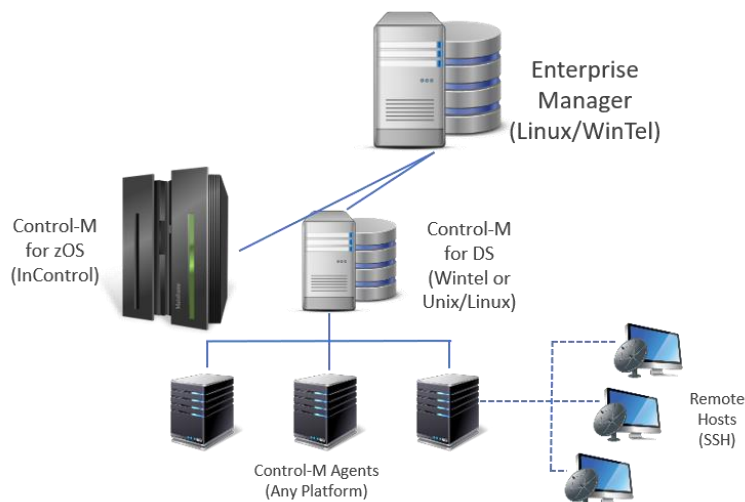
Standards include identification and where required detailed definition of service definitions, site standards, site customizations and GUI options.

The remainder of this document will discuss best practices for specific methods required to administer and enterprise workload.

## Best Practice: Control-M Implementation Architecture

This section includes some basic best practices for implementing BMC Control-M.

Control-M utilizes a standard three-tiered architecture.



**FIGURE 2: BASIC CONTROL-M ARCHITECTURE**

1. Enterprise Manager (EM) - mostly manages user interfaces and monitoring among a few other things being the top tier.
2. Control-M Server (CTM) - the workhorse that manages workflow and job execution and houses the definitions.
3. Control-M Agents – reside on servers where or remote hosts where the workflows will execute. There is an agentless capability. Application specific control-modules or plugins also reside on the agents.

There are two databases, one for the Enterprise Manager (EM) and one for the Control-M Server.

Referring to the typical IT environments diagram above, there will be multiple instances of each component depending on what works best in an organizations environment.

The reality and benefits of cloud computing provide even more flexibility when implementing the Control-M Architecture.

A key consideration is that one Enterprise Manager can manage multiple Control-M servers. One Control-M server can manage multiple agents. For mainframe implementations there are no agents to consider.

Control-M inevitably becomes a mission critical application. As such, in most organizations, there are a higher or more rigorous set of requirements for production environments for critical applications.

---

Control-M has practical limits with the number of jobs that can execute at one time and the number of agents to connect to any one Control-M server. Most environments, especially to start, are not bound by these limits simply because Control-M is very efficient and can manage thousands of jobs and hundreds of agents per server with sufficient allocated resources. In today's server environment with virtual machines and cloud resources allocating hardware and memory are performed with software which minimizes resource considerations from a Capital Expense (CAPEX) perspective.

In medium to large implementations where hardware and support cost are not the major consideration, multiple components on multiple servers (one for EM, one for CTM) is the most common configuration. This configuration also addresses the segregation requirements for compliance and security. SSL is also available to encrypt communications between Control-M components as required.

With small to medium implementations using one Enterprise Manager to manage Dev and QA/Test environments is not uncommon.

In most environments, databases are managed as a separate component regardless of application. Control-M fully supports this model. Control-M does have the option to provide a Postgres database installation for standalone environments. Postgres is an open source database that is very mature and well known. In small to medium implementations the Postgres database included with Control-M is sufficient. Control-M also support Microsoft SQL and Oracle databases.

In most implementations issues arise with the ports that must be opened and maintained for Control-M to communicate across the IT enterprise. Do to Control-M's enterprise class capabilities careful consideration and understanding of these requirements must be identified to ensure a smooth and successful implementation.

In most implementations the need for Self-Service is also required. Self-Service is a web-based interface for Control-M. By default, the Enterprise Manager controls and manages the Web Server that is used for Self Service and some other Control-M Components (Batch Impact Manager, Workload Change Manager). In some organizations Web Servers are managed separately with implementation specific requirements. Control-M supports these types of environments.

Automation API is a component of Control-M that is also managed by the Enterprise Manager. It provides and programmatic API and Command Line integration to Control-M through HTTPS. If this feature is required careful consideration must be given to understand the impact of allowing the enterprise workload to be affected programmatically without human intervention. However, in dynamic environments where an automated release pipeline is required, Control-M can be configured to provide these capabilities for the Enterprise Workload.



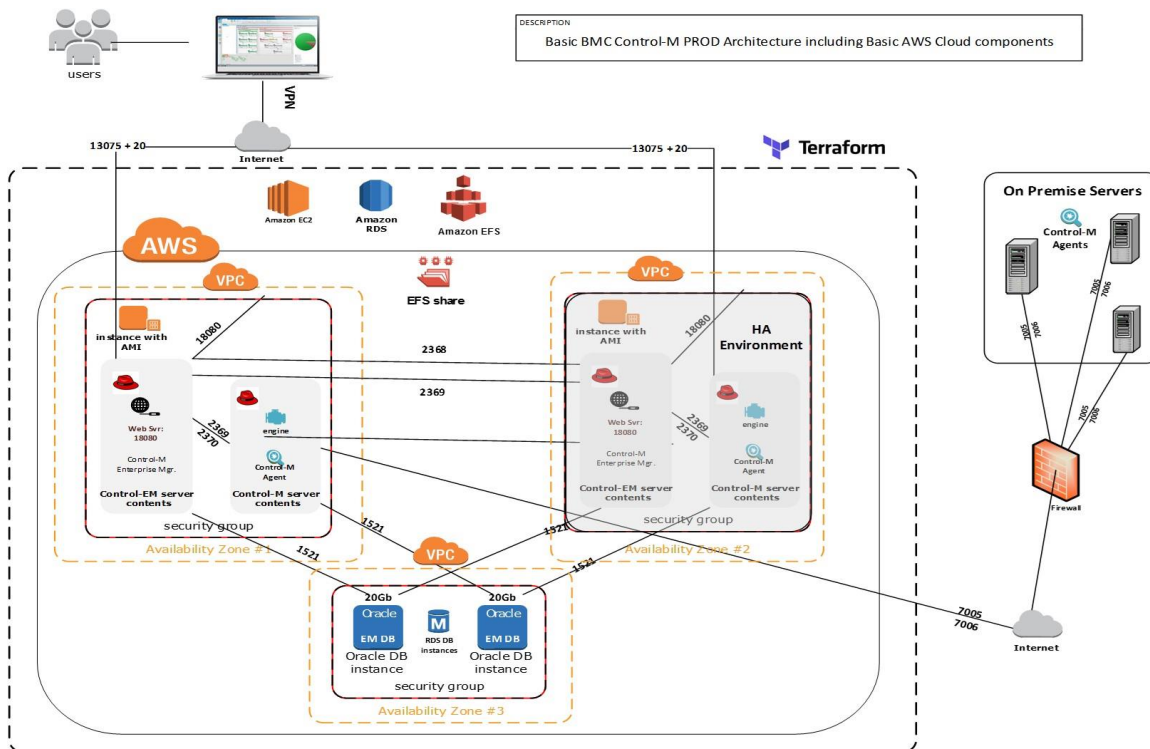
## Best Practice: Cloud Implementations

Modern system and application architectures consider the cloud as an extension of their existing infrastructure when time to market and/or excess resources are required. These hybrid architectures are the norm not the exception. As such, Workload Automation can leverage the same advantages, both as a hosted service where the entire Control-M infrastructure resides or some combination thereof.

Both Amazon and Microsoft fully support Control-M running on application servers. Amazon marketplace has a variety of BMC Control-M pre-configured solutions.

Cloud implementations provide numerous advantages for most organizations. It is especially helpful for small to medium organizations that may not have the resources or skill sets required to manage a mission critical enterprise class workload application like Control-M.

Main points of the architecture are where the agents will reside. The Control-M application infrastructure usually resides either on-site or in the cloud on an environment basis (i.e. considering DEV, QA/TEST, PROD, DR as an environment).



**FIGURE 3: CONTROL-M HOSTED IN THE AMAZON CLOUD**

The figure above is a real-world implementation of Control-M hosted in the Amazon cloud and created with Infrastructure as Software product, Terraform. In this implementation all the agents (except the default Control-M system agents) reside on premise.

---

All The considerations for cloud environments are beyond the scope of this document.

---

## **Best Practices: Containers**

---

Best practices for containers, like Dockers is no different than any other entity. Whether the container will reside on premise or in the cloud, Control-M can be used to automate the creation and management tasks around the release pipeline for containers as simply as it can for any other application or system. Because it is platform, technology and application agnostic the only concern is to identify which type of integration to use: API, Application Integrator, Messaging, or CLI and scripting.

One of the key strengths of Control-M in these scenarios is its ability to determine and manage the impact of maintain the containers on the rest of the workload and to standardize support and operational procedures and automate remediation as required. This is sometimes referred to as Operationalizing.

*Best practice for using Control-M for containers is to treat the container like any other application except, of course, that it may require additional upfront support from Subject Matter Experts (SMEs) to determine the best approach for an organization and to create the templates and reusable components that enable the automation and integrate with Control-M.*

## Best Practice: Workload Services

The concept of providing service has utility from the customer's viewpoint to a technical guru or SME deep in IT and all those in between. For this reason, Service Level Agreements (SLAs) that measure and report on customer satisfaction of the service provided are the best approach to managing both business and IT today. To accurately measure and report the true level and satisfaction and the cost of support we need to have a method to establish and maintain accountability and transparency (measurement and reporting). The names of the services provided to the customers need to be aligned with the types of automated functionality required to provide as IT services from application level all the way through the IT infrastructure. Best practices for accomplishing this begin with identifying or creating a service model.

In its simplest form a service model is a list of the types of services a provider will be offering to any individual or group. A more relevant example maps IT components to a set of capabilities and functionality the business services provide to customers that consume constituent IT services.

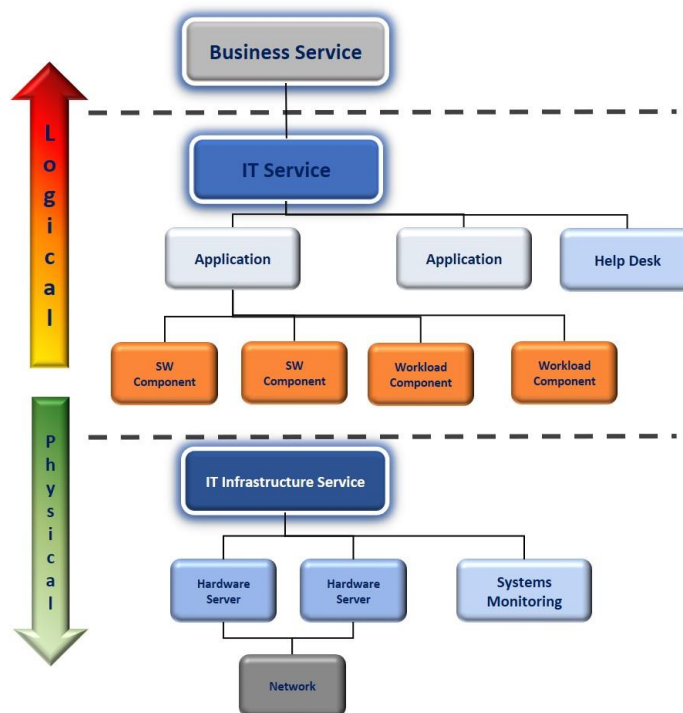
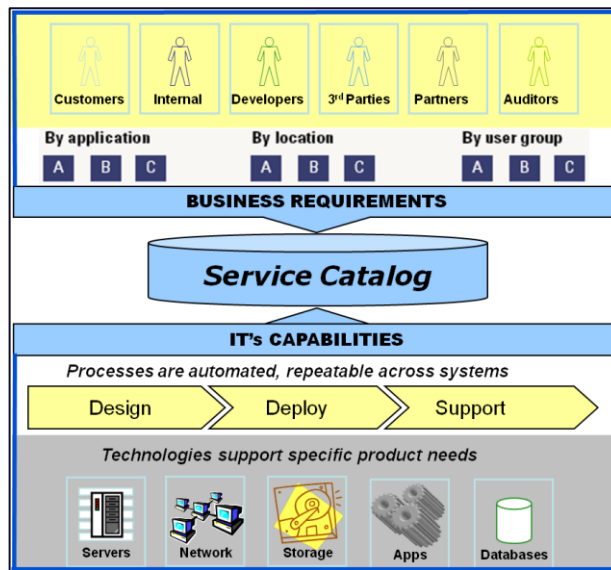


FIGURE 4: SERVICE MODEL

Service models begin with identifying a business service that is provided to customers. For example, a securities firm offering checking and savings accounts might have a business service called Account Management.

Internally within an organization, IT services are the agreed upon groupings of IT applications and associated support processes required to both provide and support the business service offered to the customer.

IT Infrastructure services are the groups of IT Infrastructure systems and support processes required to run the IT applications and associated support processes in support of the business operations and business services offered to the customer.



**FIGURE 5: IT SERVICE OFFERINGS**

The figure above shows a more mature representation of IT service management where service models are well-defined and managed using a service catalog. A service catalog is a more comprehensive list of the IT services provided that are mapped to well-defined support processes to meet service level requirements defined in service level agreements with the business and customers.

One of the main considerations for defining a service model is the requirement for automated integration with existing IT Service Management processes and associated Help Desk (Service Desk) tools. In a mature implementation, this integration is automated. A key component of any mature IT Service Management environment is the Configuration Management Database (CMDB). The CMDB consists of Configuration Items (Cis) that form the building blocks for all the IT services. The configuration items (Cis) define the components of IT that will be rigorously managed through service management processes, including change control and incident management at a minimum.

As the name implies, IT Service Management certainly starts with some type of service model. In the diagram above the service model is well-developed and defined. The point being, the names of supported business services and associated IT services are well-known and supported. This model enables end-to-end visibility, accountability, including support costs, for workflows within the enterprise workload. The implementation of the service model can be automated using a CMDB and service catalog. In mature implementations, the names of applications and sub-applications are also defined in the CMDB and can be transferred to as part of naming standards within Enterprise Workload Automation and Control-M.

---

In mature implementations of Enterprise Workload Automation that includes mature IT Service Management, management, the service/support processes such as, managing availability, trouble tickets, change orders and notifications/escalations are automated using the workload automation and service management tools.

In any case, a modern implementation of Enterprise Workload Automation begins with identifying relevant business and/or IT services that are supported by the workflows running in the Enterprise workload. This can be as easy as identifying what the daily pain points are within an organization, e.g. daily calls about the status of backups or file transfers, etc... Then associating the automated workflows that support these pain points with in logical groups as workload services inside the Enterprise Workload Automation tool like Control-M.

The service model and its CMDB and/or data dictionaries are the source of the names required in the naming convention of a mature enterprise workload. Using these entities and mapping them to the enterprise workload workflows enables accountability, transparency and supportability.

*Best Practices for Workload Services are:*

- 1. Build a service model to ensure workload services are aligned with IT and Business services and become proactive and improve customer satisfaction.*
- 2. Align workload services with existing services in the service/support tools and systems (like the CMDB)*
- 3. Use Self-Service to expose the automation performed by Workload Automation at the appropriate levels of the organization without overwhelming details for the most relevant business and infrastructure services.*
- 4. Manage workload services over time using trend analysis and problematic job identification to enable predictive and adaptive automation.*

## Best Practices and Standards for Workload Automation Components

As an enterprise workload automation tool BMC Control-M, is both platform and technology and, in most cases, application agnostic. This means that a modern standards-based interface provides automation capabilities across all platforms, technologies and applications without the requirement for specialized skills and knowledge. The tool employs graphical user-friendly interfaces (GUIs) to standardize and simplify automation for all the tasks required to execute any well-defined process.

Control-M accomplishes this by both separating capabilities into types of automation work in the workload automation lifecycle and by using Job Editing Forms (JEFs) that “look and feel” the same no matter what type of platform, technology or application is being included in the enterprise workload.

The screenshot displays the 'Job: OS\_Job#1' editing window. The 'General' tab is selected, showing various configuration fields. The 'What' dropdown is set to 'Command', and the 'Run As' field is highlighted with a red border and a red exclamation mark icon, indicating a required or error state. The 'Parent Folder' is set to 'Folder#2'. The bottom section provides a definition for 'What' and includes 'Hide More Info' and 'OK/Cancel' buttons.

**FIGURE 6: JOB EDITING FORM (JEF)**

All best practices and standards begin with establishing naming conventions. To assist monitoring, reporting and ensure accountability and transparency, the name and format for these conventions should be chosen based on known entities within an organization whenever possible. Each standard or convention must be associated with a suggested owner and influencer. This creates accountability and facilitates understanding and collaboration. These are essential as the Enterprise Workload grows.

---

## Best Practice: Grouping the Workload for IT service alignment

---

There are four levels of grouping available in Control-M. All of them should be used to both manage the workload and manage the value of Workload Automation in an organization. These are:

Grouping	Owner/Influencer	Description
Services	Business/IT Mgmt.	Business Services, IT Services, IT Infrastructure Services as defined in a service model that require automation from workflows and jobs.
Applications	IT Mgmt./Development	The name of the application that supports the Business and IT services and/or the support team that has the Subject Matter Experts (SMEs) that may be required to notify and resolve workload issues about workflows and jobs.
Sub Application	Development/Production Control	Next level application grouping of jobs for large applications and support teams in support of the Business and IT services to notify the Subject Matter Experts (SMEs) that may be required to resolve workload issues.
Folders	Production Control/IT Operations	Groups of jobs in a common workflows that are monitored and used to design the enterprise workload.

**TABLE 1: WORKLOAD GROUPINGS**

Services and directly associated applications have already been discussed in the previous sections with the service model. Going forward the rest of this document will address the needs and capabilities of BMC Control-M to create and administer and enterprise workload.

## BEST PRACTICES: SITE STANDARDS

BMC Control-M has built-in tools that provide governance for standards and best practices. These are “tools” that enable pre-definition and automated enforcement with meaningful warning messages across the “planning” domain where enterprise workload is created and maintained.

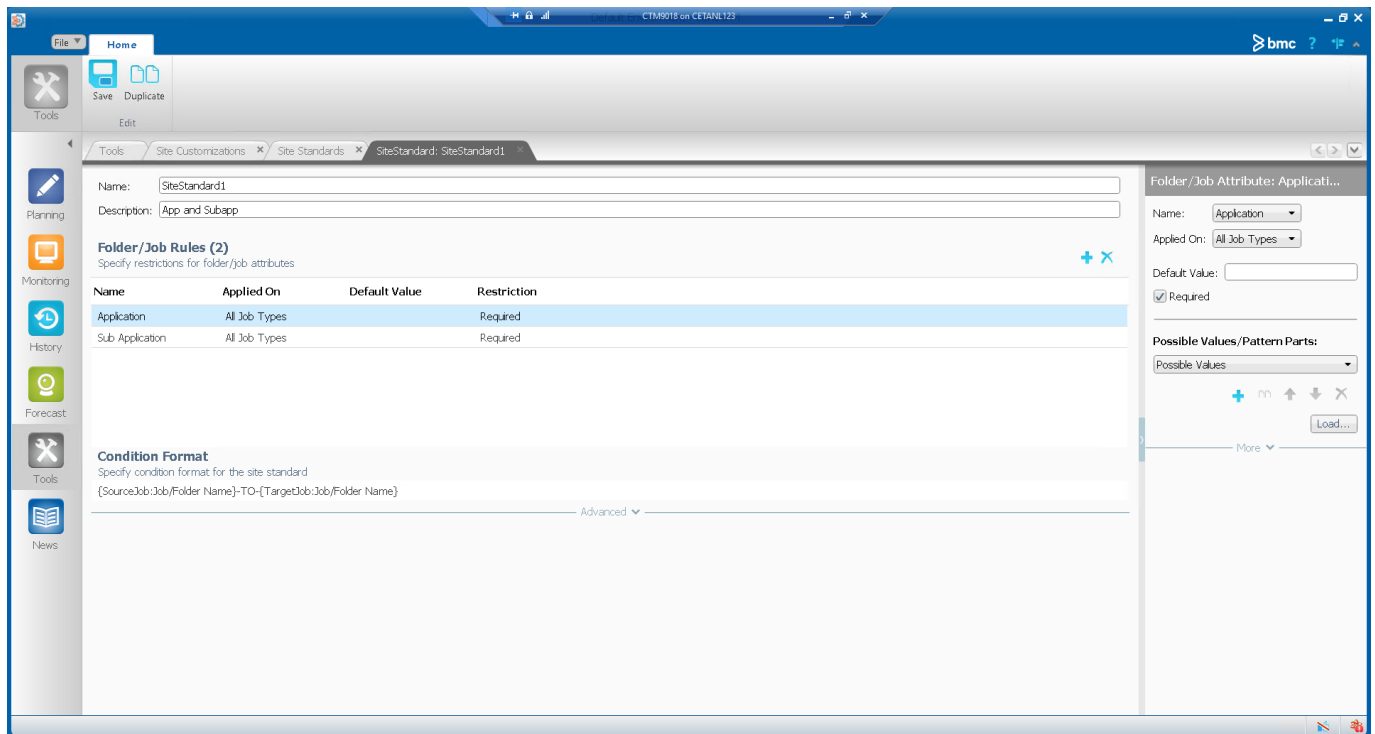


FIGURE 7: SITE STANDARDS TOOL

As discussed best practices and standards start with pre-defined naming conventions and site-specific criteria. These include ensuring workload definitions contain specific patterns and other criteria, such as environment identification with prefixes, and application and sub-application names for next-level support and service mapping and alignment.

*Best practices are to use the Site Standards tool to define and help govern the use of most standards for naming conventions. The example above simply requires some value in the application and sub-application fields of folder and job definitions, but the capability exists to have much more granular control.*

*Best practices for granularity are to be as specific as you can, without be onerous and removing all the decisions from the schedulers. In other words, there should be enough flexibility built-into the standards to allow schedulers to define folders and jobs without coming back to management with each new and/or unexpected field constraint or requirement.*



## BEST PRACTICES: SITE CUSTOMIZATIONS

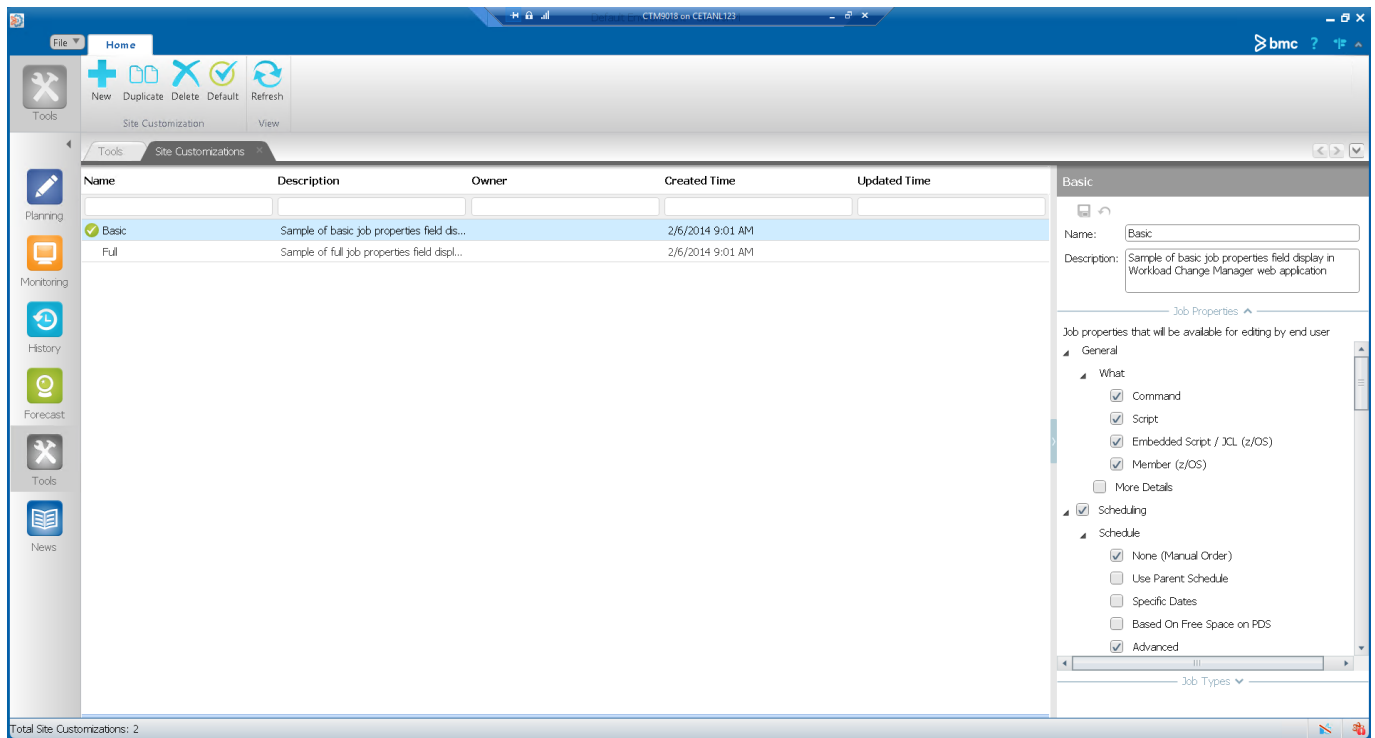


FIGURE 8: SITE CUSTOMIZATIONS

The recent introduction of Workload Change Manager (WCM) extended the scheduling and automation capabilities of Control-M beyond Production Control/Scheduling teams and IT Operations to the Application Administration, Development and Service Management teams. Adding these new roles to Control-M administration required more control of all the fields and attributes required to accurately define the enterprise workload. This is simply because the new roles do not have the perspective of the enterprise workload, this is the purview of the Workload Automation specialists in Production Control and scheduling team.

The Site Customizations tool provides extremely granular control on what and how workflow and job definitions are created.

*Best practices are to create pre-defined site customizations and continue to enhance them as the need arises and new types of automation in the form of new job types are added to the enterprise workload.*

The example above is a very basic example that shows the level of granularity available.

---

## Best Practice: Using Folders<sup>1</sup>

---

At the folder level the names chosen for the folders are critical to the management and reporting of the enterprise workload. The names of the folders and jobs are within the purview of the production control and IT operations group to help monitor and administer the enterprise workload. As such, the names should be relevant to how the workload is monitored and managed. In other words, choose names that enable understanding and identification of “WHAT” the jobs are doing in a way that relates them to the other groupings. This can be tedious on IBM mainframe and some other platforms as you are limited in the number of characters available. There is a limit in the distributed environments, but it is much more practical. The point being to choose a convention and document it so it can be followed.

For example:

In a mainframe environment where there is an 8-character limit

- First three refer to the application known to IT Mgmt.
- Next three refer to the development and/or support group required to help resolve
- Last two are numeric and are used to account for versions going forward.

There are two types of folders in BMC Control-M:

- Smart Folders – are a single unit of work similar to a single job, despite the fact that they contain multiple jobs. Jobs in a smart folder inherit many of the attributes at the folder level, including post processing and naming for application and sub-application. Rule based calendars can be created in a smart folder and will only exist for that smart folder.
- Regular Folders – include one or more jobs. They are processed independent of each other and are grouped based on naming conventions and functionality. They organize jobs according to scheduling resources.

---

### Smart Folders

---

Smart folders are “smart” because they can be used to automate a lot of manual entry for groups of jobs and/or entire workflows when all the jobs in the workflow have common attributes. There are very good reasons not to use smart folders, number one being that all jobs in a smart folder are treated as a single-entity. You can convert a regular folder by simply checking a box. You must be careful to understand the implications to each job in the folder that was converted.

---

<sup>1</sup> BMC Control-M Scheduling Folders: [https://www.youtube.com/watch?v=GUfSLd\\_zm5o](https://www.youtube.com/watch?v=GUfSLd_zm5o)

---

## **Best practices for using smart folders are:**

1. *If you have created a dummy job to start many jobs (an entire workflow or workflows) or to indicate the completion of an application where the end job varies daily.*
2. *If there are Autoedit variables common to a group of jobs but they do not warrant becoming Global variables (i.e. control the scope of variables).*
3. *Control Resources can have exclusive control at the application level versus defining them to each job.*
4. *The Active From and UNTIL feature is simpler to administer because it affects the entire flow within a smart folder*
5. *Rule Based Calendars (RBCs) can eliminate or avoid the creation of Calendars as they can be created within a smart folder and will exist only within the scope of the smart folder.*

---

## **Regular Folders**

Regular folders are can include one or more jobs and are organized according to scheduling criteria. Regular folders are normally processed independently of each. Each job is handled according to its scheduling conditions. Before Smart Folders existed, scheduling techniques had to be used to adjust workflows or start workflows at different points. If a job had the same criteria as other jobs in a folder they had to be copied and/or adjusted manually. For large job flows with similar attributes and scheduling criteria this was very tedious and required a lot of effort.

*Best practices for regular folders are to use them when the special attributes of Smart folders are not required. Each type of folder has benefits independent of each other. Most of the decision on whether to use one or the other will depend on how you decide to configure your workload, and in some cases, how it is already configured (for large and legacy implementations).*

---

## **Cyclic SMART Folders<sup>2</sup>**

Starting with BMC Control-M Version 9018 cyclic folders can now be scheduled where an entire workflow or set of jobs can be scheduled as cyclic, meaning all the jobs in a workflow run versus just running cyclic jobs individually. Like a cyclic job all the jobs in the workflow run as a single entity.

The main consideration is how BMC manages the conditions (dependencies) for the entire job flow, not just an individual job.

Because it is a new feature within Control-M there are not a lot of best practices from an implementation point of view. There are a few basic considerations based on the issues the cyclic folders feature was designed to address.

*Best Practices for Cyclic Folders: Configure what needs to occur when any or all jobs does not complete successfully.*

---

<sup>2</sup> BMC Control-M Cyclic Folders: <https://www.youtube.com/watch?v=W5Iij3kCfIY&t=152s>

---

## Best Practices: Alerting

---

The main policy that needs to be kept in mind for alerting is that we want to monitor based on exception. This implies that we don't expect someone to monitor every single job in every single workflow to watch it change state. Rather we want to only require their attention when an exception has occurred that requires manual intervention. This is the purpose of the Alerts interface in the Monitoring Domain of Control-M.

Alerting is also referred to in a broader sense as notification and escalation. These are the main reasons to use the alerts screen.

There are many ways to cause an event in the alert screen. Including some system defaults that are automatically set. (e.g. when a job fails send an event to the alerts screen).

Here are a basic set of best practices for sending alerts:

1. The situation requires human intervention (prompting someone and pausing execution until there is a response).
2. The system is providing information for a situation to be escalated to another level of support.
3. The system is providing information to an end-user for receiving notification(s). For example, when a Service Level will be missed and there is nothing that can be done about it.

A valid argument can be made that even though we can notify the people that are responsible for monitoring the workload, we can also automatically notify those that can resolve the issue. This will vary depending on best practices within an organization.

## Best Practice: Scheduling<sup>3</sup>

---

From a Control-M perspective there are four types of scheduling:

1. Automatic – using calendars and conditions to determine when a workflow and its jobs will enter the daily workload
2. User Daily – Same as automatic but the user defines at what time of the day or where in the daily workload a group of workflows can begin
3. Manual – User “orders” a workflow or job into the daily workload
4. Event Driven – a system or user event is detected that “orders” the workflow or job into the daily workload

---

<sup>3</sup> Control-M Scheduling Techniques: <https://www.youtube.com/watch?v=e2U7wJcMutg>

---

By design a scheduling tool like Control-M uses an over-arching concept like a business day where a business closes its books for the current day to determine “when” to begin and end a daily workload. BMC Control-M refers to this as automatic and/or “New Day”<sup>4</sup>.

The time of a “New Day” is configurable. When the time for a “New Day” is reached Control-M initiates some rather complex processing to end and clean-up and initiate a new day’s workload. Best practices are to keep the number of workflows and jobs to a minimum during New Day simply because this is when resources are already being used by the system and because just setting things to start when the New Day begins is not a good way to determine or understand the actual timing and resource dependencies for workflows and/or jobs.

User Dailies can and should be used to demand in substantial portions or even entire workloads based on the best time for workload/workflows and jobs run instead of just letting them default to the beginning of the workload day default, New Day. Scheduling workload in this manner provides more control and enables better understanding of how system and application/job resources are used and when they are required. It also provides easier restoration and remediation when issues occur. In other words, you don’t have a “flood” of jobs coming in just because they can run now without understanding the impact on system and application resources.

When considering the best time for “New Day” best practiced based on experience indicate the middle of the day or, in other words, the middle of an organizations daily workload is the most advantageous. This is simply because during this time resources are most consistently available and do not change. Schedulers do not have to account for other system and application downtimes and exceptions. In other words, there is usually little unusual activity occurring during the middle of the day. Scheduling the workload this way also provides ample time to recover from any events that occurred during the previous day and to prepare for the next day’s workload.

As an example of this type of recovery and preparation and how it can be used. There is a little know utility in Control-M called CTMLDNRS. CTMLDNRS basically provides an integrity check on your daily workload to ensure there are no missing jobs/conditions that will prevent the daily workload from running as expected. It is recommended that this utility is used daily to ensure and understand the status of the workload. It is especially valuable when large numbers of changes are introduced or when there are “special” processing days that require unusual adjustments to the workload.

*Best practices for manual and/or event-driven jobs is to put them in separate folders to ease monitoring administration and maintenance.*

---

<sup>4</sup> Control-M NewDay: [https://www.youtube.com/watch?v=Fh5uOUm\\_ax4](https://www.youtube.com/watch?v=Fh5uOUm_ax4)

---

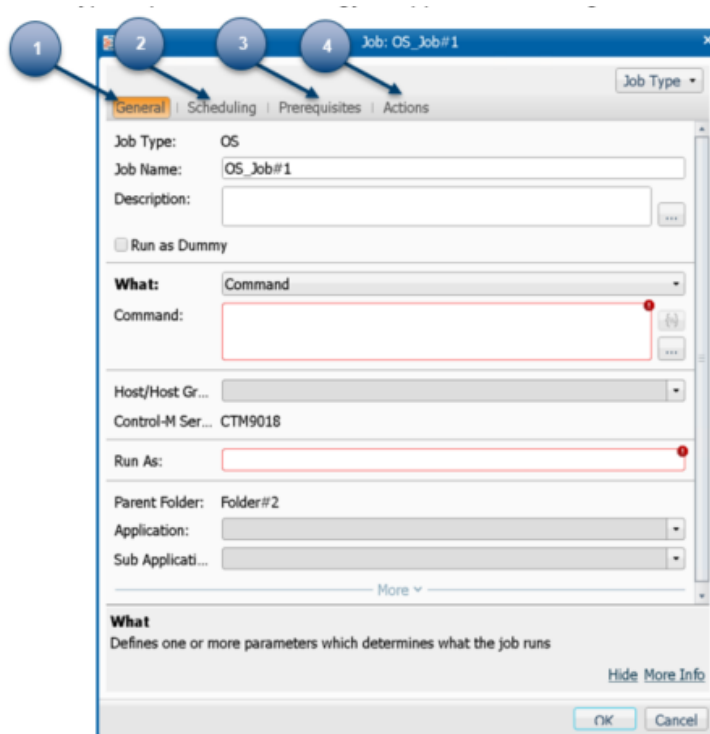
## Job Editing Form

---

A job in Control-M is not a simple command of “WHAT” to execute, although that is certainly a critical part of a job. For Control-M a job consists of both pre and post conditions and criteria, what-to-do, or the execution criteria and the post-processing tasks, which in some cases can be very complex, especially when they contain automated notification/escalation, remediation and output management.

When a job completes with an “OK” status, this indicates all the defined criteria have executed successfully. Likewise, a “FAIL” status indicates any one of the conditions or post-processing tasks may have failed, not just the job execution.

All job types are defined via the Job Editing Form (JEF)



**FIGURE 9: JOB EDITING FORM WITH TABS**

The JEF contains four tabs as illustrated above. Each Tab contains many features and capabilities. The General Tab is the most familiar as it contains WHAT to do as well as the common and often required attributes to get a job to run on a server or remote host. Likewise, the WHAT section is the section that changes with each different job type. All the other tabs and their capabilities remain the same. This is what gives Control-M the power and flexibility to administer and enterprise workload on all platforms, for any application, in any environment. Most of the criteria required to get automation to occur, is encapsulated in this form in an easy to understand format that can be both copied and updated very quickly and with little or no manual entry. This is the standard that makes Control-M such a good enterprise workload automation tool.

The four tabs in the JEF are:

- **GENERAL** – used for describing WHAT the job is and where it will run and establishes the naming conventions required by best practices.
- **SCHEDULING** – provides the date/time criteria for scheduling each job or whether it is manually “ordered” and/or requires confirmation from a human being to run.
- **PREREQUISITES** – provides dependency information and provides a method to specify other scheduling criteria beyond date and time.
- **ACTIONS** – provides the ability to specify any type of automated notification/escalation, automated remediation and/or output management Control-M supports.

With the addition of the Application Integrator tool into Control-M, the number of job types is only limited by what you need in your organization. The remainder of the scheduling section will focus on the Operating System (OS) Job Type. Most of the best practices documented apply to all job types.

**FIGURE 10: OPERATING SYSTEM (OS) JEF**

## General Tab

As expressed earlier in this document, most of the best practices for fields are around naming conventions. There are two types of names that will be referred to:

- **Long Name:** This is the field name usually more well-known across the organization
- **Short Name:** This is the same field name as the Long Name only shortened or abbreviated for levity and to keep the naming convention consistent. It is also necessary on the Mainframe and some other platforms as there is a still a limit of 8 characters.

---

**Job Name/Member** – Jobname on the distributed platforms and Member on the Mainframe.

Mainframe is limited to 8 characters. Prefixing is important for easy identification of workflows and jobs between environments.

Best practice is to prefix folder and jobnames (workflows) with an environment prefix, (e.g. PRD for production, QA for Quality Assurance and DEV for Development).

The next three or four letters should refer to the application short name, (e.g. BILL for billing).

The last one or two characters should be reserved for a numeric value so that as the workflow changes it is easier to add and take away jobs that perform similar functions in the workflows.

Building on our example:

Environment	Application	Numeric	Jobname
PRD	BILL	01	PRDBILL01
QA	BILL	01	QABILL01
DEV	BILL	01	DEVBILL01

**TABLE 2:** BEST PRACTICES JOB NAMES

In this example we are accounting for the least number of characters (i.e. Mainframe type environment). Regardless this enables a consistent naming convention that is reusable across all platforms.

In distributed environments the ability to use 64 characters exists. However, the key to best practices is to keep them simple and reusable. The point is there is no need for sentences that allow different spellings and excessive typing when we can reserve that type of information for the description field and apply a standard there also.

**Description** – a more descriptive version of the job name for additional important information.

The description field is often overlooked or even just a copy of the jobname. The description field is useful for containing information required for level 1 and 2 support personnel that is important for them to know what to do and is a searchable text field for reporting. For example, full application name, support team name, information about the type of processing (i.e. This job runs at 1800 because that is the cut off time for orders and this job sweeps the database table at that time for processing.)

*Best Practices for the Description field are to establish some sort of standard for the first few words to enable searchability and knowledge base integration. The text should be concise and consist of only information is a few words as possible that is required to understand what the job is doing and how to support it.*

**What Section** – where we define what we want Control-M to do.

One of the most frequently used OS job types is the SCRIPT type job (As seen in the figure above).

The benefit of embedding allows a single source and portability of the script code to any LPAR or Server for execution without the added administrative overhead. Likewise, when changes are required they can be made in one place at one time.

*Best practices for Script jobs are to use an embedded script whenever possible.*



---

## Host/Host Group – Where the job will run.

Host groups can be used to perform some primitive load balancing or to provide a way to take servers out of the workload for maintenance and/or other resource constraints.

*Best practices for Host Groups is to take advantage of the ability to group servers with common attributes and resource constraints to provide increased levels of availability.*

## Run As – the user name the job will execute as on the server.

This value will depend on whether the agent servers are configured to run as individual users and which user performed the install. There are many considerations especially as they relate to a particular user's environment in terms of variables and paths.

*Best practices for RUN AS user is to choose the user id's that will execute jobs on the servers or remote hosts that provide the security and reporting needs to keep you in compliance.*

## Parent Folder – the name of the folder that the job or folder is a part of.

The folder naming convention is just as important as the job naming convention. Names of the folders should be chosen to reflect the next level support and service model relationships discussed earlier in this document. The best way to look at a folder is to consider it to be an entire workflow of tasks (jobs) or a segment of a bigger workflow. Including references to the application and support team that are responsible for the work the folder is automating should be used.

*Best practices for Parent Folder or any folder name is to include some application short name and include numeric reference if it is a complex and changing workflow within the folder.*

## Application and Sub Application – the names of the applications and functional components of an automated workflow.

Application and sub-application should be the short or long names from the service model derived earlier. This information provides administrative and support personnel with the responsible party.

*Best practices for Application and Sub Application are to use the names from the service model.*

## Variables Section – The Variables section allows you to define variables that will be resolved at runtime.

Autoedit Variables are a very important feature of Control-M. An established naming convention and "scoping" (Global vs Local) need to be considered before using them.

Another advantage of the OS job type is the availability to use variables that are resolved at runtime to enable a more dynamic workflow that can be automatically adjusted based on variable values that can be updated while the workload is running.

Type	Name	Value
Local	Var1	1
Local	Var2	2

**FIGURE 11: AUTOEDIT VARIABLES**

*Best Practices are to use AUTOEDIT variables to enable dynamic runtime processing or to enable automated notification/escalation and/or remediation.*

**Override Path** – a mainframe field type that provides a secondary path for execution

The Override Path can be useful in the distributed environment for recovery and remediation information including Autoedit variables that are set based on execution results

*Best Practices for OVERRIDE PATH are to use it to provide information required for recovery or remediation.*

**Pre-Execution and Post Execution Commands** – commands/scripts that can be execute before or after a job runs.

**Documentation** – a path or hyperlink to relevant and required information for administration and support of the job.

---

Most sites today have a knowledge base with web integration via hyperlink. There is a provision within Control-M to provide a file path to a documentation file as required. The documentation field is an important part of a successful and sustainable workload automation implementation.

*Best Practices for the Documentation field are to provide hyperlinks to knowledge bases that contain relevant information required to support and administer the job definitions.*

**Priority** – specify the priority of a job when more than one of the same type can run at the same time and system or application resources are constrained.

Priority has more meaning in the mainframe environment or environments where large workloads are executing on the same server. Adopt a convention that provides flexibility in terms of how the workload can be adjusted at the job level when situations arise that require more control. For example, adopt a convention that specifies possible values between 00 and ZZ. Then define every job with a default value of “05” and then adjust as determined by the performance of the workload.

Consider that Control-M has other methods, including Quantitative and Control Resources, as well as Workload Policies, that also provide enhanced control of your workload.

*Best Practices for the Priority field are to adopt a convention then define all jobs based on the convention to provide additional workload control as required. Use Quantitative, Control and Workload Policies whenever possible.*

## **Scheduling Tab**

The scheduling tab of the JEF is where the date/time parameters of when a job needs to run are defined. Technically, it is not the responsibility of the scheduler to determine when a job needs to run as this is often strictly an application resource requirement that is not known by the scheduler. However, it is the responsibility of the scheduler to understand the impact of the new workflows (jobs) on the existing workload, especially from an enterprise workload perspective, as the individual development teams or teams requesting Workload Automation do not know about other automation running across the enterprise.

## **Calendars**

The main issue with scheduling date/time parameters is to reduce the number of yearly calendars. The addition of Rule Base Calendars within folders and jobs enables much more flexibility in defining “when” a job should run based on a set of “rules” that are not dependent on the year.

*Best Practices for the scheduling date/time parameters are to use Rule Based Calendars whenever possible and not to create calendars that must be maintained on a yearly basis.*

---

## Cyclic<sup>5</sup>

If the automated workflow running in Control-M contains a job that runs many times a day, and runs the same way, then defining it as a cyclic job has many benefits. From a management perspective and under current licensing agreements<sup>6</sup>, a cyclic job only counts as a single job against the task count no matter how many times it runs per day. From a scheduling perspective defining a job a cyclic enables one job definition for many jobs and provides special parameters and features that govern the special circumstances around cyclic jobs.

*Best Practices for the CYCLIC jobs are to use them whenever the same type of automated task executes numerous times a day with the same dependencies. Also, ensure the job and any pre and post processing requirement are well defined and, if possible, identical. Make sure to configure what happens when the cyclic job fails.*

## Prerequisites

The prerequisites tab allows all the conditions, (IN and OUT) and Control and Quantitative Resources, are defined.

**IN and OUT Conditions:** When a dependency is created by dragging a line between jobs this creates IN and OUT conditions. Control-M names these by default with from job to job names. This is a good convention to adopt as it makes the job self-documenting in terms of how it flows.

The ability to include AND and OR logic exists. The main concern is to use these types of capabilities only when standard or easier to understand capabilities will not work.

*Best Practices for IN and OUT conditions is to adopt a naming convention for manually added IN and OUT conditions that provides relevant information about the purpose of the condition using the same types of standards discussed in the Jobname/Memname section.*

**Control Resources:** Control Resources are logical in that they do not have to be linked to a physical instance of a job or folder. They are used like an “on or off” switch. In other words when a resource is available run. When it is not available don’t run. For example, the enterprise workload runs on three platforms Mainframe, Linux and Windows. The Windows platform has security updates that sometimes cause issues for the cross-platform dependencies that are built-into Control-M. By adding a Control Resource of platform type (using a naming convention) to each job based on the platform it runs on we can manage the workload when platform level issues occur. Simply remove the resource and no more jobs will run on that platform.

---

<sup>5</sup> Control-M Cyclic Folders with Cyclic Jobs: <https://www.youtube.com/watch?v=Z-Hks0fRCi0&t=84s>

<sup>6</sup> Spring 2018

---

*Best Practices for Control-Resources are to take advantage of them by understanding resource constraints and issues that can be managed at a more logical level based on over-arching knowledge or understanding. When defining Control Resources make sure to use naming conventions that enable understanding of the workflow they are affecting and why.*

**Quantitative Resources:** Quantitative Resources are logical in that they do not have to be linked to a physical instance of a job or folder. Quantitative Resources are based on the amount or number of the quantitative resource defined. In other words, if we create a quantitative resource with an amount or quantity of 10, when 10 of those resources are in use, no other jobs will run until sufficient quantitative resources are available. For example, we are running a large database workflow and we know that certain database queries and/or stored procedures take a lot more resources than the other database jobs in the workflow. We can define a quantitative resource and give more of that quantitative resources to the jobs that take longer and require more resources to make sure they run as quickly as possible and do not interfere with other jobs in the workflow.

*Best Practices for Quantitative are to take advantage of them by understanding resource constraints and issues that can be managed at a more logical level based on over-arching knowledge or understanding. When defining Quantitative Resources make sure to use naming conventions that enable understanding of the workflow they are affecting and why.*

## **Actions**

The Actions tab enables the ability to automate post-processing based on job status. There are many options and categories. The workflows that can be created with this tab generally fall into three categories:

1. Notification/Escalation – notifying appropriate personnel based on service/support within and organization and enabling escalation between support levels and to ensure customer satisfaction.
2. Automated Remediation – responding to jobs that did not complete OK. This includes the ability to set conditions, examine output, rerun, or demand in additional workflow(s).
3. Dynamic Workload – the ability to create and set variables that are used by other workflows and/or jobs to respond dynamically based on real-time analysis and conditions.

### **Best Practices for Actions are:**

1. Start with automated notification and escalation by integrating with existing service management tools within the organization and following existing service/support processes.
2. Use Control-M System Autoedit variables to standardize actions whenever possible.
3. Integrate actions with Service level requirements to become proactive/
4. Over time and as additional information and knowledge become available create additional actions to automate as much of the remediation or dynamic workflow as possible
5. Parse output and set variables that enable logic and dynamic adjustments to the workload

---

## Templates

---

Key concepts for any form of automation are to identify patterns and create reusable workflows (automation) that responds to patterns in a repeatable and predictable manner. Control-M provides the ability to create job type templates to improve the reusability of common job configurations. Using templates reduces manual effort and skill level requirements. It also enables quicker workflow administration and service/support improving the quality of work.

Best practices for Templates are:

1. *Identify patterns within common workflows and/or jobs and create templates using a naming convention that enables understanding.*
2. *Manage templates and minimize the number of job types (templates) to keep the most relevant job types available.*
3. *Use Control-M System Autoedit variables whenever possible to maintain standards and conventions.*

## Workload Policies

---

Workload policies provide a way to manage at the workload level rather than the workflow and/or job level. Workload policies are used when common characteristics across the workload have been identified and an over-arching decision and remediation can be affected rather than configuring individual folders and/or jobs.

Policies can be set in advance and also be set active for specific dates and times.

Some examples of workload policies are:

1. Need to put an application on hold (no matter how many workflows and/or jobs are affected) on Saturday so changes can be deployed.
2. An application is scheduled to move to a new server half-way through the nightly workload and need to re-direct workload to the new server.
3. There is a maintenance window where workload cannot run

*Best Practices for Workload Policies are:*

1. *Identify over-arching issues and use Workload policies to manage them*
2. *Be sure to test and make sure the policies perform as expected because they affect the entire workload.*

## Conclusion

---

This document is a compilation of common best practices that accumulated over time and based on real world experience. It is by no means comprehensive. The parts of workload automation addressed represent the most common and relevant best practices no matter what type of implementation of BMC Control-M or workload automation.