Whitepaper

Effective Release Management for HPOM Monitoring

Implementing high-quality ITIL-compliant release management processes for HPOM-based monitoring
Content

Overview ................................................................. 3
Release Management .................................................. 4
ITIL Compliance ........................................................ 6
Release Management Solution ..................................... 6
The Release Management Process ............................... 8
MIDAS Packages ........................................................ 12
MIDAS Package Definitions ....................................... 14
Package Naming Conventions .................................... 14
Package Deployment ................................................ 15
Release Rollback ...................................................... 16
Change Management ................................................. 17
Conclusion .................................................................. 19

Copyright Statement

Copyright 2012 by Blue Elephant Systems GmbH. All rights reserved. No part
of this document may be copied, reproduced, translated to another language,
transferred to electronic media, or transferred into machine readable format
without the prior written consent on Blue Elephant Systems GmbH. The
information contained in this material is subject to change without notice. All
product names are the property of their respective trademark or service mark
holders. BLUE ELEPHANT SYSTEMS and the BLUE ELEPHANT SYSTEMS logo are
registered in the US Patent & Trademark Office. All other product or service
names are the property of their respective owners. If you have any questions
about these terms please contact us via e-mail at info@blue-elephant-
systems.com.
BLUE ELEPHANT SYSTEMS GmbH, Zettachring 2, 70567 Stuttgart, Germany.
Overview

As organizations grow IT equipment evolves, work-habits change and the monitoring requirements in the IT framework become increasingly complex and dynamic. However, financial support to expand the resources required to manage these growing monitoring requirements remain limited – a situation made worse by the recognition that IT management frameworks have to be equally applied within the operations center. This prompts the question: If for reasons of quality or standards compliance, a service-management framework such as ITIL is required in the enterprise, why has it not yet been applied to HPOM?

Without a release management strategy, it is often difficult, if not impossible, to establish with any certainty what has been deployed and where it is in your managed environment. And even if it is possible to see what has been deployed, it is often hard to confirm that what has actually been implemented is what is really expected. People sometimes change things (even in productive environments) and forget to push the changes back to the test and development environments.

Currently, HPOM provides very little support for release management or change management; there is not much easy-to-use functionality available in HPOM to manage versions of a monitoring configuration. In the area of monitoring-configuration management, the only support provided by HPOM is the configuration-download feature that enables you to transfer configurations between HPOM servers.

- **Release management**
  The control of the integration and flow of development, testing, deployment, and support of software versions.

- **Change management**
  The definition and implementation of structures and tools used to control organizational changes.

- **Package management**
  A collection of software tools to automate the process of installing, upgrading, configuring, and removing packages in a consistent manner.

- **Configuration management**
  A process for establishing, maintaining the consistency of the information that exists with regard to the elements of a monitoring solution.

- **ITIL**
  Defines the organizational structure and skill requirements that apply to an information technology (IT) organization.
The obvious consequence of this is that all aspects of release management must be implemented outside HPOM. Although this is possible with manual steps that have been well-defined, it is still difficult to implement and, as a customer-specific solution represents a high total life-cycle cost.

This whitepaper examines the areas where HPOM does not fully address customer needs and problems in the area of monitoring solutions and release management. Further, it provides some background on the underlying concepts, and demonstrates how the benefits of the MIDAS release management features can be used to improve the quality and efficiency of the IT monitoring-configuration management processes whilst significantly reducing costs.

**Release Management**

In the world of software development, “release management” is the part of the software-management process that enables you to control the development, testing, deployment, and support of software releases.

In the context of HP Operations Manager (HPOM) and Blue Elephant Systems’ MIDAS product suite, the term “software” is extended to include configuration – the way you use software features and functionality to meet your own particular wishes and requirements. This could, for example, be to develop a monitoring solution to apply to the HPOM servers deployed in your IT environment.

A monitoring solution is a collection of configuration items such as policies, supporting scripts, binaries, and so on. Combined, the configuration items provide the monitoring instrumentation for a certain software product, for example, a monitoring solution to monitor Oracle instances or SAP processes. The concept of a monitoring solution is similar to the HPOM Smart Plug-in (SPI).

Not implementing (or following) release management processes can quickly lead to problems with the integrity and consistency of installations and configurations. One obvious consequence is that it can be difficult, time-consuming and also expensive to troubleshoot problems.

To understand the benefit of the MIDAS release management processes and tools, it is essential to understand the underlying concepts of release management.

The following list shows the central and most important goals of release management in software development (and monitoring configuration):
The underlying goals of MIDAS release management:

- Production efficiency (minimum downtime)
- Service quality (maximum uptime)
- Consistency (same configuration across the HPOM server landscape)
- Standards compliance (process oriented)
- Quality assurance within extended organization(s)
- Reduced maintenance costs

In the IT world, software releases are often classified into the following types: **major**, **minor**, and optionally, **patch**. The type of release serves a specific purpose, for example to introduce new features, improve existing ones, or fix bugs and patch problems. Typically, a release is denoted by a series of release numbers, as illustrated by the following examples:

- **Major Release:** Includes new features and has a version number before the decimal point, for example: 1.0, 2.0, 3.0,...

- **Minor Release:** Includes upgrades and enhancements to existing features and has an upgrade number *after* the decimal point, for example: 1.1, 1.2, 1.3,...

- **Patch:** Includes bug fixes, security updates, software patches and so on, and has a patch number after minor release number, for example: 1.1.1, 1.1.2, 1.1.3,...

In MIDAS, and more specifically in the context of monitoring configuration, releases are typically denoted by a release name that you can choose yourself, for example, “SAP Monitoring”, and a release version number, for example “1.0”, which MIDAS can track for you. MIDAS enables you to choose between major and minor releases and provides suggestions for the version number. Note that if you are using a version-control system such as CVS, the release number is not to be confused with the CVS version number. Further many changes are possible (even likely) in CVS (for example, during development and testing) before a release version of a package is created.

In MIDAS, a release takes the form of a package (for example SAP Monitoring 1.2) that is created by resolving a package definition against the actual configuration on an HPOM server; all the configuration objects that match a package-definition rule are combined into a release package. The release package includes all the data required to recreate the objects on another server, saving time and increasing efficiency when using MIDAS to test and deploy new releases.
Release Management Solution

With the product MIDAS Administrator, MIDAS offers a standard but flexible solution to the problem of ineffective or missing release management tools. MIDAS Administrator integrates release management functionality seamlessly into the standard HPOM administration GUI.

MIDAS Administrator introduces the concept of the package – a collection of objects (for example, policies, scripts, and binaries used by policies or any other kind of instrumentation) into a single entity, which is easy to create and manipulate. These monitoring packages can be stored and managed in a version-control system and, when necessary, transferred between HPOM instances as a deployment mechanism. In this way, MIDAS enables you to ensure the consistency and accountability of the HPOM configuration throughout the HPOM server landscape, whilst saving time and increasing efficiency in testing and rolling out new releases.

MIDAS provides the following release management features:

- **Package control**
  A packaging concept allows administrators to collect any configuration items into a single unit, which is easy to manage and control. You can include any file or directory in a MIDAS package.

- **Deployment control**
  Control, manage, and review which monitoring-solution packages are deployed where, when, and by whom.

- **Version Control**
  The MIDAS release management concept includes an integrated Version Control System, which is used to store monitoring...
packages externally to HPOM. The feature enables automatic visibility and control of monitoring release versions and ensures greater consistency and accountability.

- **Administrator control**
  Manage the monitoring solution deployed on multiple HPOM servers within a single centralized GUI. When combined with the deployment and version-control mechanisms, you can perform root-cause analysis on configuration-related problems far more easily and efficiently.

Combining all the benefits provided with the MIDAS release management feature enables you to facilitate the enforcement of recommended management “best practices” as encapsulated in standards such as those recommended in ITIL, and specifically those related to change and release management.

*Figure 1: MIDAS Administrator extends HPOM by adding advanced features to enable change management and release management.*
The Release Management Process

The lack of any visible and easy-to-follow release management process to control changes to the monitoring-configuration very quickly leads to a situation where details of the deployed monitoring solution are unclear or inconsistent. The resulting inconsistency can cause additional work not only for those people trying to understand the contradictory messages that the monitoring solution generates, but also for those people trying to fix the reported problems or find out why the monitoring solution no longer works as expected after the latest deployment.

Whatever the size of the company implementing a release management process is, the core factor to success is the process of documentation. In a small company, it is likely that an individual is responsible for all aspects of the end-to-end release management process; in larger companies, different teams or entire departments might be responsible for various aspects of release management. In both cases, documented processes are critical. The individual needs to keep an eye on the overall status while still being able to perform the detailed procedures that make up the management process. In larger companies, the team responsible for the various release management tasks needs to be able to rely on colleagues adhering to the same rules so that planned goals and deadlines process are met.

In many companies, lack of time or resources can often lead to a situation where the documentation requirements in the context of the release management process are not taken seriously. This is a problem since it undermines the whole release management process. MIDAS release management tools can help you solve this problem by automatically tracking who installed a package, where, and when, and saving the information for auditing purposes. Since the audit information can be recalled at any time, the documentation overhead is greatly reduced, which leads to a faster acceptance of the documentation requirements associated with the release management processes and helps to improve the release management process itself.

However, even more importantly, without a release management process in place, it is often impossible to investigate any changes that have been made or attempt to perform any roll back to a previously deployed solution.

To effectively manage the release process of your monitoring-configuration packages in MIDAS, you need to split the end-to-end release management process into the following phases:

1. **Development Phase**
   Create a monitoring configuration, create the MIDAS package definition, create a monitoring-configuration release package from the package definition, and store both the package and the package definition in the version-control system.
2. **Testing Phase**
   Upload the release package to another HPOM server, perform quality-assurance tests to ensure that the released package meets your monitoring requirements, and check the released package for completeness.

3. **Production Phase**
   Upload the release package to the production server(s) and ensure that the released version works as expected.

4. **Refinement Phase**
   If problems occur in the production environment, avoid making any changes there. Instead, make changes to the rolled-out package in the test environment and roll the updated package out again with a new package release.

5. **Retirement Phase**
   Remove the old (replaced/superseded) package from the HPOM servers.

---

**Figure 2: The life cycle of effective release management.**
Putting in place a release management process enables you to avoid the problems associated with the integrity of a deployed package. For example, if a new release is rolled out and leads to problems as a result of insufficient testing, it is often tempting for someone to make a small change to the existing configuration in the productive environment, which is then forgotten. At a later date, IT colleagues do not know or cannot remember who made the change, where it occurred, or why.

Figure 3: The 5 phases of the MIDAS release management process in details.

Ideal requirements for release management are 3 HPOM servers.

Although release management is recommended for companies of all sizes regardless of the number of servers available, the ideal release management environment would include at least three (3) HPOM servers, namely:
- **Development**  
  One HPOM server for development purposes: You use this server to create and develop monitoring configuration objects such as policies, message groups, and so on. You can bundle the newly developed configuration objects into a package that you upload to the MIDAS Web Application server. The packages can be stored here in a version-control system.

- **QA/Testing**  
  One dedicated HPOM server can be used to perform quality-assurance tests on pre-release packages. If the test phase is successful, you can roll out the package to the HPOM production servers.

- **Production**  
  One (or more) HPOM production servers make up the “live” environment. If further tests during the release management “refinement” phase establish that minor changes are required to the uploaded configuration, you can make the changes in the production environment and roll the changes back into the packages stored in CVS on the MIDAS Web Application server.

The following graphic shows how you use a separate MIDAS Web-Application server (with the CVS database) to control the complete release management environment from testing, through quality management, to the finished production servers from one central location.

![Diagram](image-url)

*Figure 4: Using a separate MIDAS Web Application server to control the release management environment.*

The basic principles of release management can clearly be beneficial to a single server environment. However, the clear separation of the development, test, and production roles cannot be effectively applied when all configurations are on a single server. Since development
changes can directly affect the production environment, more care must be taken when making even minor changes to the monitoring configuration. The scope of this document does not permit any detailed analysis or description of such a scenario.

**MIDAS Packages**

One of the most frequently encountered problems in release management concerns not knowing (or being able to find out) which version of a monitoring solution has been deployed or, if deployed, what the deployed version actually contains. In addition, even if it is possible to find out what is currently deployed, it is not always possible to establish if the deployment has been approved or planned. Often minor changes are made manually (outside of any release management process) to fix critical problems on one particular server in the production environment even though this is not recommended.

If small patches are deployed too quickly in production environments, for example, to fix problems with individual configuration items, and one or more servers are omitted by oversight from the patch-deployment process, it very quickly becomes impossible to guarantee the integrity and consistency of the monitoring solution deployed.

Partial updates of individual configuration items might solve a particular problem on one HPOM server but compromise the overall functionality of the complete monitoring solution deployed across multiple servers. The resulting compatibility issues can lead to a confusing duplication of messages or inconsistent and contradictory messages.

In MIDAS, the core concept of release management is encapsulated in the **package**. A package is a collection of objects configured within or outside of HPOM, which implement the monitoring for a service or an application, for example:

- **HPOM**
  HPOM monitoring-configuration objects can include policies, node groups, operator profiles, applications, and so on. Note that it is not a good idea to include nodes in a package since the nodes in the development, test, and production environment are often not the same.

- **Non-HPOM**
  Non-HPOM monitoring-configuration objects can include MIDAS instrumentation, configuration files, documentation, and so on.
A package is a snapshot of the up-to-date monitoring configuration.

A list of packages enables you to provide an up-to-date status of the monitoring configuration of a selected management server, which you can make available for general use and deployment. The package concept enables you to comply with recommended best practices by improving the efficiency and consistency of the deployment of the monitoring solution. For example, viewing the monitoring solution as an integrated unit, it is much easier to map changes to problems and track the history of the change-related problem to a particular package deployment.

The MIDAS release management package enables you to create an archive that contains all or specific types of HPOM server configuration, for example, node groups, policies, instrumentation, and configuration files. The archive (package) is saved in a form which allows you to transfer the package contents between servers quickly and easily.

MIDAS makes package management simple by enabling you to check the packages you create into version control, update them regularly, and make particular versions available for general release. A package release is identified by a unique tag that you assign to it. You can use the package concept to ensure that the monitoring configuration deployed on all your production HPOM servers is (and remains) consistent.
To help you ensure quality control, you can only generate a MIDAS package manually, for example, after the contents of a package have been tested and approved for release. One obvious benefit of such wide-ranging control of the package contents, package deployment, and package version is that you save time and increase efficiency when using MIDAS to test and deploy new releases of monitoring solutions.

However, there are additional benefits, too: The packaging concept enables you to improve ITIL compliance by more effectively managing any changes to the deployed solution, for example, by ensuring consistency across multiple servers and performing clean-up operations or rollbacks if required. You can also use the advanced auditing features to track who rolled out which package when and where.

### MIDAS Package Definitions

One of the biggest problems that release management tries to address concerns the integrity of the release, that is, what is in the release package that you want to roll out? Which objects do you want to include in a package, and do they belong together semantically? MIDAS uses the package definition to create a package and the package definition uses rules that enable you to define with great accuracy which objects to include in such a package.

Package definitions are much simpler to understand and use if good naming conventions are applied both to the monitored IT environment and the configuration items you use to monitor it. Indeed, the package definition can even be used as a means of enforcing compliance with defined naming conventions. Any items that do not conform to the naming conventions will be missed by the package-definition rules and, as a result, not be included in the release package.

The contents of the MIDAS release package are determined by one or more package definitions. For example, you can choose to package particular configuration elements in the HPOM database (message groups, policies, and so on), or only files that have been modified by the customer (self-written scripts). Alternatively, you can restrict the contents of the package to the MIDAS instrumentation, for example, user roles and permissions, and documentation.

### Package Naming Conventions

An effective release management process should involve the development and implementation of naming conventions, which can be applied to the individual items included in the monitoring solution. With proper naming conventions in place, the task of implementing a release management process is greatly simplified.
MIDAS uses the package definition to create a package. The package definition can be configured to include any objects. Moreover, the package definition can even be used as a means of enforcing compliance with defined naming conventions; any items not conforming to the naming conventions will be missed by the package-definition rules and, as a result, not be included in the release package.

The following list of rules are good practice recommendations for any IT environment regardless of whether you use MIDAS as a configuration-management solution or not:

- **Policies**
  Policies relating to a service or application should be defined under a policy group which includes (or is) the service/application name.

- **Policy Assignment**
  The assignment of policies should be done by means of node groups, even if there is only one node in the group.

- **Object Names**
  Inter-object relationships should be represented in the object names. For example, node groups should include details of the application/server they apply to and relate to the policy group(s) that should be deployed to them.

If you develop and adhere to naming-convention practices similar to those listed here, the creation and management of package-definitions and the implementation of a release management process based on packages will be much easier and more efficient.

### Package Deployment

Without a clear release management process it is often difficult to establish which version of a monitoring solution is currently deployed in the production environment, or if the deployed version is consistent throughout the monitored landscape. Inconsistent deployment policies can lead to confusing messages or even missed alarms, that is, inconsistencies in the package-deployment process can lead to a situation where changes have been made to the monitoring configuration, and important alarms are no longer being reported.

Being able to control, manage, and review which monitoring-solution packages are deployed where, when, and by whom is an essential part of the ITIL service-management requirement. The visibility of the package (and its contents) enables you to track which monitoring solution is deployed where at any time. You can also review the package-definition, creation, and deployment history, which greatly simplifies the process of deploying new monitoring solutions and, as a result, enables you to prevent the accidental or unauthorized change to your monitoring solution.
MIDAS package-management features enable you to compare the contents of two packages. For example, you can compare the package deployed on a development system with the package deployed on a test system. You can also compare the contents of different versions of a package stored in a version-control system. This includes binaries, for example, automatic actions associated with a policy. Binaries are compared using the checksum mechanism.

Figure 6: Packages can be created of any object and are stored as archives in a repository. As MIDAS offers transparent insight into your monitoring environment you are able to compare deployed and archived packages. In case of mismatch (*) simply deploy these packages to ensure the high level of monitoring-service quality.

Release Rollback

Change management processes require that your release management strategy enables you, in the event of problems during the deployment of a new release package, to return the HPOM server to the state it was in prior to the new package deployment. This includes restoring any
objects that were deleted during the deployment process or removing any new objects that were added.

To resolve the issue of inconsistency between packages, MIDAS has implemented a clean-up mechanism that provides an impact analysis feature which checks if any items have changed between releases and enables you to clean up and rollback to a previously deployed item.

For example, if any items have been removed from the new package that is to be deployed, MIDAS prompts the user for verification before these are removed from the server on which the new release package is to be deployed.

This clean-up functionality greatly simplifies the process of removing an installed package from an HPOM server or to return to a previous version of the package if a new deployment causes problems. This makes the back-out plan – the possibility to return the system to the state it was in before the release deployment if unexpected problems occur – a simple procedure and applicable to all releases.

Note that a release rollback plan such as the one described here will only work if all configuration is uploaded by means of mutually exclusive packages, this means there is no overlap between packages, and the same items are not present in multiple packages. MIDAS helps you achieve these aims by using the package-management feature to increase the visibility of your monitoring solutions and improve deployment consistency throughout the monitored landscape. MIDAS package-management provides an overview of package deployment in your IT landscape. The overview includes details of the package version, package contents, and package location. Taken together this can help you improve compliance with your change management and release management strategies.

**Change Management**

Change Management ensures all modifications to the IT configuration or other operational procedures are performed according to an agreed plan. This includes authorized procedures, identifying the impact of each change, a back-out procedure to reverse any changes, as well as a revision control system to keep track of older versions of configurations where possible.
Changes to the monitoring solution within the production environment should not be permitted except within the framework of a strict release management process. Changes should be permitted only in the development environment and migrated through to the production servers after carrying out due testing and quality controls. Only in this way, is it possible to ensure the quality of any changes. However, to ensure a timely response to critical problems that occur in the monitoring solution, most organizations accept the risks of making changes directly in the production environment. They also accept that this can lead to problems with regard to the integrity and consistency of the monitoring solution.

MIDAS enables you to compare the configuration of the production servers with that defined in a release-package or to other servers. This shows very clearly if any changes have been made in the monitoring configuration. This visibility makes it much simpler to compare unauthorized changes in the production environment with the original package, which went through the quality-assurance tests in the development environment. MIDAS provides help in allowing you to easily move configuration between servers. This feedback of production changes into development is critical as development of monitoring solutions normally follow a very iterative process as are demanded by today’s dynamic, ITIL-compliant environments.
Conclusion

If you are interested in using change management processes to implement a consistent and ITIL-compliant release management strategy, the MIDAS release management solution can greatly simplify your efforts. The packaging concept that MIDAS provides enables you to assemble an unlimited number of monitoring-configuration items such as policies, supporting scripts, and so on, into a single unit that is easy to define, release, deploy, and manage.

This simple but elegant solution to the problem of release management is designed to help you reduce the time, effort, and cost involved in developing and implementing a release management process, while ensuring that the implemented process complies with ITIL-related best practices and recommendations.

To summarize, the MIDAS release management solution offers the following benefits:

- Assemble unlimited and related monitoring configuration items in one easy-to-manage release package
- Enforce the automatic versioning of monitoring releases
- Simplify the deployment of new release versions
- Ensure the visibility, integrity, consistency of the currently deployed monitoring solution version (enable root-cause analysis and establish what is deployed on which HPOM servers and managed nodes)
- Facilitate the implementation of naming conventions
- Enable change management (including root-cause analysis) and roll-back processes

If you would like to know more about effective release management for HPOM monitoring or are interested in a demo, please visit www.blue-elephant-systems.com, send an email to sales@blue-elephant-systems.com or give us a call on +49 711 400 425 25.