



Best Practices for SAS®9 Metadata Server Change Control



Table of Contents

Introduction	1
SAS Partial Promotion Tools	1
Change Control Process	2
Plan for Change Control	3
Environment-Level Metadata Structure	3
Metadata Content Structure	5
Control Access to Metadata	7
Establish Process for Change Control	8
Promote Approved Changes	9
Export Content	9
Full repository promotion	10
Partial promotion	11
Document Changes	12
Import Content	12
Validate Changes	13
Verify Changes	14
Document Results	15
Conclusion	15
References	16
Bibliography	16

Introduction

IT infrastructure Library (ITIL) Service Support processes define “change control” as the process of controlling change, including requests for change, approvals, execution, and validation. “Change management”, on the other hand, is the process responsible for handling changes to IT infrastructure or IT services (Stickel 2005).

Change control is the process to coordinate changes to content in a given environment in a rational manner. Typically, organizations set up three environments – a playpen environment (Development), a quality assurance environment (Test), and a published environment (Production). Changes can be made in an ad-hoc fashion in Development (DEV) and promoted to Test (TEST) for quality assurance. After the content is validated, the changes can be promoted to Production (PROD) for broad end-user consumption. Promoted content is usually not editable by end users, and any changes must go back through the DEV/TEST/PROD process.

With SAS®9, the SAS Metadata Server is at the core of the intelligence platform. The SAS Metadata Server facilitates the sharing of content across SAS applications to provide a consistent representation of the business to all constituents in your organization. As a result, it is critical that the information that is surfaced to end users in a production environment has been rigorously protected from the impacts of change. This paper focuses on “change control” for the SAS®9 Metadata Server, and identifies some best practices for implementing processes around the usage of metadata promotion tools. This is intended for SAS system administrators who are familiar with the SAS®9 architecture and the role that the metadata server plays. We focus on how to structure the metadata environment, the tools available for handling changes in metadata, and best practice processes that control how change occurs.

SAS Partial Promotion Tools

With the release of SAS 9.1.3 Service Pack 4, the tools that support partial promotion have been enhanced to integrate both Data Integration (DI) and Business intelligence (BI) generated content. As a result, there are more options available to manage content in the SAS metadata environment.

- BI Manager plug-in version 1.4 for SAS Management Console

The BI Manager plug-in was introduced with SAS 9.1.3 Service Pack 3. This plug-in consolidates the functionality of and replaces the Stored Process Manager and Business Report Manager plug-ins, and extends functionality into the area of partial promotion. Partial promotion is supported by the ability to export selected metadata content from one repository and import it into another (or the same repository). The export results in a SAS package file (similar to a .zip file) that contains the selected content from the source metadata repository. There is a user interface for re-mapping properties to match the properties in the target repository in order to account for different paths, library names, and server names.

Best Practices for SAS®9 Metadata Server Change Control

With SAS Management Console, the BI Manager plug-in version 1.4 was extended to support DI-related content. The plug-in now provides partial promotion for the following metadata objects:

<u>BI-related content</u>	<u>DI-related content</u>	<u>Other content</u>
Reports	DI Jobs	Mining results
Stored Processes	Libraries	Metadata Tree folders
Information Maps	Tables	
Data Explorations	Generated Transformations	
	Notes	
	Documents	
	External Files	

- SAS® Data Integration Studio 3.4

SAS Data Integration Studio 3.4 has integrated the same underlying framework as the BI Manager plug-in as it relates to partial promotion, including copy /paste-special, which provides the ability to copy and paste metadata objects within a single repository as well as across repositories. The legacy copy/paste support that was in earlier versions of SAS Data Integration Studio is still supported, but we recommend that the newer paste-special function be used for promotion activities.

Change Control Process

The goals of an effective change control process for SAS metadata are straightforward:

- To minimize the impact of changes on metadata integrity and service levels
- To allow coordination and planning of changes to provide a stable production environment
- To maximize productivity of the personnel involved in the promotion process

As a best practice, the change control process should be a repeatable methodology that consists of four phases: planning, process, execution, and validation (Figure 1).

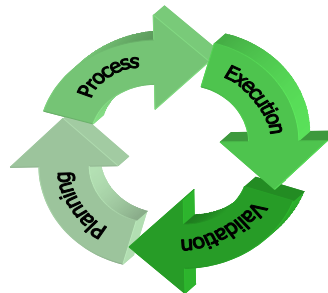


Figure 1. Four Phases of Change Control Process

As business intelligence is pushed out more and more to the end users, there is a growing demand to support both production and ad-hoc reporting requests. Corporate-level reports will still be centrally managed, but business end users need the ability to ask ad-hoc questions to perform their day-to-day jobs. This presents a challenge for SAS system administrators to control where changes are made, such that ad-hoc changes in the production system do not impact standard management reports.

Plan for Change Control

Before finalizing your SAS software deployment, it is important to take some time to plan the details of your change control process (Figure 2). Your organization's needs and system architecture will have an impact on how you should configure the metadata that is associated with your SAS environment.

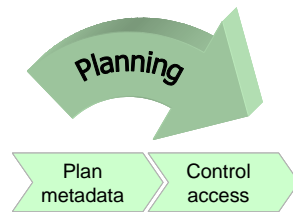


Figure 2. Planning for Change Control

It is imperative that SAS system administrators clearly define and restrict where changes can be made throughout the reporting environment. There should be clear separation between production-level reports and ad-hoc end-user reports. In SAS®9, there are two primary layers where change should be controlled: at the environment layer and at the metadata content layer.

Environment-Level Metadata Structure

As a best practice, it is recommended that you set up at least three separate functional metadata environments to support your SAS®9 deployment: Development (DEV), Test/QA (TEST), and Production (PROD) (Figure 3). Each of these environments would reflect a complete install of the SAS®9 platform, including backend application servers and mid-tier components. There is no need to have multiple installs of the desktop client applications to access these environments because a single instance can access multiple environments.

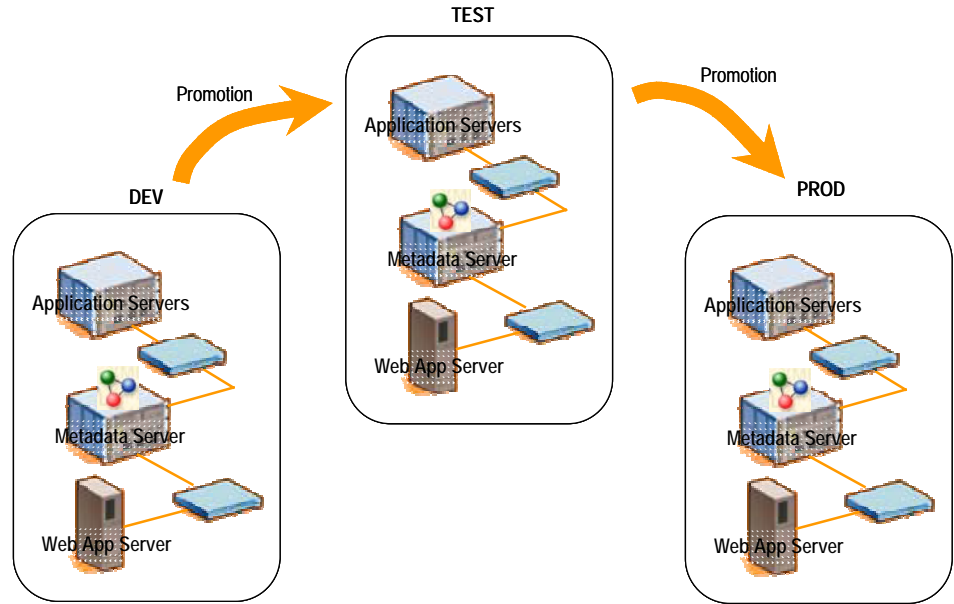


Figure 3. Physically Separate Environments for DEV/TEST/PROD

With this setup, you can completely isolate changes made to one of the environments. Staging each environment is also straight-forward because full repository promotion is a viable option as a staging mechanism. If your environment includes the SAS Information Delivery Portal, you can create an initial portal deployment in the DEV environment. Then, with full repository promotion, you can duplicate that baseline environment into TEST and PROD.

In this scenario, changes are isolated to individual environments, and processes can be put in place to manage when promotions occur. Each environment can be on separate sets of servers, or partitioned on a large server (or servers) with multiple CPUs. The distinction is that each environment is completely isolated from each other.

An alternative scenario would be to combine DEV and TEST in a single environment and keep PROD in its own environment (Figure 4). The DEV/TEST content would be segregated via metadata—which means using separate folder structures within a metadata repository tree.

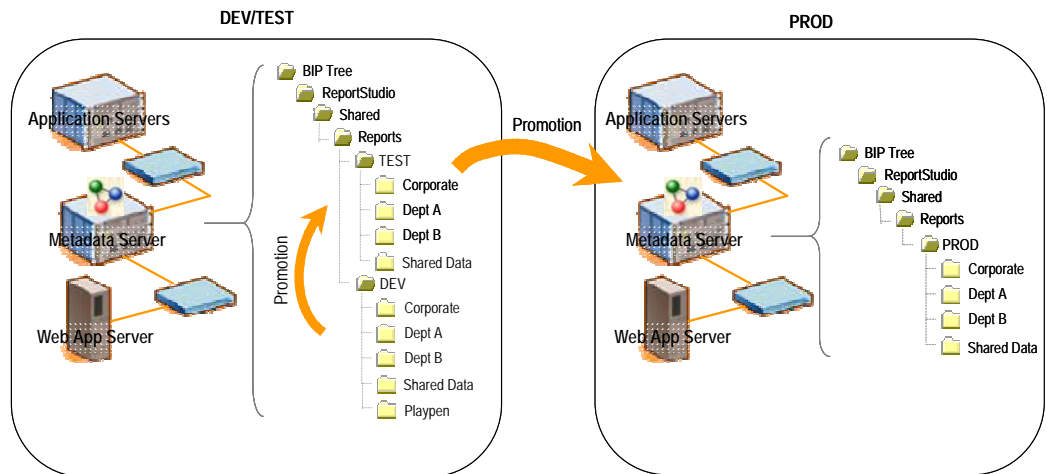


Figure 4. Combine DEV/TEST and Keep Separately from PROD

In this setup, you do not need as much server hardware to support, as you can leverage a single machine for multiple environments. Partial promotion techniques work best here, as they support copying of content both within a repository as well as across repositories. Note that the folder structures for DEV, TEST, and PROD are virtually identical. This makes the partial promotion process much easier to manage and control. The BI Manager plug-in in SAS Management Console provides partial promotion support, and it assumes that the target repository will have the same folder structure as the source repository.

It is a best practice to always have the PROD environment physically separated from your DEV and TEST environments—to isolate production content from ongoing changes. It is possible to extend the single environment to support a production environment, if you do not have resources to support multiple physical environments, but much care should be taken to set up an appropriate tree structure and apply authorizations appropriately.

Metadata Content Structure

Historically, production environments were primarily the domain of IT departments. IT controlled the corporate content that was published. Business users could only view what was available, and any requests for changes or new content were processed through IT.

With the SAS®9 Enterprise Intelligence Platform, a number of tools are provided that empower business users to create and share content. Business users have access to the right information at the right time. They can ask questions and get answers when they need them. However, this means that the production environment now needs to handle the creation and management of ad-hoc content. Therefore, business users must be granted write access to the production metadata repository.

To control the impact of changes from multiple constituencies, it is imperative that the metadata is organized to segregate corporate content from ad-hoc content (Figure 5). Changes to corporate content should only occur via established change control processes. Business users should not be able to change corporate content. Ad-hoc content should be separated, such that ongoing changes do not impact any corporate content.

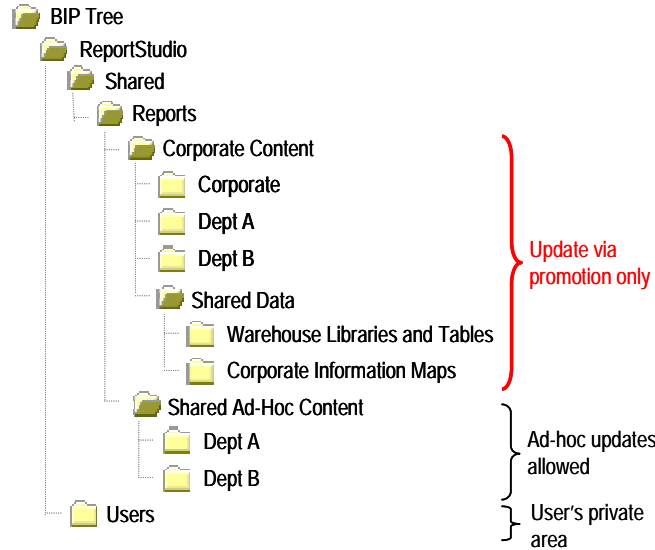


Figure 5. Example Metadata Organization to Separate Corporate Content from Ad-Hoc Content

To support a more “communal” organizational structure, you must take steps to configure your SAS®9 Enterprise Intelligence implementation, as follows:

- For SAS Web Report Studio, configure your default information maps directory to be ReportStudio¹ (instead of using the default ReportStudio/Maps), so information maps can be stored anywhere in that root folder, including /Shared and /User folders. The setting for your deployment can be set by creating (or editing) the LocalProperties.xml file, which is located in the “customer” folder under your install directory for SAS Web Report Studio (for example, Program Files/SAS/SASWebReportStudio/3.1/customer). Then, restart your Web application server (you do not need to redeploy your Web application). To make the change, open the file in a text editor. At the end of the file, directly above the </properties> line, you should insert the following text²:

```
<citation.model.repository>
  <path>
    <!-- default value: / -->
    <root>/</root>
    <!-- Note, these are appended to <root> -->
    <!-- Default value: ReportStudio -->
    <citationweb>ReportStudio</citationweb>
    <!-- Default value: ReportStudio/BannerImages -->
    <bannerImages>ReportStudio/BannerImages</bannerImages>
    <!-- Info maps are searched from this location down -->
    <!-- Default value: ReportStudio/Maps -->
    <maps>ReportStudio </maps>
    <!-- The "root" for shared files (reports subdir goes here) -->
    <!-- Default value: ReportStudio/Shared -->
    <shared>ReportStudio/Shared</shared>
    <!-- The "root" for a user's files; username as a subdirectory is
```

¹ Another option is to configure the default information map location to be ReportStudio/Shared, which is the top-level folder for SAS Web Report Studio shared content. The downside of this is that you will not have access to any information maps in the original /Maps folder.

² You can also copy this text from the WebReportStudioProperties.xml file that's located in your deployed location (for example, \Tomcat\webapps\SASWebReportStudio\WEB-INF\).

```

    <!-- created here, (reports subdir goes under that -->
    <!-- Default value: ReportStudio/Users -->
    <users>ReportStudio/Users</users>
    <!-- The subdir created in the shared/user area for reports -->
    <subdirReports>Reports</subdirReports>
    <!-- The subdir created in the shared/user area for queries -->
    <subdirQueries>Queries</subdirQueries>
  </path>

  <!-- Repositories often have more than 1 "root folder", this parameter
  <!-- indicates which should be used if there are more than one. -->
  <rootFolderName> BIP Tree </rootFolderName>
</citation.model.repository>

```

Display 1. Sample Contents from LocalProperties.xml File

- Starting with SAS 9.1.3 Service Pack 4 (Ship Event 26 or higher), both DI and BI content can be stored in the same metadata tree and can be managed together. Move content from SAS Data Integration Studio into the ReportStudio/Shared directory. (In Figure 5, the folder “Shared Data” is intended to contain metadata for libraries, tables, and information maps that are available as corporate-wide data sources.) To move folders, you can use the BI Manager plug-in or SAS Data Integration Studio to copy the appropriate folders to the new location. Once you have verified the content, you can delete the old folders from their original location.

Within the “Corporate Content” area, the folder structure should also reflect how change will be managed within your organization. If different departments will have specific content that needs to participate in the promotion process, then folders should be created specifically for those departments. If change will be managed via projects, then folders should be created for each project. Remember that the folder structure that is used in DEV must be identical to the folder structures in TEST and PROD.

Control Access to Metadata

You can use metadata permissions to restrict the ability to change corporate content. The specific settings can vary among the DEV, TEST, and PROD environments, depending on your promotion process, but all environments will need to restrict non-authorized users to write and update access to the content. To provide security for the corporate content area so that only users authorized to perform promotion activities can write or update content in that area, follow these steps:

1. Create a group whose members are users who are authorized to update corporate content.
2. Deny WriteMetadata authorization to PUBLIC users and SASUSERS on the root folder for the corporate content.
3. Grant WriteMetadata authorization to the group created in Step 1 above.

By planning ahead and structuring the folders appropriately, you can greatly help restrict the scope of change for any one area. Changes that are related to one event should minimize cross-over to other folders, or there should be a parent folder that contains all sub-folders that participate in the change. This step is important for controlling what can be changed. For more details around setting metadata authorizations, please consult the *SAS 9.1.3 Intelligence Platform: Security Administration Guide (SAS 2006)*.

Establish Process for Change Control

Written change control processes (Figure 6) are required to deter inappropriate changes from happening. Inappropriate changes include changes made without complete review and approval from stakeholders in the organization. Having established processes ensures the quality and integrity of the impacted content.

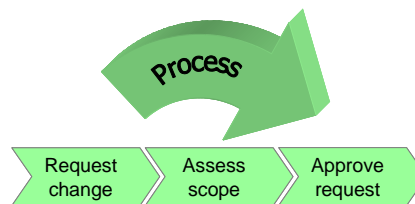


Figure 6. Establish Process for Change Control

In a survey of 40 corporate IT managers, 60% admitted that their change control processes are not effective in communicating and coordinating changes in their production environment. Gaps in the process included lack of communication, lack of thorough testing, and ineffective approval processes (Stickel 2005). Therefore, as a best practice, it is important to establish a consistent change control process that addresses these gaps. This process should include the following elements:

- Request for change – includes communication to stakeholders
- Assess scope of the change – includes testing the requested changes
- Approve appropriate changes

The request for change should be the first step of the process. This starts the change control process by identifying what needs to be changed (that is, promoted) and communicating the request to all stakeholders. The request should clearly state who is requesting the change, what objects are to be changed, when the changes should occur, and justification for the change. All stakeholders impacted by the proposed change should receive notification of the request. Gather as much detail upfront as possible, so expectations can be measured against the results of the promotion.

When requesting change, you may require that the content to be promoted has already been consolidated into a SAS package file (by using the BI Manager plug-in), and the package file is included as part of the request. Doing this upfront helps with documenting exactly what is requested to be changed. The downside is that the changes may still be fluid until the actual promotion job, so having an earlier export file would not be accurate. There are ways of mitigating

risks associated with timing by either creating the SAS package file when the promotion job is executed or by using a source code control system to manage the SAS package file. Use of a source code control system is discussed in more detail below.

The scope of the change should be assessed to determine if the change can be contained to a single parent folder. If the change crosses multiple peer folders without a clear parent (before the root of the metadata tree), then justification must be provided on the need for a wide-scope change. This should send up a warning that a change of this magnitude could impact the entire repository, so it must be reviewed closely before consideration.

Approvals from all stakeholders should be required before the promotion can occur. This ensures that the request has been communicated appropriately, and everyone has agreed on the expected outcome. Approvals should be documented for auditing and traceability. Systematic tracking of these approvals is not provided today by the SAS platform. You should implement a tracking process based on your organization's best practices.

Promote Approved Changes

After approvals have been acquired, the next phase is the execution of the changes (Figure 7). This means actually promoting the revised content from DEV to TEST or from TEST to PROD. There are three main steps to ensure smooth execution within the SAS environment.

- Export content
- Document changes
- Import content

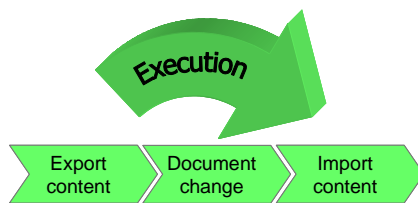


Figure 7. Execute the Promotion Jobs

Export Content

There are two main methods for handling promotion of metadata content: full repository promotion and partial promotion. Full repository promotion is typically used to stage a baseline system or promote a wide range of changes over an existing deployment on a scheduled basis. Partial promotion is used in an ongoing manner, in support of project-level changes over time.

Full repository promotion

Full repository promotion is more of an environment management tool. It involves overwriting the target environment with all the metadata content from the source environment. Any changes that were made in the target environment are lost. Therefore, full repository promotion should only be done in the following specific situations:

- Establishing a baseline system infrastructure
- Promoting an initial Portal environment (such as a baseline set of portal pages)
- Duplicating an existing environment for another application
- Scheduling a regular promotion to re-synchronize the target environment with a wide set of metadata changes.

Full repository promotions can only occur across separate metadata environments. This means that you cannot use full repository promotion to copy content from one custom repository to another custom repository in the same metadata server. Object ID references are copied during the promotion, so you cannot have duplicate ID references in the same metadata environment.

As a best practice, you should always promote all repositories in the metadata environment. Because dependent repositories contain associations to metadata content that exist in its parent repository dependency tree, promotion of the entire metadata environment will maintain the integrity of those associations.

To execute a full repository promotion, you can use the Metadata Manager plug-in in SAS Management Console or create a SAS program using the provided macros. Documentation for these tools can be found in the *SAS 9.1.3 Intelligence Platform: Administration Guide (SAS 2006)*³. These tools only handle promotion of metadata; they do not deal with any associated physical files. You should set up a complementary process to handle physical content that is executed at the same time as the metadata promotion. This could be a combination of regenerating content (such as re-running jobs and reports) and copying physical content from one location to another. The physical content that you should consider is:

- webDAV content for reports and other portal content
- source code directories for stored processes and DI jobs
- physical data tables and databases
- custom portal themes and custom portlets

³ As of November 2006, the guide is called *SAS 9.1.3 Intelligence Platform: System Administration Guide*.

Partial promotion

Partial promotion provides more granular promotion capabilities that allow fine-grained control over which metadata objects are promoted. As such, promotion is not restricted to multiple metadata environments but can be executed within a single repository, across custom repositories in a single environment, or across repositories in different environments. This flexibility allows partial promotion to be the primary mechanism for executing changes within the SAS environment.

Essentially, the act of partial promotion is exporting content from the source repository, and importing that same content into the target repository. When exporting content, SAS creates a SAS package file (.spk file extension) that is essentially a .zip archive. The package file is stored on the file system of your environment.

SAS Data Integration Studio 3.4 and SAS Management Console provide a common set of wizards for exporting metadata. The metadata objects that are supported are:

<u>BI-related content</u>	<u>DI-related content</u>	<u>Other content</u>
Reports	DI Jobs	Mining results
Stored Processes	Libraries	Metadata Tree folders
Information Maps	Tables	
Data Explorations	Generated Transformations	
	Notes	
	Documents	
	External Files	

The SAS package file provides the flexibility to store external files in addition to the metadata objects. Therefore, you can also export associated physical content with the metadata in the export package file.

As a best practice, you should strive to only promote a complete folder structure, instead of selecting individual objects. If you need to promote objects from multiple folders, you should promote the parent folder that contains all relevant content (even though this might include additional metadata that is not changing). By promoting the parent folder, you provide a well-defined container in which the changes occur. If something should go wrong with the promotion action, you can easily isolate the impacts to the container. The person who owns the request for change should be responsible for the content in the export package file—either by creating the file himself, or working with a Promotion Administrator to create the file.

For detailed documentation of partial promotion capabilities in the BI Manager plug-in and SAS Data Integration Studio, see the *SAS 9.1.3 Intelligence Platform: Administration Guide* (SAS 2006)⁴.

⁴ As of November 2006, the guide is called *SAS 9.1.3 Intelligence Platform: System Administration Guide*.

Document Changes

As part of your change control process, you should maintain a history of changes. Capturing what changed, who changed it, and when and why the changes occurred, provides a complete record that can be used for auditing and tracing.

Most organizations use a source code control system such as CVS, Subversion, SourceSafe, or Endeavor. Using such systems can provide revision history, release management control, and audit capabilities. It is typically required that source code is checked out of the source environment then checked into the target environment. Although the SAS package file is not source code, you can use a CVS system to help manage and document promotion activity.

After the SAS package file is created, you can check the file into a CVS system. Relevant information should be captured when this occurs, such as the owner ID, the date, and a description of the contents. In addition, there should be documentation about how to execute the change, which would include the following information:

- The name of the target repository
- The name of the parent folder to import into
- Any reconnection information that might be required for the target repository
- Any dependency requirements that specific content must already exist in the target repository

Changes to the package file itself should result in a new version of the complete package file being checked in. Currently, incremental change management that is provided by a CVS system cannot be leveraged with the SAS package file. Only package files that have been checked into the CVS system and documented appropriately should be available for import. When a change request is made, the specific version of the package file should be included.

Import Content

Promotions should be managed on a scheduled basis in order to isolate when changes can occur.

The import process should follow an established set of steps to ensure that all precautions have been taken for a successful import.

- Ensure that the documented dependencies exist in the target repository.

As part of the documentation process mentioned earlier, determine if the objects to be imported have dependencies on content in the target repository. In order to ensure that the import process runs smoothly and as expected, this content should be verified to exist in the target repository before the import is performed.

Dependencies can be identified during the export from the source repository. When selecting objects to export, the BI Manager plug-in will show the dependencies for a specific object being considered. If the dependent object is not already part of the export request, you have the option to add it. If it already exists in the target repository, it does not need to be included in the export file. Any inclusion of additional content in the export should conform to standards of change containment established in your organization.

If the import process includes updating existing objects in the target repository, you should make sure that the update is necessary. Importing over an existing object, such as a table, overwrites dependent metadata objects with new IDs, which could impact existing DI job mappings. If there are no changes being made to existing objects, you should not include them in the import process. The BI Manager plug-in provides flexibility in selecting a subset of objects within a folder to import.

- Back up the target location and associated physical content.

Before importing the approved package files, a backup of the target location should be made. If changes have been contained to a parent folder area, it is fairly straightforward to use the same Export wizards from SAS Data Integration Studio or the BI Manager plug-in to create a backup. By selecting and exporting the target folder, you can create a backup of both the metadata and the associated physical content in the form of a SAS package file and archive it. If the changes impact a significant portion of the metadata, you should consider backing up the entire repository.

- Launch the import wizard, and refer to the documentation for reconnection information, if needed.

Using SAS Data Integration Studio or the BI Manager plug-in, you can launch the Import wizard to load the information from the package file into the target repository. Because there might be subtle differences in the names of resources in the target repository, the import process enables you to identify the appropriate mappings. In addition, the person who executes the actual import might not be the person who owns the changes, so it is critical that the re-mappings be determined ahead of time and approved by all stakeholders.

- Review the log files for successes or failures.

At the conclusion of the import process, a log is generated that highlights the successes and failures of the import. Metadata content is imported first, and associated physical content will be imported only if the metadata was imported successfully. The log files should be archived along with the record of the change request.

Validate Changes

After the import is completed, the changes should be validated (Figure 8) to ensure that the new content is functioning as expected, and that the legacy content has not been impacted. The results of this validation should be documented.

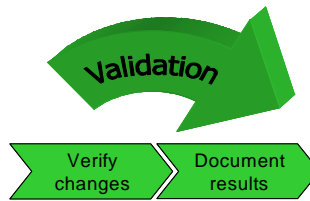


Figure 8. Validate Changes Made During the Change Control Process

Verify Changes

The owner of the change request should verify that the changes were imported appropriately and that the new content is functioning as expected. In addition, a standard regression test should be run to validate that any unchanged legacy content continues to function without problems. These steps need to be part of the standard change control process.

If there are problems found, it might be necessary to back out the changes from the target system. Depending on the scope of changes made, you should consider either restoring the repository from a backup or re-loading from a backup file created during the partial promotion process. If the changes were wide-ranging, it may be better to restore the full repository to make sure that all changes are backed out. If the changes were restricted to a target folder and a backup was made prior to the import, the backup can be imported to reverse the changes. Importing the backup will update both the metadata and the associated physical content.

Before starting the restore process, it is highly recommended that you back up the existing state of the repository, using full repository backups. This will provide you with a checkpoint state that you can return to, if you run into problems during the restore.

Here are three methods to consider when restoring metadata from a backup:

- Restore the full repository.

To restore the repository, you can use the Metadata Manager plug-in in SAS Management Console or create a SAS program using the provided macros. Documentation for these tools can be found in the *SAS 9.1.3 Intelligence Platform: Administration Guide* (SAS 2006)⁵. This approach will restore all metadata, but it will not restore any associated physical files. You will need to update those manually, if needed.

- Update in place using a backup file from the partial promotion process.

By importing the backup directly to the same target folder location, you can “reverse” any changes to objects that existed before the import. This process will essentially overwrite objects with their previous state. However, this will not delete any new objects that were created as part of the import. You can manually delete any objects after the fact.

- Delete and re-import using a backup file from the partial promotion process

⁵ As of November 2006, the guide is called *SAS 9.1.3 Intelligence Platform: System Administration Guide*.

Another method is to delete the target folder in its entirety and re-import from the backup copy. This method will return the target folder to its former state and includes deleting any new objects. Obviously, having a good backup is a pre-requisite for this approach.

Which method you use is dependent on the situation. The delete and re-import method is more risky because the content must be deleted before it is restored. If there is a problem with the backup file, you will not be able to recover the content. In either case, there should be an established process for requesting a reversal and gaining approval.

If the changes were not cleanly contained within a single folder, reversing the changes becomes more challenging. This situation makes the “delete and re-load” method difficult to justify if you cannot specifically determine or control what should be deleted. To reverse the changes to the previous state would probably require either using the “update in place” method with additional manual steps to delete any new objects that were created from the erroneous import or restoring the full repository.

Document Results

Whatever the results of the validation, they should be documented as part of the change control process. All stakeholders should be informed of the outcome. Any additional changes that might be required should go back through the process in order to provide the essential documentation that would be necessary to understand what activity is taking place in the metadata repositories.

After the promotion is completed, you should have complete documentation for what happened. This would include:

1. A copy of the .spk file that was created
2. The log file from the import process
3. The complete record of the change request, including who requested the change, who made the change, where the change was made, reconnection parameters, and existing dependencies.

Conclusion

Establishing a formal change control process is essential for managing change within the SAS metadata architecture. Because the SAS Metadata Server is the central gateway for sharing content across SAS applications, it is critical that changes to the content be managed in a controlled environment, which contains the following key elements:

- Planning – how change is managed will have an impact on how metadata should be structured
- Process – establish a clear set of processes to handle, communicate, and approve requests

Best Practices for SAS®9 Metadata Server Change Control

- Execution – establish best practices for using SAS tools to conduct the promotions
- Validation – incorporate clear validation rules and tests to ensure metadata integrity

The benefits for having an established process are:

- Better communication across the organization
- Better documentation and ability to audit the changes that are occurring
- Improved control over the content in production systems
- Minimized impacts on the SAS Metadata Server to maximize service levels

By combining best practices for process within the SAS®9 environment, you can establish an effective change control system for metadata-based content.

References

Stickel, Edward. March 14, 2005. "Change Control vs. Change Management: Moving Beyond IT". Available at www.itsmwatch.com/itil/article.php/3367151.

SAS Institute Inc. 2006. *SAS® 9.1.3 Intelligence Platform: Administration Guide*, Fifth Edition. Available at support.sas.com/documentation/configuration/bicag.pdf.⁶

SAS Institute Inc. 2006. *SAS® 9.1.3 Intelligence Platform: Security Administration Guide*, Fifth Edition. Available at support.sas.com/documentation/configuration/bisecag.pdf.

Bibliography

Amadeus International Inc. 2006. "Change Control Best Practices". Available at www.amadeussolutions.com/english/practices/bp_change_control.htm.

Kaplan, J. February 2003. "Change Management Best Practices". Available at auditnet.org/docs/CMbp.pdf.

Leopoldi, Rick, RL Consulting. June 19, 2002. "IT Service Management. Change Management Methods and Implementation Best Practices". Available at [www.itsm.info/ITSM Change Management Best Practices.pdf](http://www.itsm.info/ITSM%20Change%20Management%20Best%20Practices.pdf).

⁶ As of November 2006, the guide is called *SAS 9.1.3 Intelligence Platform: System Administration Guide*.



**THE
POWER
TO KNOW.**

SAS INSTITUTE INC. WORLD HEADQUARTERS 919 677 8000

U.S. & CANADA SALES 800 727 0025 SAS INTERNATIONAL +49 6221 416-0 **WWW.SAS.COM**

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. Copyright © 2006, SAS Institute Inc. All rights reserved. 410703.0906