

SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

Applies To:

Any SAP NetWeaver Portal implementation.

Summary

The article discusses how to keep track of portal content in large-scale implementations to improve the portal's long-run manageability.

By: Amit Levavi

Company: SAP NPI Solution Office

Date: 23 Sep 2004

Table of Contents

Applies To:.....	2
Summary	2
Table of Contents	2
Introduction.....	3
Keeping track of portal content – What is it and why should we do it?	3
What should be tracked?.....	4
Content Initiatives	4
PCD Objects	5
Manage by prefix.....	5
Manage by folders.....	5
Sharing PCD content	6
User Management Groups	6

Navigation	8
Development Elements.....	9
Enterprise Portal Client Framework Events	10
Transports	11
Summary	12
Author Bio.....	12

Introduction

Portal implementations usually start small – Activity centric portals (e.g. ESS portal for HR-related activities), role-centric portals (e.g. Sales Persons Portal), Region-specific portals or any other type of implementation based on a limited scope.

However, when implemented wisely, the benefit from the portal soon becomes apparent and these portals begin to grow and become conglomerates of content, servicing many regions, departments, roles and processes. Such portals introduce a new challenge – portal content management.

This is the first article in a series which discusses the new challenge that large-scale portal implementations introduce - “Portal Content Management”. This article will focus on the content “objects” you should track during ongoing portal operations to ensure manageability.

Keeping track of portal content – What is it and why should we do it?

Why should we “track” portal content? Why shouldn’t we simply deploy it, test it, and if it’s OK forward it to the production environment? And what is “tracking” content anyway?

Firstly, Keeping track of portal content means maintaining a reliable and up-to-date overview of all deployed portal content. This may be done with various means ranging from automated tools to a mere Excel spreadsheet, but the key is the information that is continuously maintained.

The reason for keeping track of portal content is twofold:

- **Preventive actions** – When new portal content is added, or existing portal content is changed, it might interfere with other existing portal content. For example, when setting MergeIDs for navigation

entries it is important to make sure that these IDs are only common for navigation entries which *should* be merged. But how do you know which MergeIDs already exist?

Keeping track of portal content helps prevent clashes and interference between the various portal content “pieces” which are deployed on the portal.

- **Corrective actions** – Just like with any other system, no matter what is done to prevent problems, sometimes “s#\$! happens”. In that case it is imperative to be able to determine who might be responsible for the failure, and also who should be notified about the upcoming correction.

Just so you don’t dismiss the last paragraph as a “logging issue” let me give you an example – Say an iView is re-used through delta links a number of times. Once a change is required for that iView, all the owners of the shared instances should be notified (in advance, if possible) so they would be able to check the effect of the change on their applications.

What should be tracked?

I hope that by now you are convinced that some aspects of portal content should be tracked. Let us now look at the various content “objects” that are “the usual suspects” for tracking:

Content Initiatives

The main element which should be tracked is the “content initiative”. This is not a portal term, nor is it an SAP term, but a content initiative is the overall “umbrella” under which all the activities required to set up new portal content are centralized.

A content initiative might include many pieces which, unfortunately, are not configured and/or developed in a single place, and usually also not by a single person. Exactly for this reason every content initiative should be tracked and, in addition, each such initiative must have a single contact person.

To further enhance this requirement, consider an administrator in a large-scale portal implementation which discovers an error in a portal resource deployed through one of the content initiatives running on the portal. It does not make sense in such a scenario that the administrator will have to keep track of the identity of the developer for each par file or iView, but it does make sense that once an iView is determined as erroneous the portal administrator will have someone to turn to, say, the project manager of the project who is responsible for the creation of the erroneous resource, who can then find the right developer among the developers from his team.

Give the above statement, I suggest the following guidelines:

Guideline No. 1 – Content initiatives should be recorded along with a contact person.

Guideline No. 2 – All tracked content should refer to a specific content initiative.

Just like any rule, I'll mention a few exceptions to guideline no. 2 later on.

PCD Objects

Clearly, any content initiative will surely contain PCD objects (iViews, Worksets and Roles to name a few). The task at hand then is to find a way to manage these PCD objects in a way which facilitates the corrective and preventive measures discussed earlier.

Manage by prefix

A common way to do that is to assign each content initiative a prefix which will be used for all the technical names of its PCD objects. For example, using a *com.mycompany.ess* prefix for all PCD objects related to the ESS implementation in the portal.

Naturally, if using this scheme, the reasonable way to track PCD objects would be to keep a record of the prefixes assigned to each content initiative. This way, when a new content initiative comes along you have a record of all existing prefixes and it is possible to guarantee that a “fresh” prefix is assigned to the new initiative (preventive measures). Furthermore, if a PCD object is causing problems (seen through the log for example) than using its prefix you can easily determine the responsible content initiative (corrective measures).

In my point of view this is a problematic way to track PCD objects. The main downsides of the “managing-through-prefix” method are the following – First, there is no way to enforce this scheme through authorizations (no authorization is checked on prefix level), and second that it leads to long technical IDs for PCD objects (especially the case when assigning prefixes to folder names).

While the latter downside is mainly an annoyance it might also prove as a technical constraint since the PCD objects are realized as independent files and therefore certain operating systems and file systems may impose their own restriction on the naming convention (e.g. at most 255 characters for NTFS files).

Manage by folders

A better way in my opinion is to assign a dedicated PCD folder for each content initiative. The rule is than simple – Whatever is inside this folder is the sole responsibility of the content initiative. Whatever is outside that folder is not.

When using this scheme, the way to track PCD objects is to manage the assignments of folders to content initiatives.

Implementing correct ACLs on the folders will clearly prevent clashes between content initiatives, since each initiative will be restricted to operate within its own folder. This way the portal authorization concept aligns perfectly with the PCD administration concept.

Sharing PCD content

Regardless of which of the two methods above you choose, special attention should be given to PCD content which is shared by several content initiatives. This scenario usually includes a provider of PCD content which creates PCD objects and a consumer of that content which makes use of the original content while, optionally, modifying it slightly.

Actually, sharing content is such an important subject that I should probably defer it to a separate article. However, I will conclude that shared content should be tracked on its own account.

To sum up the most important statement of this section I suggest the following guidelines:

Guideline No. 3 – Content initiatives should be allocated PCD folders which would be dedicated solely to their usage. This allocation should be tracked on regular basis.

Guideline No. 4 – ACLs should ensure that the content initiatives and they alone can create and/or modify content within their designated folders.

(Partial) **Guideline No. 5** – Sharing content between content initiatives should be tracked on an ongoing basis. Exactly how to share and what to track will be discussed in a different article.

User Management Groups

In a well managed content initiative, content is distributed to the users by a dual-assignment mechanism – Roles are assigned to groups and users are also assigned to groups. Upon logon, the portal (or rather, the UME) determines to which groups the user belongs to and from there proceeds to identify the roles associated with these groups.

The reason for this methodology is basically outside the scope of this article. However, to provide this section some support I would like to make the following claims:

- Groups are (supposed to represent) organizational entities which link a number of people (e.g. sales managers in North America), while roles are views on a particular process (e.g. “Approver” role in a sales order entry process) which combine a number of tasks in this process into a scope usually handled by a single individual.
- Assignments of roles to groups are managed by the owners of the business process (sometimes called business process owners) which decide which roles (i.e. which set of activities) are assigned to which groups in the organizations.
- Assignments of users to groups is an organizational task (performed by the manager of the group or by the HR department) which focuses on knowing where specific individuals belong to.

Ideally, groups would exist independently of applications and would represent the “real” organizational structure which exists independently from IT (yes! The organization exists even when you shut down IT! Amazing isn’t it?).

However, sometimes a content initiative is also the first time that a certain process is being managed. Therefore it is often the case that, as an integral part of many content initiatives, groups are being set up and group-administration tasks are being created.

Exactly for this reason user management groups should also be managed and tracked throughout portal operations. Furthermore, just as PCD objects have namespaces and guidelines, UM groups should have that as well to prevent interference between groups set up by different content initiatives.

However, UM groups should not necessarily be assigned to content initiatives (meaning, the portal administration may not have to keep track which content initiative initiated the set up of a specific group) and the reason for that is twofold:

- First, there is no technical correlation between UM groups and any other objects in the system. Groups are independent objects that are simply “floating” in their own “UM world”.
- Second, and more important – As stated before, ideally groups are representing an organizational notion which exists across applications. Therefore, if done in the right way, groups are likely to be re-used across content initiatives anyhow.

To sum it all up into practical statements, the following recommendations are given:

Guideline No. 6 – User management groups should be managed centrally, including namespace definitions and a governance model to allow/prevent the creation of new groups (prevention when, for example, an existing group might be adequate).

Guideline No. 7 – User management do not have to be coupled with content initiatives since they will hopefully be used across initiatives. However, an owner should be recorded for each UM group that may serve as a contact person when UM-related conflicts arise.

One reservation to guideline no. 7 – Although the groups may not be part of the content initiative, the content initiative on the other hand will not function without the proper groups since there will be no way to assign the content to the right users. Especially notice the need to align the transport mechanism of groups and PCD content between the different environments (Development, QA and production). This *would* be simpler if the groups are managed while linked to content initiatives.

Navigation

One of the “trickiest” parts in managing an ongoing portal implementation is too handle the portal navigation “scheme” without losing grasp over time. The SAP NetWeaver Portal actually provides many “neat” tools to align the navigation with required situation (merge and sort capabilities to name a few) but it is missing one major capability – providing a global overview of all navigation entries that already exist in order to determine the right place for new “incoming” navigation entries.

Consider the following scenario – Your “up and running” portal provides, among other things, ESS services, which currently contain “My salary” and “My Time Reporting” options, as seen in the TLN extract below:



A new initiative from the finance department is being launched which is aimed at showing the employees their stock options. They would like a new tab under the ESS tab called “My Stocks” which would appear between the “My Salary” and “My Time Reporting” tabs. The portal administration team is told to facilitate the new initiative with their navigation requirement, but how?



From a technical perspective, the new initiative should be given two things:

- The technical details in order to hook onto the “My ESS” tab – Namely, the MergelD of the “My ESS” entry.
- The technical details in order to fit between the “My Salary” tab and the “My Time Reporting” tab – Namely, a sort priority number which would be greater than the former and lower than the latter.

Sounds easy, right? Only from a technical perspective!

The real challenge in performing the abovementioned steps is to discover what are the right numbers (for MergelDs and sort priority values), and even more than that, to discover if there are other content initiatives that might already “be around” the same navigation area (e.g. a “My Leased Car” tab that also happens to be between “My salary” and “My Time reporting”. This introduces a new question – Should “My Stocks” come after or before the “My leased car” tab?).

To rise to this challenge one must keep track of all navigation entries created by all content initiatives. Keeping track of navigation entries means recording (at least) the following attributes of each created navigation entry:

- The navigation path to that entry (e.g. “My ESS” \ “My Stocks”)
- The merge properties (mergelD and merge priority) for that entry
- The sort priority for that entry
- (optionally) The content initiatives which make use of this navigation entry.

Notice that without keeping track of the content initiatives which make use of each navigation entry it is still possible to determine the right attributes for new navigation entries. However, if a change should be made in an existing navigation entry than it’s almost impossible to find the initiatives that should be notified.

As usual, to sum up this section:

Guideline No. 8 – Navigation entries created by content initiatives should be tracked on an ongoing basis.

Development Elements

Content initiatives rarely “settle for” mere PCD content and usually include some form of coding as well. The coding can take the more standard forms of PAR files or Web Dynpros (which would be incorporated into the portal using Web Dynpro iViews) or they may include any other form of development projects incorporated into the portal through the various integration capabilities (the application integrator, XML\XSL-based rendering to name a few).

Regardless of the exact form the development assumes, there are some things which should be considered when done in the context of large-scale portals.

Enterprise Portal Client Framework Events

Enterprise Portal Client Framework (or EPCF for short) is a Javascript-based framework, aimed for allowing loosely decoupled iViews to interact on the client side. The framework exposes a Javascript API which provides a “publish & subscribe” mechanism to facilitate this interaction.

When using EPCF to communicate between iViews, the sender iViews publishes an EPCF event while declaring its namespace, event name and, optionally, some parameters. It looks something like this:

Raise an event	<pre>EPCM.raiseEvent("urn:com.sapportals.portal:testApp", "test", "myData", "mySenderID");</pre>
-----------------------	--

The receiver iViews subscribes to the event by notifying the EPCF mechanism of his “wish” to do so:

Subscribe event	<pre>EPCM.subscribeEvent("urn:com.sapportals.portal:testApp", "test", myHandler);</pre>
------------------------	--

The fact that this mechanism does not require explicit connection between the receiver iView and the sender iView makes it possible to “hook up” to events which were created by another content initiative. This can be extremely beneficial when wanting to extend the capabilities of existing content (e.g. It’s very common to subscribe to the events published by the MSS business package’s team viewer iView in order to extend this package’s offerings).

However, the same quality also introduces the risk of “accidentally” subscribing to events you did not intent to listen to. This is most likely to happen when using very generic terms for event namespaces and event names such as

```
EPCM.raiseEvent ("urn:com.mycompany:portal", "entry_chosen", myData).
```

Clearly, many content initiatives might use this very generic term (if allowed to do so) which might cause an unwanted interaction between the iViews of these initiatives.

Therefore, the following guideline is highly recommended:

Guideline No. 9 – EPCF namespaces should be centrally administered. A new content initiative that would like to use EPCF would apply for a namespace which would be assigned solely to that initiative.

Transports

As stated before, more often than not “portal content” is not restricted to being developed in the portal but also incorporates other systems and tools. For example, portal content which is based on Web Dynpro iViews which make use of newly developed RFCs in a backend R/3 system actually consists of three transport routes – the PCD content, the Web Dynpro application and the backend RFCs.

Unfortunately, the transport management tools for all of these different types of objects are not yet integrated. However, this does not mean that we should not (manually) align the various transports, otherwise it will be very difficult to understand what is the state of each initiative. If, in “older” systems you were used to declaring the state of the deployment to production by saying which transport number is in production, than now, with multi-tiered applications, the state of deployment becomes multi-dimensional (the combination of the transport state the of PCD content, the transport state for the Web Dynpro application and the transport state for the RFC content).

Therefore, it is advisable to keep track of the following attributes:

- Manage releases for content initiatives.
- Each release should be a combination of transport states for each transport route of the content initiative.

As a way-of-working with releases, it is advisable to align the transports to the QA and productive environments in such a way that these environments will only maintain transports states representing a certain release (meaning not to transport the PCD content before the Web Dynpro application is also ready to be transported). Otherwise, what is being tested (in QA) or executed (in production) is not a “real” release but rather a transient and irreproducible application state.

To sum it up:

Guideline No. 10 – Content initiatives which encompass several development environments should align the transports in those environments into releases.

Guideline No. 11 – Only complete releases should be tested in the QA environment or executed on the productive environment.

Summary

Big portals which do not keep track of their content are bound to end up in a mess. In this article I have tried to outline several aspects which are worth the effort of “tracking” in order to make the portal more manageable on the long run.

However, this list is far from complete. There are certainly more things to track and more ways to do so than described here. I would be delighted to receive feedback and I promise to incorporate it (along with your name ☺) in my next articles concerning big portals, so feel free to send your feedback to amit.levavi@sap.com.

Best of luck with *your* big portals.

Author Bio



Amit is an ESA solution architect with over 8 years of IT experience as either an SAP customer, an SAP partner and, over the past year and a half, an SAP employee. He holds a B.Sc degree in industrial engineering and is (really) trying to complete his M.A. in mathematics (but doesn't have the time). He is based out of Leiden, the Netherlands.