



PNMsoft Knowledge Base

Sequence User Guides

Deployment Best Practices

Oct. 2014

Product Version 7.0 and above

© 2014 PNMsoft All Rights Reserved

This document, including any supporting materials, is owned by PNMsoft Ltd and/or its affiliates and is for the sole use of the PNMsoft customers, PNMsoft official business partners, or other authorized recipients. This document may contain information that is confidential, proprietary or otherwise legally protected, and it may not be further copied, distributed or publicly displayed without the express written permission of PNMsoft Ltd. or its affiliates.

PNMsoft UK 38 Clarendon Road Watford Hertfordshire WD17 1JJ

Tel: +44(0)192 381 3420 • Email: info@pnmsoft.com • Website: www.pnmsoft.com

Microsoft Partner

Gold Application Development

TABLE OF CONTENTS

General Document Information	1
Purpose.....	1
Prerequisites.....	1
Overview	2
Deployment Concepts and Best Practices	2
Creating a Package	3
What is a Package?	3
How is a package created?	3
What files should I attach to the package?	4
How should the Sequence package and other items be grouped together?	4
Should the developer write a deployment/package guide?	4
Checking the package in the Testing environment	5
How to check that the package is OK	5
Pre-requisites to Deployment	6
Additional Recommendations	6
Rollback:	6
Deployment.....	7
How to import the package:	7
Single Server.....	7
Multi-Server	7
What items should I import?.....	8
Scenario 1: First Deployment Ever	8
Scenario 2: Subsequent Imports	8
Scenario 3: You have changed one activity / one form	8
Order of Deployment for Complex Scenarios.....	8
External Objects	9
What are External Objects?	9
How can I identify which database objects I need to attach?	9
Deployment steps:	9
Selecting objects to be imported:	9
Map permissions/recipients screens:.....	11
Config Files	11
Post Deployment	12
Post deployment checks:	12

General Best Practices	13
Troubleshooting	14

General Document Information

Purpose

The purpose of this document is to enable Sequence Administrators to deploy workflow packages from one environment to another in a smooth, efficient way, using best practices.

Prerequisites

- Intermediate knowledge of Sequence.
- You must be a Global Administrator.

Overview

Sequence enables you to quickly and efficiently package business process applications and deploy them from one environment to another. Deployment is a common aspect of almost any BPM project, as work is initially conducted in a Development environment, is then moved to a Testing (UAT) environment (and sometimes a Staging environment), and only then to the Production environment.



If done correctly and according to best practices, deployment should be a smooth procedure, enabling work to continue with the process in the new environment.

The intent of this document is to describe how to perform deployment using best practices, to minimize any issues, and ensure a smooth transition.

Deployment Concepts and Best Practices

The following sections include our recommendations for the major elements of deployment, with focus on common scenarios. These should be viewed as guidelines – each deployment may have specific needs which override these recommendations. For questions and assistance, please consult PNMsoft Support.

Creating a Package

What is a Package?

A package is a portable export of the workflow in its current state. It includes all the workflow definition objects and their related objects.

Sequence knows how to include many of these objects automatically. In some cases, you as the administrator must tell Sequence if and how to include additional objects (these cases are detailed below).

The package is a set of XML files that are wrapped up as a zip file. A package can be imported into another Sequence environment.



How is a package created?

It's easy to create a workflow package using the Version Management wizard.

Workflow Version Management

Process Name
AdStream

Owner
Paul Hedgeland

- [Restore Workflow](#)
- [Import Workflow as a New Version](#)
- [Duplicate Workflow](#)
- [Create New Version](#)
- [Create Workflow Package](#)

Version Name	Comments	Stored Date	Download	Version Details	Delete
B	A new version of the AdStream workflow	26/10/2014	Download Version	...	✕

Finish

Workflow Versions Management Wizard

[Click here](#) for a step-by-step guide which describes how to create packages and perform other version management actions.

What files should I attach to the package?

Any file that is relevant to the workflow (such as css, js, txt, xml) should be attached to the package, including images that are used in forms/tasks.

Note: We advise using external css and js files for better management and performance.

To ensure these files are attached to the package, before creating the package, place all such files in the Shared Resources folder (usually under C:\Program Files\Sequence\Shared Resources). Then, in the App Studio, select Manage > **Attach Objects** to attach these objects to the workflow.

See the [Version Management Guide](#) for how to select External Objects during package import.

How should the Sequence package and other items be grouped together?

We recommend to create a *Development* folder and a *Release Candidate* folder, and to place the package (and any other relevant files) in the release candidate folder.

As a developer, when creating an application, we highly recommend that you maintain a list of all external objects that are relevant to your application. This will ensure that you do not forget any objects when performing export.

We suggest creating a zipped folder with all workflow artefacts.

Should the developer write a deployment/package guide?

Yes. Any deployment should be documented using the PNMsoft Deployment Template document. The developer should complete the relevant details in the template sections. These sections include:

- Summary
- Package Description
- Change Requests
- Issue Fixes
- Manifest
- Pre-requisites for Deployment
- Target Servers
- Pre-deployment Actions
- Workflow Deployment
- MI Deployment (if relevant)
- Configuration Files Changes (if relevant)
- Hard coded URLs (if relevant)
- Post Deployment Actions
- Post Deployment Checks
- Recovery Plan



Checking the package in the Testing environment

The package should be created in the Development environment (Dev), and part of the testing of the package should be to deploy that package to the Testing environment. If it doesn't import, or problems are found, then the whole cycle should be restarted from the Dev environment, not fixed from the Testing environment.

Execute sanity tests of the application including:

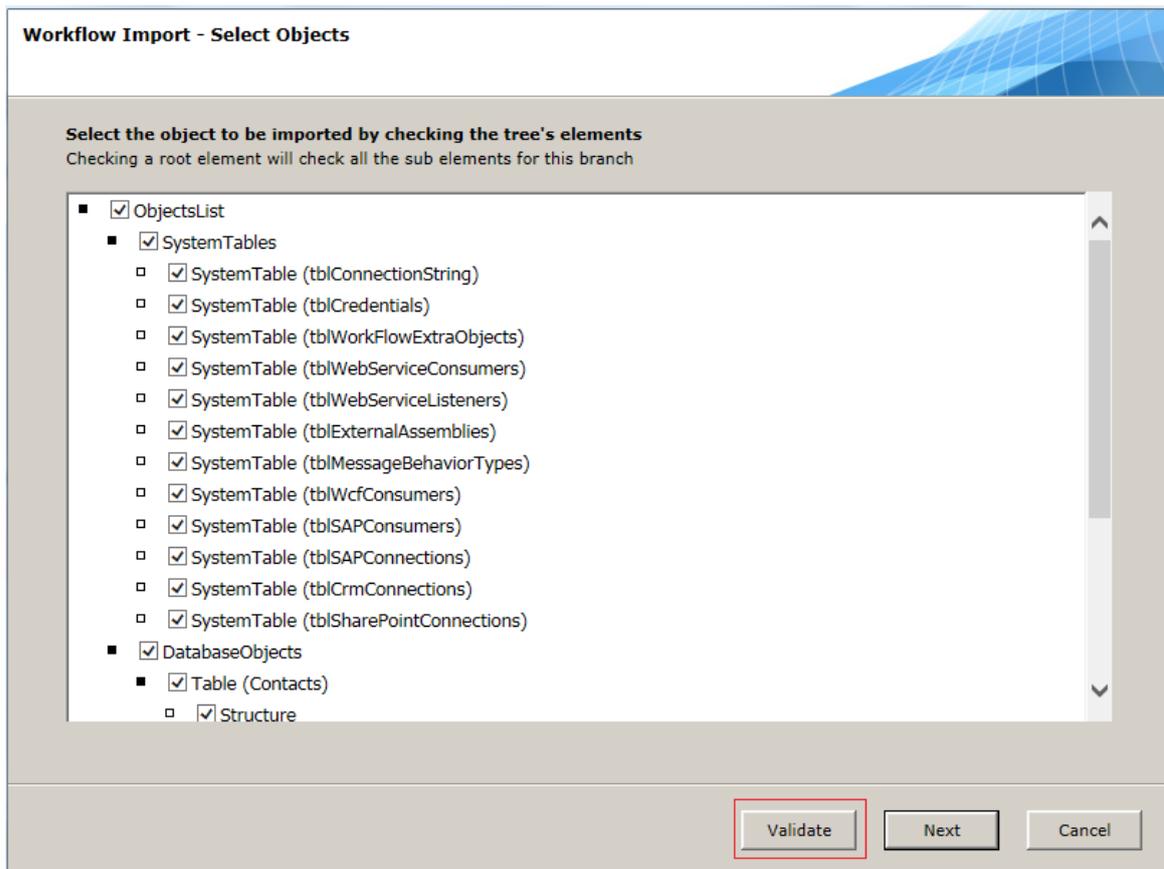
- Portal (runtime)
- Forms
- Integration
- Other (if relevant)

to verify that the packaging was done correctly.



How to check that the package is OK

When you import the package (for example, in your testing environment), in the *Workflow Import - Select Objects* screen, click **Validate** to validate the package before import. Any errors will be displayed.



Package Validation

Pre-requisites to Deployment

Before executing a deployment, ensure the following steps are performed:

- Access is scheduled (via remote desktop) to the Production SQL Server for ~4 hours, if possible also to all the environment servers. This is important in case you need to run SQL scripts post-deployment or if DLLs or files need to be deployed to the servers, OR if you need to make changes to the config files.
- Backup of database confirmed by client DBA team.
- Backup the current production state (workflow and files). Download and save it locally.
- Assemble any data you will need from the client to validate that the workflow / forms / MI is working correctly.
- Full VM snapshot / backup, in case of major changes to the application.

Additional Recommendations

- For high-volume applications, schedule system downtime and plan to deploy during off-hours.
- For applications that were developed using DLLs and/or external code, deploy during off hours.
- Stop all BRS instances before deployment.

Rollback:

Before performing deployment, declare a rollback date and time. If the deployment does not succeed by that time, perform a rollback.

There are several possibilities for performing a rollback. For example:

- Restore a backup package of the original application.
- Restore a VM image.
- Restore Database, etc.

Deployment

How to import the package:

See the [Version Management guide](#) for a step-by-step guide on how to import a package.

The strategy for importing a package effectively depends on your environment topology. The section below describes the recommended steps for importing a package for several common topologies:



Single Server

Import or restore (depending on if it is your first or subsequent deployment), or import as a new version.

We recommend importing as a new version (assuming this is not the first deployment), unless the version was already created in Dev, or the process was not developed with version orientation (i.e. there are explicit references to workflowID or activityID).

Multi-Server

Ensure all of the above for single server, and also verify that external files, DLLs and config changes are deployed to all servers in the environment.

Here are examples of common multi-server topologies and our deployment recommendations for each:

Topology #1:

Administration and Runtime on one server

And two or more clusters like the above.

1. Perform full import on the first server.
2. On the second server, import only external objects.
3. Any object that is not automatically deployed within the package should be deployed manually on both servers (e.g. DLLs, config file changes).

Topology #2:

One Administration server, One Runtime server.

1. Perform full import on the Administration server.
2. Manually copy files from the package to the Runtime server (all files that were attached as external objects).
3. Any object that is not automatically deployed within the package should be deployed manually on both servers (e.g. DLLs, config file changes).
4. *if the BRS is installed on another server (not the Admin or Runtime), then you need to manually deploy DLLs and config file changes to that server.

Topology #3:

Multiple Administration servers, Multiple Runtime servers.

1. Perform full import on the first Administration server.
2. On the other Administration servers, import only external objects.

3. Manually copy files from the package to all Runtime servers (all files that were attached as external objects).
4. Any object that is not automatically deployed within the package should be deployed manually to all servers (e.g. DLLs, config file changes).
5. *if the BRS is not installed these server (not the Admin or Runtime servers), then you need to manually deploy DLLs and config file changes to these BRS servers.

[Click here](#) for a guide on installing Sequence using various topologies.

What items should I import?

This depends on the scenario. Since there are several possible scenarios, we have included the recommended deployment strategy for three common scenarios:

Scenario 1: First Deployment Ever

For first deployments, you will most likely want to deploy all workflow elements and relevant objects, including:

- Permissions
- Reference Data

You should carefully verify that you have included all necessary items, including all external objects, files, views, and stored procedures.

Scenario 2: Subsequent Imports

Usually it is best to import all items except for reference tables data. This is important in cases where you manage the reference tables in production, and you do not wish to overwrite them.

Note: Consider selecting or deselecting Permissions if you have made changes to permissions in the source environment.

Scenario 3: You have changed one activity / one form

Here you will probably want to import only the workflow elements (and not external objects).

Order of Deployment for Complex Scenarios

In complex applications with multiple objects, tables, views and stored procedures, the order in which Sequence places the objects may be important.

In such cases, it is recommended to deploy in stages: first all tables, then all views, then all stored procedures, then the rest of the objects (activities, permissions, reference tables). Why is this important? Because you may have dependencies between different SQL objects, that force you to build them in a certain order.

In very complex scenarios, you may even want to split the tables import stage into sub-stages, and import the tables one after the next.

External Objects

What are External Objects?

External objects are files (such as js, css, other files, etc.) and/or database objects, which are important to the process execution, but which are not directly linked from the workflow definition.

How can I identify which database objects I need to attach?

Automatically included: Any database object that was defined as part of the workflow development is automatically included by Sequence. For example, all the UACT tables, UTCMB tables, stored procedures which are used in an activity or SP query (and the objects that are dependent on them).

Note: Do not explicitly include database objects that are included automatically. This can cause import failure.

What objects are not included?:

All other database objects that are relevant to the workflow execution.

For example: views or tables that are used by a stored procedure.

Or

Views and stored procedures that are used in an SQL expression.

Or

Stored procedures or views that are used directly in the form code (client or server).

Or

SQL functions that are used to retrieve recipients.

Deployment steps:

This section includes best practises for the actual import step within Sequence's Version Management wizard.

This wizard includes the following import options:

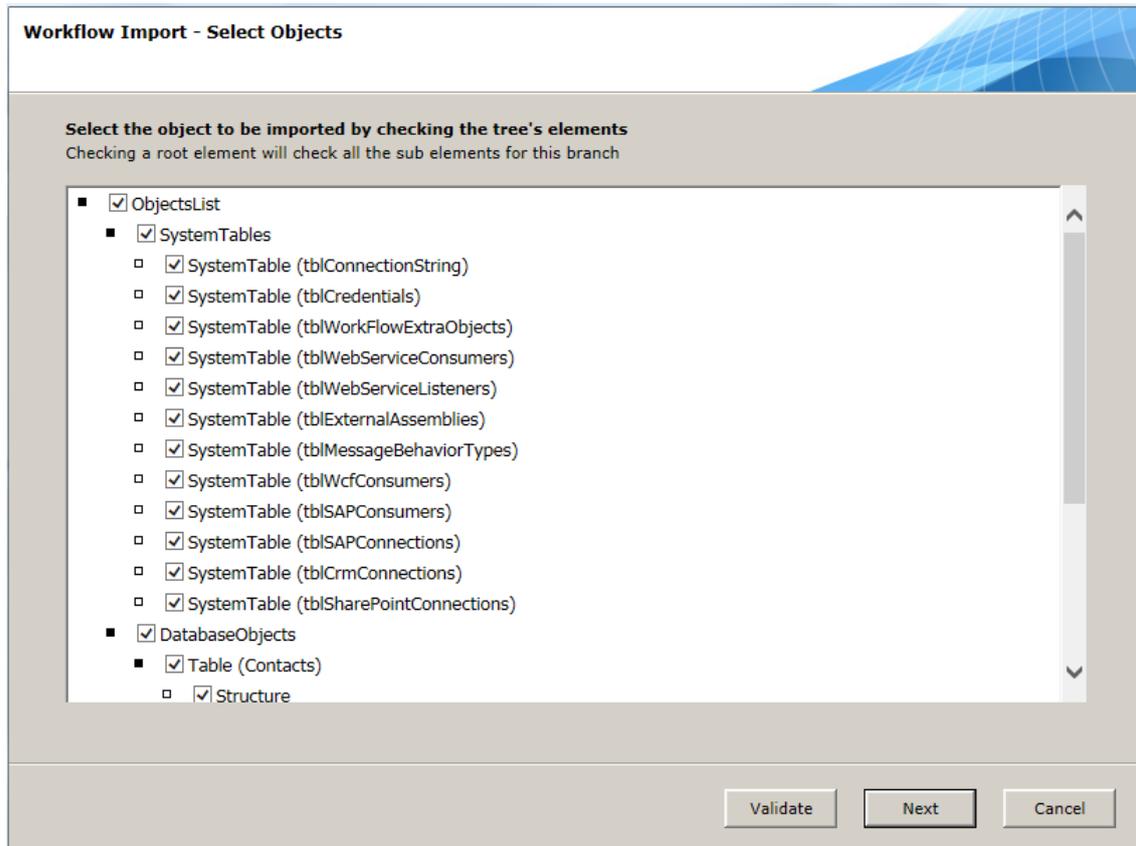
- **Restore a Workflow:** overwrite the existing version with the packaged version.
- **Import Workflow as New Version:** do not overwrite the existing version. Import the packaged version as a new version.

See the [Version Management guide](#) for a step-by-step description of how to perform each of the above actions.

Either of the above actions may be appropriate for you. Keep in mind that the first option, Restore a Workflow, will overwrite your active workflow version, and may delete runtime data. If possible, we recommend importing the workflow as a new version, and then, when you are confident this version is functioning properly, set this version as the active version.

Selecting objects to be imported:

The Version Management wizard includes a *Select Objects* screen which enables you to select which objects will be imported from the workflow package. You select the objects to be imported by checking the tree's elements.



Select Objects Screen

The following section describes the meaning of each of these objects, and the recommended selection based on your requirements.

- **System tables:** All Sequence endpoints (=record in a global settings table). These are imported into the Administration > Global Settings. For first deployment, it is recommended to import these, and then edit them manually to fit the target environment. For subsequent deployments, it is not recommended to import end points, as you will overwrite the existing endpoints. But, if you have a new end point, you will need to import that table and then fix all other entries in that table. For each table, if no new record was added to the table, uncheck the table.
- **Workflow Elements:** Activities. Best to leave as is (the default is to import all activities).
- **Database Objects:** Stored procedures, tables that are connected to activities. Leave as is.
- **User Data Models:**
Here, for UTCMB tables, you must decide if to import them or not.
For example:
If the table is managed only in Production, then uncheck them here.
If you want to entries that were removed from Dev to be removed from Production, select **overwrite**.
If you simply want to update the table with changed and new entries (but do not want deleted rows to be removed), select **incremental**.
- **External Data Models:** Leave as is.

- **Workflow Permissions:** If you have a complex permission structure that was planned and tested, check this option. In any other case, it depends on your development methodology.

Note: Permissions that already exist in Production will not be removed.

- **External file objects:** Other files that were added to the application using “Manage > Attach Objects” in the App Studio (e.g. js, css, images). If these files were updated in Production, select **skip if exists**, otherwise, select **overwrite**.

Map permissions/recipients screens:

These screens enable you to map users and OUs (organisation units) from the source to target environments. Map users and permissions as relevant to your project requirements.

Config Files

If you have made changes to config files in the source environment (e.g. web.config for admin, web.config for Flowtime, brs.config), it is recommended to make these changes manually to those files in the target environment (and not to copy the files themselves). Before making changes, it is recommended to create a backup of the config file you are changing, in case you need to rollback changes.

Changes to Message.xslt must be made on a copy on not on directly on the file. Once you have created a copy and made changes to it, you can point to the new XSLT folder in the BRS config file.

[Click here](#) for more information about configuring the BRS config file.

[Click here](#) for more information about the Flowtime config file.

Post Deployment

Perform the following steps after deployment:

1. If you have imported a workflow that includes a change in a service schema, open all workflows that use that service and click Development > **Reload**.
2. If you have external code (DLLs) that have been changed as part of your deployment, perform IIS Reset on all servers in the environment.
3. Start all BRS instances.
4. Access each front end server to verify that it is up and running.
5. Access the load balancer (if it exists) to verify that it is up and running.
6. Check that Application servers have BRS, ADSS and any additional custom services are up and running, and that there are no errors in the event log related to these services.

Post deployment checks:

If possible, run an instance end-to-end and verify that all functionality is intact.

Perform any other relevant sanity tests.



General Best Practices

Here are several general tips for improving deployments:



- Be sure to attach objects that are not part of the workflow as external objects. This way, when you export-import a workflow, Sequence will know to bring these objects into the target environment. E.g. image files, DB views and stored procedures that were created specifically for the workflow. Use the “Manage > Attach Objects” option in the App Studio.
- Ensure that the Testing and Dev organisation structure are synchronized with the Production organisation structure. This will ensure smooth automatic mapping by the import process.
- If your workflow includes JavaScript, when you create a new workflow version, use JavaScripts that match that version. The path of the JavaScript file should include the GUID of the workflow template.
- Ensure that Testing data is not imported to the Production environment using reference tables.
- Periodically, consider deploying from Production back to Dev, in order to “refresh” your Dev environment.
- When the change is scoped (i.e. a very specific change), you can simply make the change in all environments manually (Dev, Testing (UAT), Production) without packaging and deployment. This is recommended for Sequence developers only.
- Avoid using hardcoded URLs. You can use a reference table to hold the URLs and consume it in the process using an expression such as:

```
{act1}.query("reference").asQueryable().where(name = "some key name").first().key
```


(assuming you have an activity named act1, and you have added the reference table in a query named “reference”.)

Troubleshooting

The following case scenarios will help you troubleshoot deployment issues:

Case	Action
Web Services, WCF not working as expected.	Validate that end points are pointing to the right locations and that the credentials match the target environment.
Flowtime links are broken.	Validate reference table values and/or URL links within the workflow.
Cannot see changes that were implemented.	<ol style="list-style-type: none"> 1. Delete local IE cache or press Ctrl-F5. 2. Perform IIS Reset on the servers.
Flowtime not working. Administration works. In Flowtime, a date error is received.	Perform IIS Reset on the Flowtime server.
Wrong styles, colors, etc.	<ol style="list-style-type: none"> 1. Delete local IE cache or press Ctrl-F5. 2. Perform IIS Reset on the servers.
User(s) can see screens/forms that they are not supposed to see.	Validate permissions.