Paper 467-2013

SAS® Release Management and Version Control

John Heaton, Heritage Bank

ABSTRACT

Release Management is the process of managing multiple different software releases between environments in a controlled, auditable and repeatable process. This paper looks at the capabilities of the SAS 9.3 toolset to build an effective Release Management process within your organization to support the migration and maintenance of SAS code through your product lifecycle.

INTRODUCTION

The process of Release Management can be defined as follows:

> "Release management is a software engineering process intended to oversee the development, testing, deployment and support of software releases. The practice of release management combines the general business emphasis of traditional project management with a detailed technical knowledge of the systems development lifecycle (SDLC) and IT Infrastructure Library (ITIL) practices."

- **SearchSystemsChannel.com Definitions, Stephen J. Bigelow, 26 Jun 2008**

"Release management is the bridge that moves assets from development into production."

- The above definition is excerpted from Chapter 5, "Implementing System Management Services, Part 1: Deploying Service Support" of The Definitive Guide to Service-Oriented Systems Management by Dan Sullivan.

The process of managing and migrating code from environment to environment is normally an after-thought once the products have been purchased, the project is underway and code needs to be migrated from development to test and then from test to production.

The practical steps of Release Management are:

- The identification of new and changed solution components
- The assembly of a Release Package by creating a static copy of the changed solution software components.
- Backing up existing software components which will change as part of the Release.
- Migration of the release package objects using an auditable and repeatable process.

Many software vendors include some capability to support this process although it is normally not very well documented, communicated or consistent across products. Release management features do not sell software licenses and are rarely included in evaluations; however the features greatly influence the risk of the product implementation and the quality of the solution.

SETTING UP THE BASICS

Before starting to setup the environment you will need the following software.

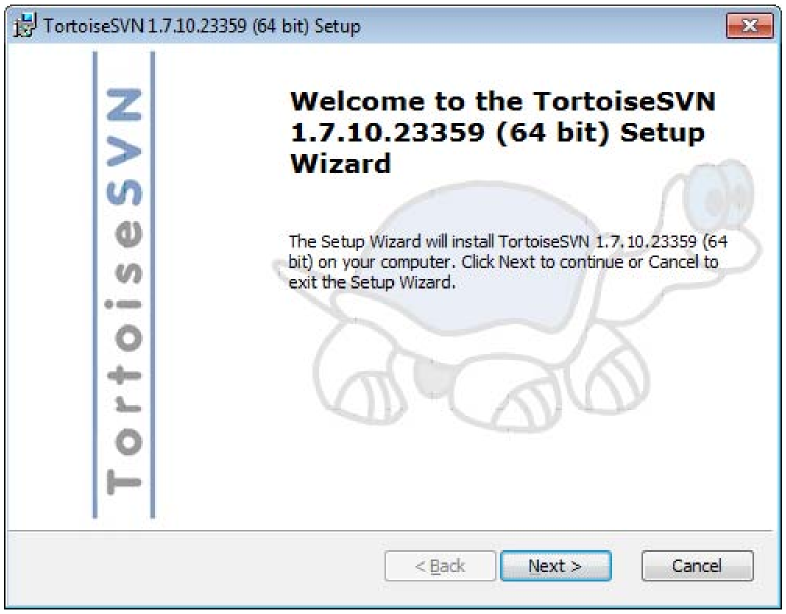
1. A Subversion (SVN) Server – for this you can either use
   a. Collabnet Subversion Edge – http://www.collabnet.com
   b. Visual SVN – http://www.visualsvn.com
2. TortoiseSVN – a Windows add for Subversion - http://tortoisesvn.net/downloads.html

For this paper I have used Collabnet Subversion Edge as the server and TortoiseSVN for the client tool.

SUBVERSION

TORTOISESVN INSTALLATION

TortoiseSVN allows you to interact with Subversion from Windows Explorer. This allows you to check files in and out from version control and is necessary as most SAS® tools do not provide native Subversion (version control) support. Once you have downloaded the relevant version of TortoiseSVN for Windows you can double click on the install package.

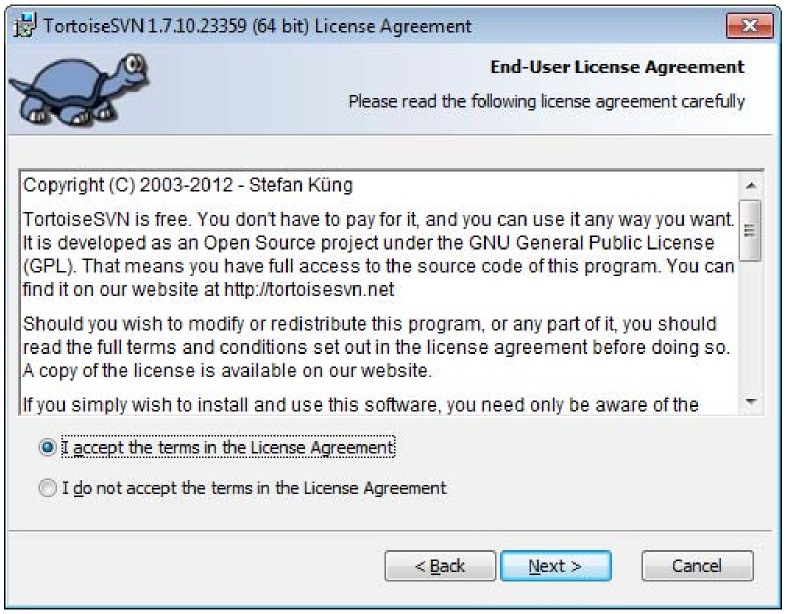


Welcome to the TortoiseSVN 1.7.10.23359 (64 bit) Setup Wizard

The Setup Wizard will install TortoiseSVN 1.7.10.23359 (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Figure 1. TortoiseSVN Install Wizard

Click Next to continue.



End-User License Agreement

Please read the following license agreement carefully

Copyright (C) 2003-2012 - Stefan Küng

TortoiseSVN is free. You don't have to pay for it, and you can use it any way you want. It is developed as an Open Source project under the GNU General Public License (GPL). That means you have full access to the source code of this program. You can find it on our website at http://tortoisesvn.net

Should you wish to modify or redistribute this program, or any part of it, you should read the full terms and conditions set out in the license agreement before doing so. A copy of the license is available on our website.

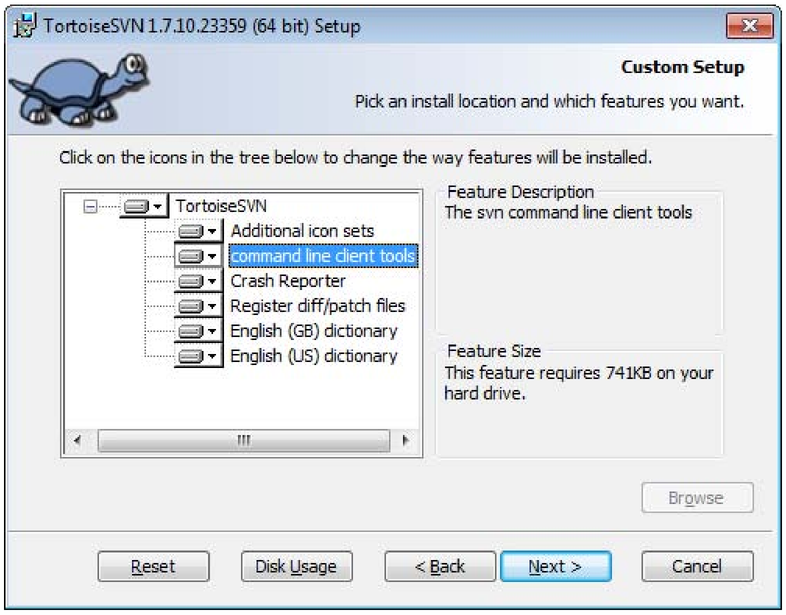
If you simply wish to install and use this software, you need only be aware of the

[✓ I accept the terms in the License Agreement]

I do not accept the terms in the License Agreement

Figure 2. TortoiseSVN License

Select the option *I accept the terms in the License Agreement* and Click Next.


TortoiseSVN 1.7.10.23359 (64 bit) Setup

Custom Setup

Pick an install location and which features you want.

Click on the icons in the tree below to change the way features will be installed.

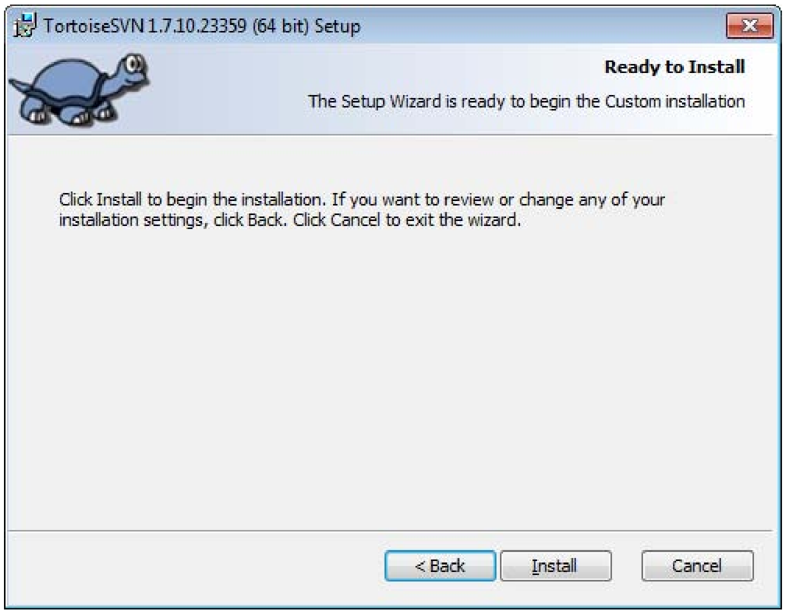
- TortoiseSVN
  - Additional icon sets
  - command line client tools
  - Crash Reporter
  - Register diff/patch files
  - English (GB) dictionary
  - English (US) dictionary

Feature Description
The svn command line client tools

Feature Size
This feature requires 741KB on your hard drive.

Figure 3. TortoiseSVN Command Line Tools

Select command line client tools and click Next. Ensure this is installed else SAS® DI Studio will not work with Subversion as it requires the command line tools.


TortoiseSVN 1.7.10.23359 (64 bit) Setup

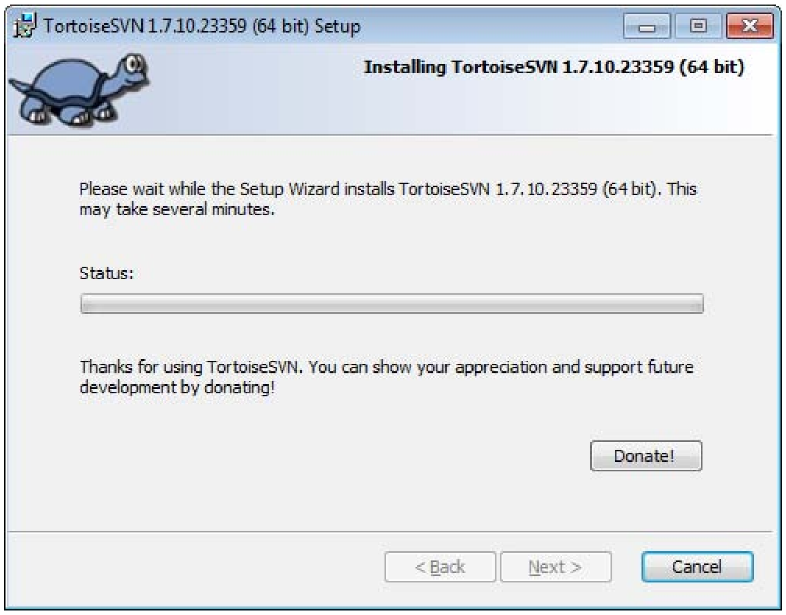
Ready to Install

The Setup Wizard is ready to begin the Custom installation

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Figure 4~~3~~. TortoiseSVN Install

Select Install.


TortoiseSVN 1.7.10.23359 (64 bit) Setup
Installing TortoiseSVN 1.7.10.23359 (64 bit)

Please wait while the Setup Wizard installs TortoiseSVN 1.7.10.23359 (64 bit). This may take several minutes.

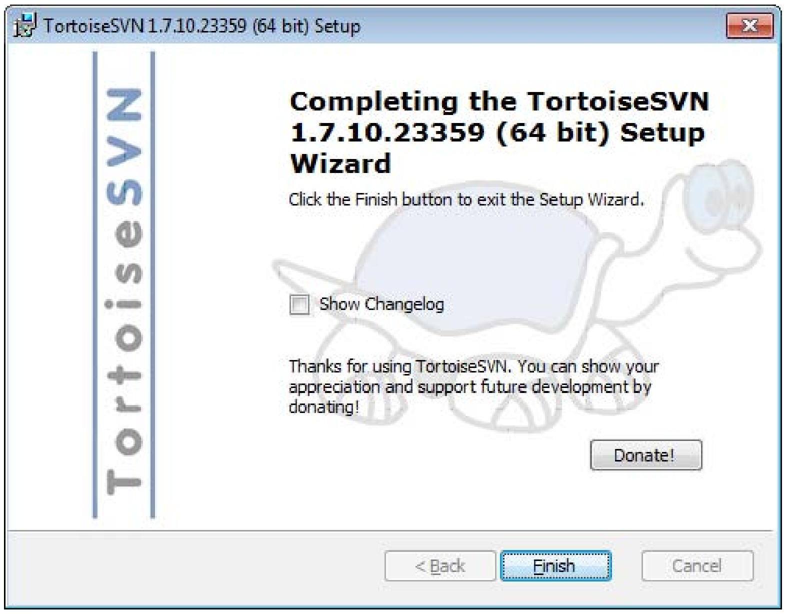
Status:

Thanks for using TortoiseSVN. You can show your appreciation and support future development by donating!

Donate!
< Back
Next >
Cancel

Figure 5. TortoiseSVN Installation Progress

The install normally takes several minutes. The progress can be monitored if necessary.



Completing the TortoiseSVN 1.7.10.23359 (64 bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

Show Changelog

Thanks for using TortoiseSVN. You can show your appreciation and support future development by donating!

Figure 6. TortoiseSVN Installation Completed

Click Finish. The installation is completed and TortoiseSVN is installed. The program can be invoked from the right mouse button when in Windows Explorer.

COLLABNET SUBVERSION EDGE INSTALLATION

Download the correct installation package and once complete double click to begin the install.

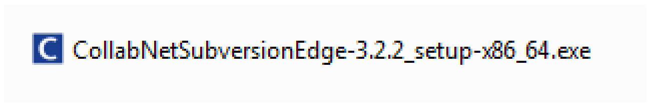
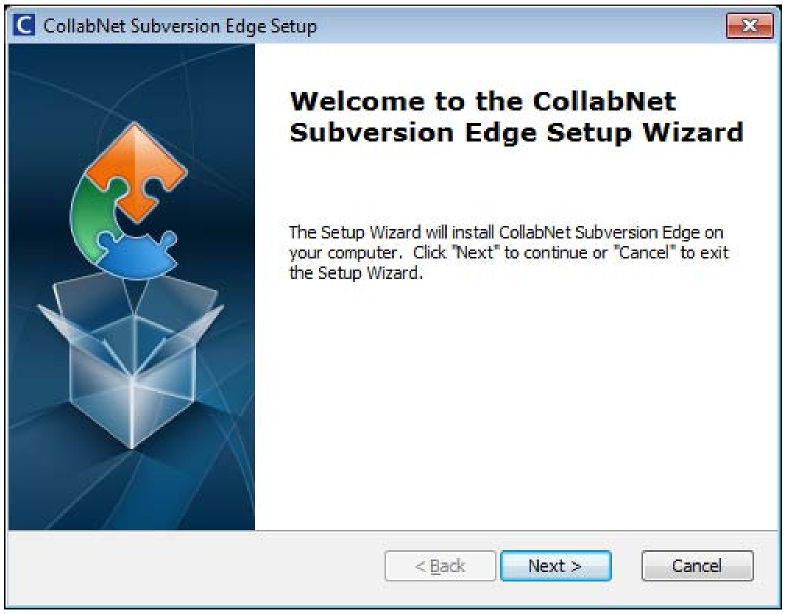


Figure 7. Collabnet Subversion Edge Installation Package



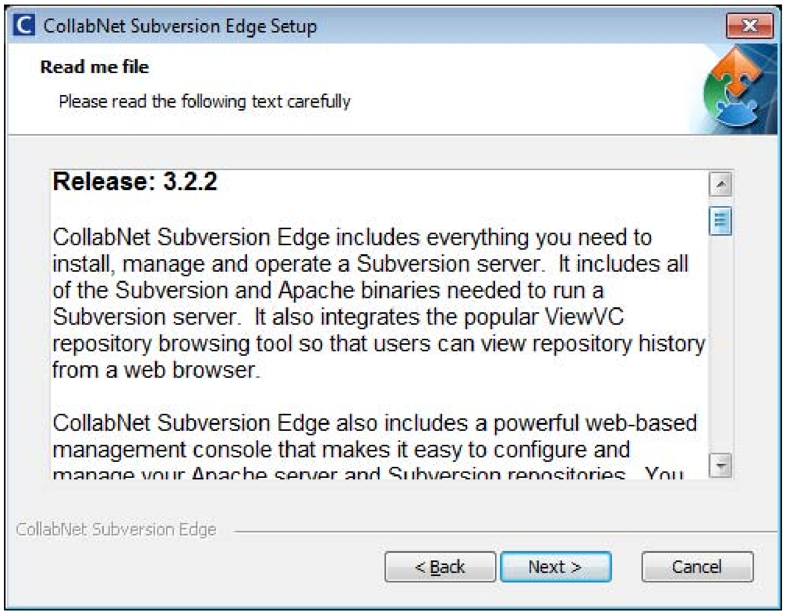
CollabNet Subversion Edge Setup

Welcome to the CollabNet Subversion Edge Setup Wizard

The Setup Wizard will install CollabNet Subversion Edge on your computer. Click "Next" to continue or "Cancel" to exit the Setup Wizard.

Figure 8. Collabnet Subversion Edge Install Wizard

Click Next to continue the installation.



CollabNet Subversion Edge Setup

Read me file

Please read the following text carefully

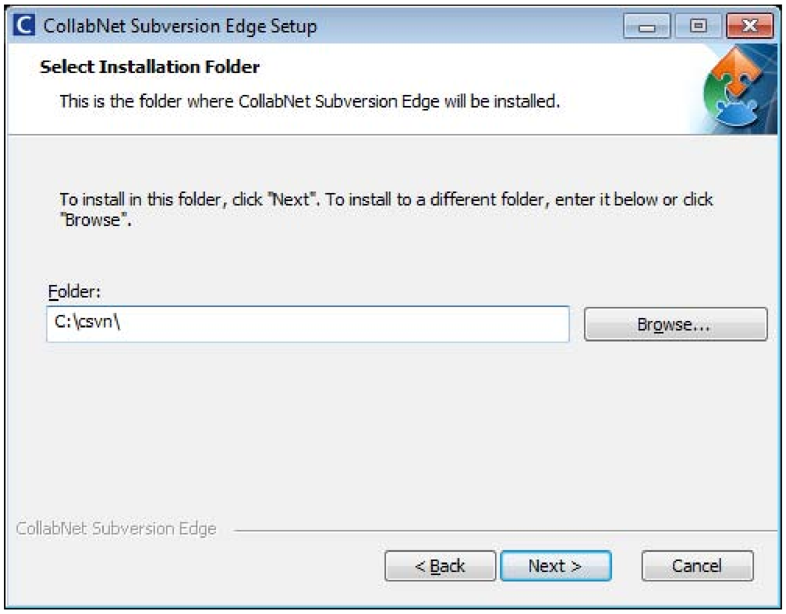
Release: 3.2.2

CollabNet Subversion Edge includes everything you need to install, manage and operate a Subversion server. It includes all of the Subversion and Apache binaries needed to run a Subversion server. It also integrates the popular ViewVC repository browsing tool so that users can view repository history from a web browser.

CollabNet Subversion Edge also includes a powerful web-based management console that makes it easy to configure and manage your Apache server and Subversion repositories. You

Figure 9. Collabnet Subversion Edge Readme

Click Next to continue the installation.



Select Installation Folder

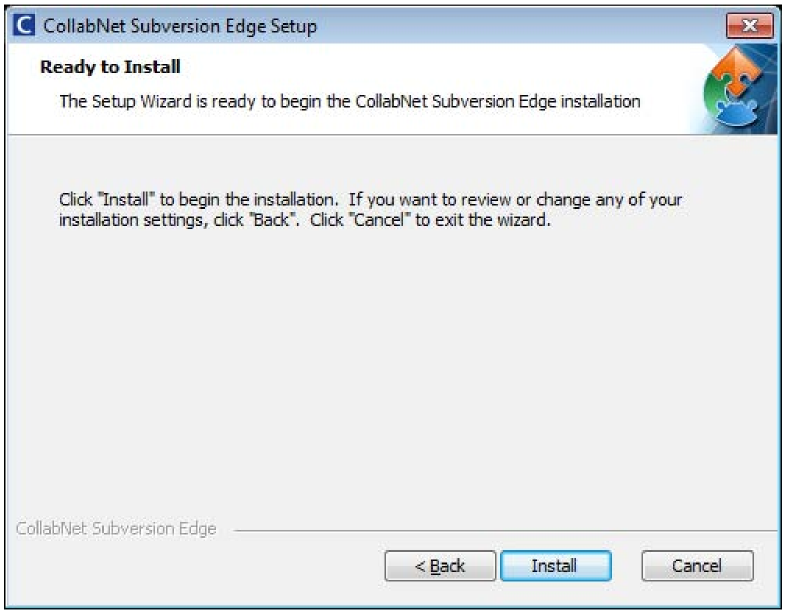
This is the folder where CollabNet Subversion Edge will be installed.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\csvn\

Figure 10. Collabnet Subversion Edge Installation directory

Verify the directory and click Next to continue the installation.



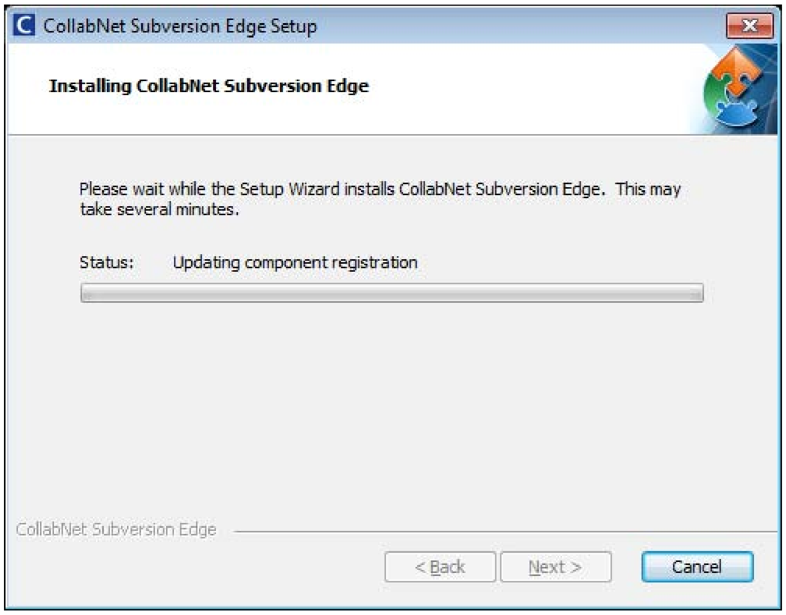
Ready to Install

The Setup Wizard is ready to begin the CollabNet Subversion Edge installation

Click "Install" to begin the installation. If you want to review or change any of your installation settings, click "Back". Click "Cancel" to exit the wizard.

Figure 11. Collabnet Subversion Edge Installation

Click Install to continue the installation.


CollabNet Subversion Edge Setup

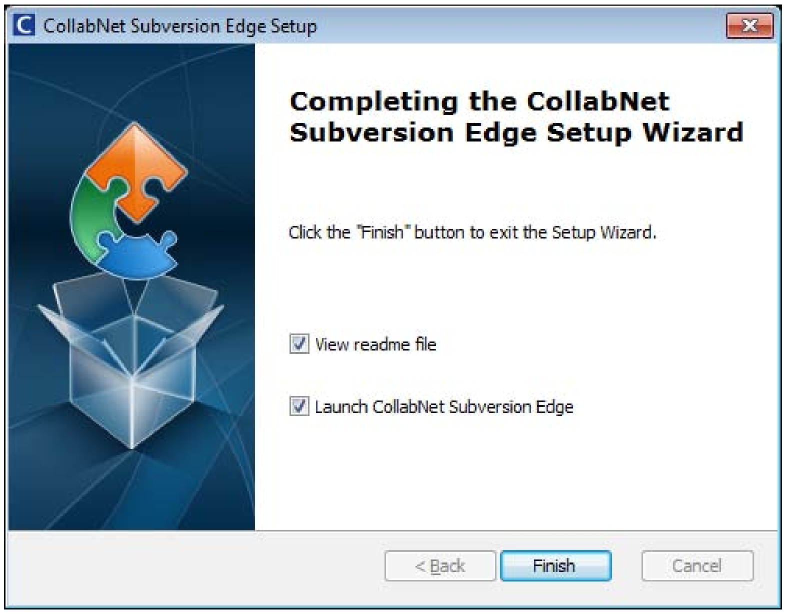
Installing CollabNet Subversion Edge

Please wait while the Setup Wizard installs CollabNet Subversion Edge. This may take several minutes.

Status: Updating component registration

Figure 12. Collabnet Subversion Edge Installation Status

Click Next to continue the installation.


CollabNet Subversion Edge Setup

Completing the CollabNet Subversion Edge Setup Wizard

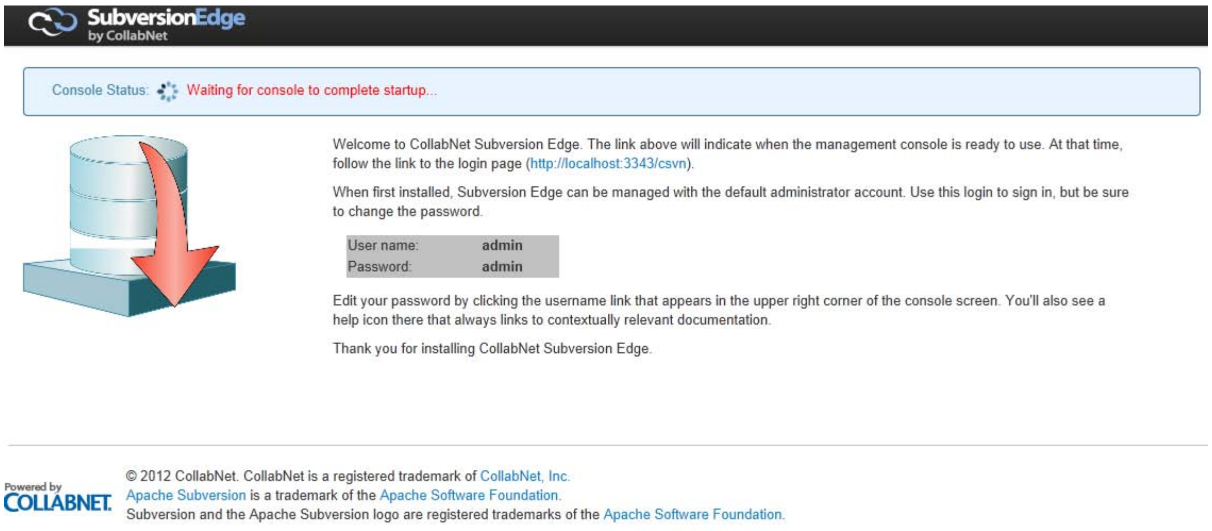
Click the "Finish" button to exit the Setup Wizard.

[✓] View readme file

[✓] Launch CollabNet Subversion Edge

Figure 13. Collabnet Subversion Edge Installation Completed

Click Finish to launch the Collabnet Subversion Edge.



Console Status: Waiting for console to complete startup...

Welcome to CollabNet Subversion Edge. The link above will indicate when the management console is ready to use. At that time, follow the link to the login page (http://localhost:3343/csvn).

When first installed, Subversion Edge can be managed with the default administrator account. Use this login to sign in, but be sure to change the password.

User name:	admin
Password:	admin

Edit your password by clicking the username link that appears in the upper right corner of the console screen. You'll also see a help icon there that always links to contextually relevant documentation.

Thank you for installing CollabNet Subversion Edge.

Powered by COLLABNET

© 2012 CollabNet. CollabNet is a registered trademark of CollabNet, Inc.
Apache Subversion is a trademark of the Apache Software Foundation.
Subversion and the Apache Subversion logo are registered trademarks of the Apache Software Foundation.

Figure 14. Collabnet Subversion Edge Startup

Once started up click on the link to the login page.

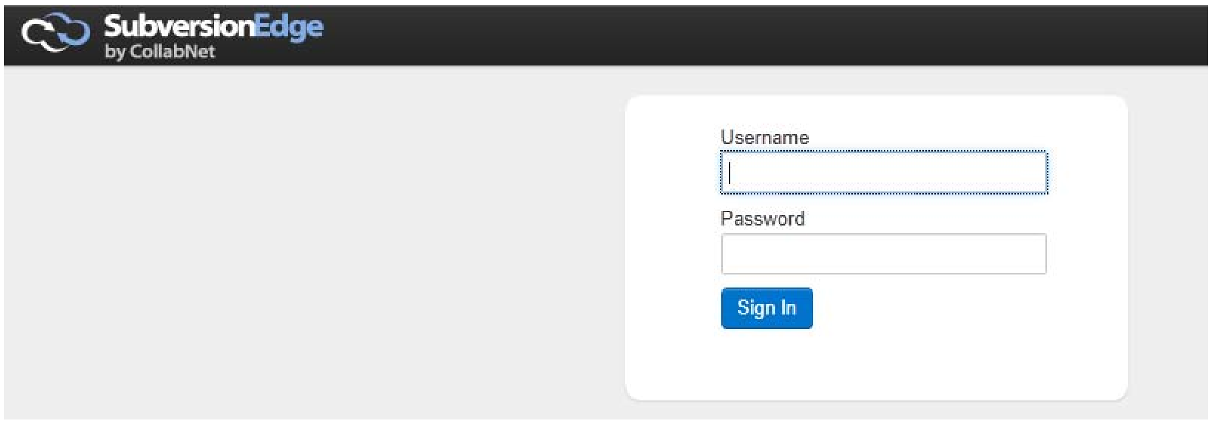
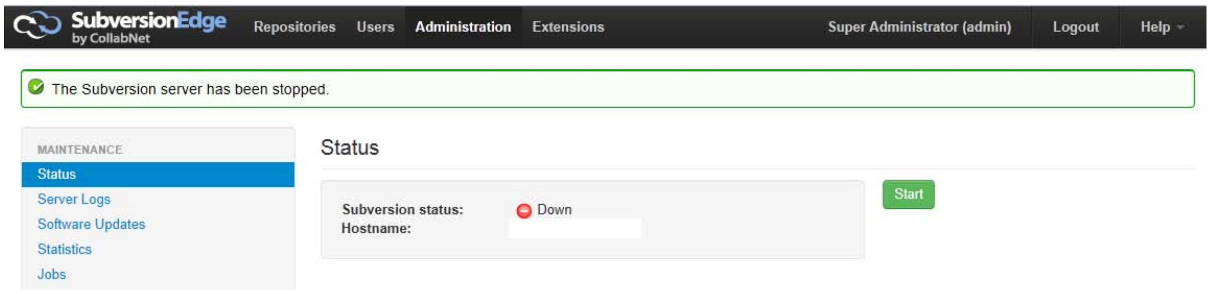


Figure 15. Collabnet Subversion Edge Login

Enter the admin username and password.



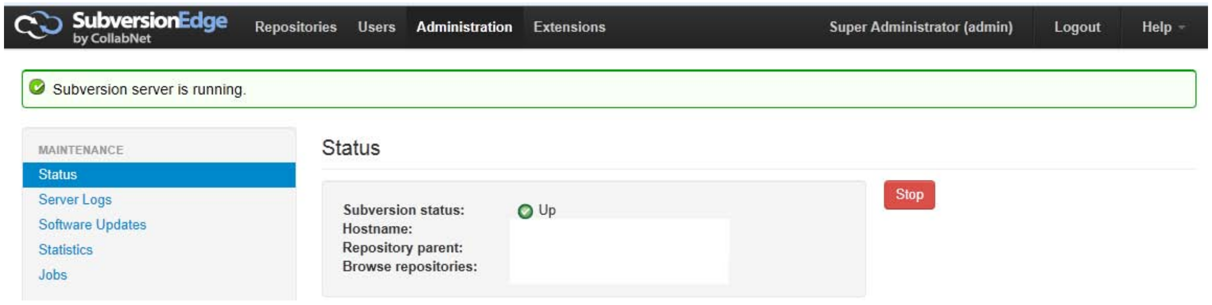
The Subversion server has been stopped.

Status

Subversion status: Down
Hostname:

Figure 16. Collabnet Subversion Edge Start Subversion

Click Start to startup Collabnet Subversion Edge.



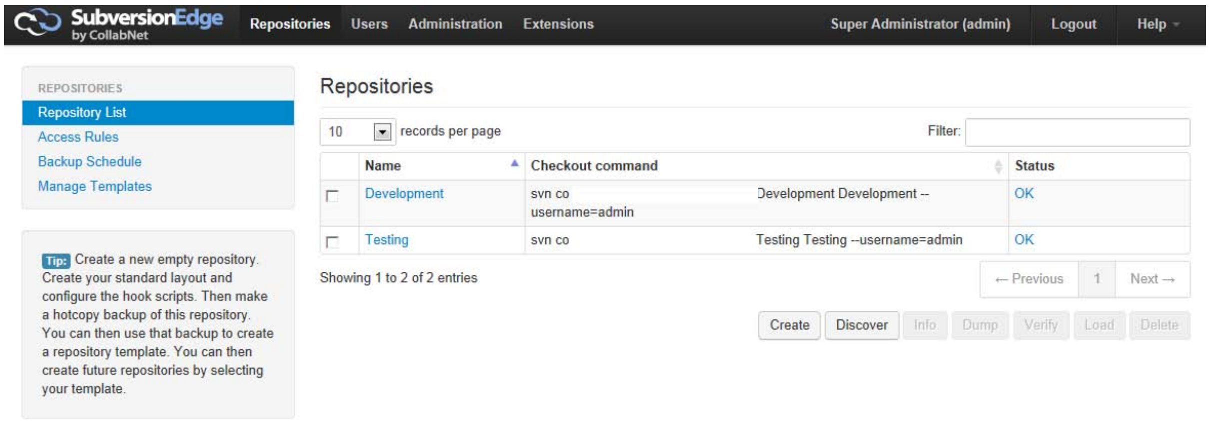
Subversion server is running.

Status

Subversion status: Up
Hostname:
Repository parent:
Browse repositories:

Figure 17. Collabnet Subversion Edge Status

Validate status of Collabnet Subversion Edge.



Repositories

10 records per page Filter:

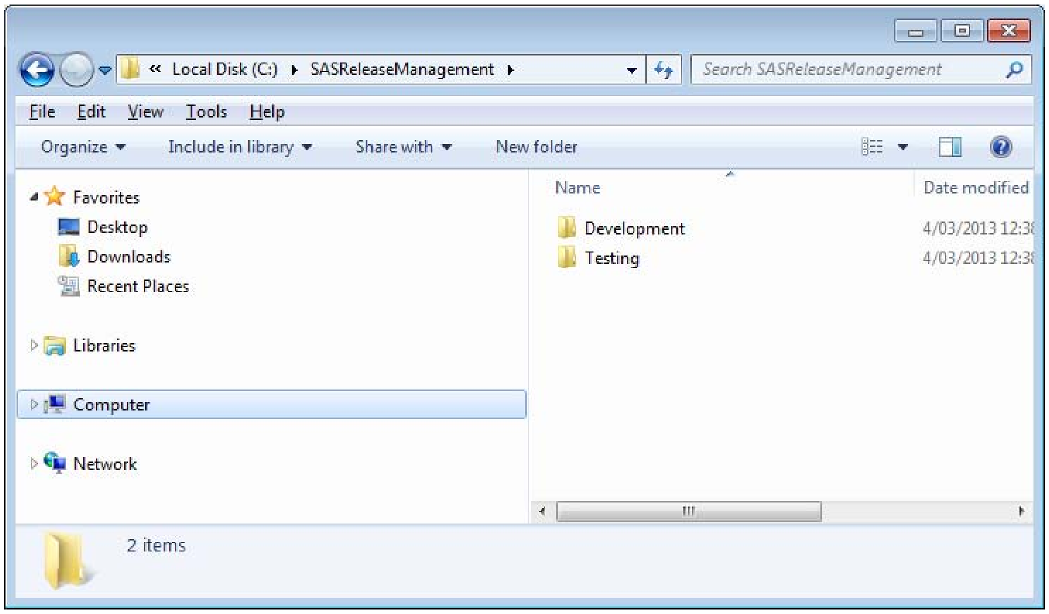
	Name	Checkout command	Status
	Development	svn co Development Development --username=admin	OK
	Testing	svn co Testing Testing --username=admin	OK

Showing 1 to 2 of 2 entries ← Previous 1 Next →

Tip: Create a new empty repository. Create your standard layout and configure the hook scripts. Then make a hotcopy backup of this repository. You can then use that backup to create a repository template. You can then create future repositories by selecting your template.

Figure 18. Collabnet Subversion Edge Create Repositories

Click on Repositories and then create one for each source and target environment. If you plan on using DI Studio with Subversion, create an additional repository for SAS® DI Studio.

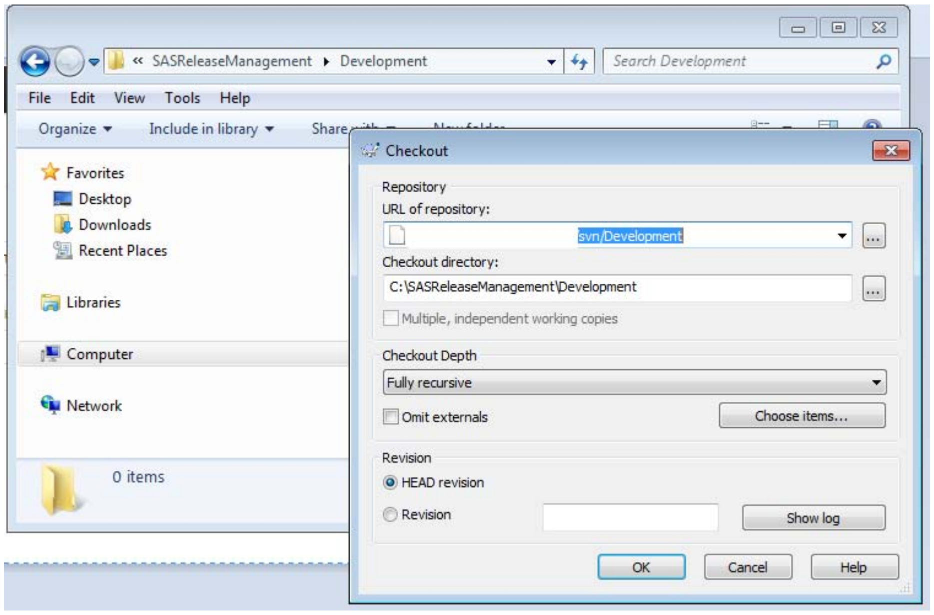


Name	Date modified
Development	4/03/2013 12:3
Testing	4/03/2013 12:3

2 items

Figure 19. Folder Structures

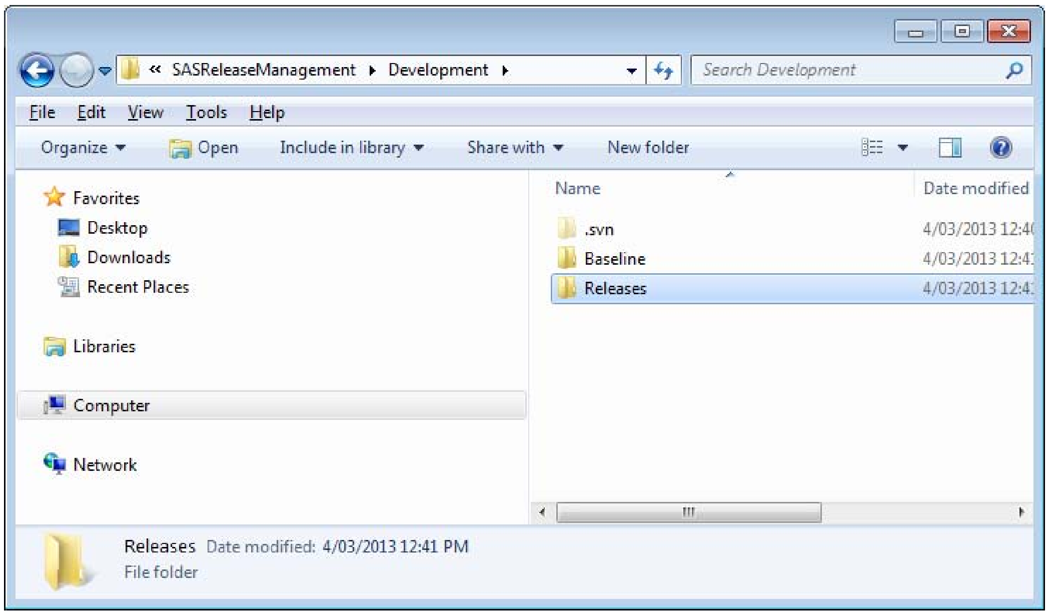
In Windows Explorer create a SASReleaseManagement Folder with a separate folder for each of the repositories defined above.



Repository
URL of repository: svn/Development
Checkout directory: C:\SASReleaseManagement\Development
☐ Multiple, independent working copies
Checkout Depth: Fully recursive
☐ Omit externals
Revision
● HEAD revision
○ Revision

Figure 20. Subversion Checkout

Navigate to one of the folders, right click and select Checkout.   Set the URL of repository to the associated environment and ensure the directory maps to the repository. Click OK. If requested enter the admin username and password.


SASReleaseManagement ▸ Development ▸
Search Development
File Edit View Tools Help
Organize ▾ Open Include in library ▾ Share with ▾ New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Computer
Network

Name	Date modified
.svn	4/03/2013 12:4
Baseline	4/03/2013 12:4
Releases	4/03/2013 12:4

Releases Date modified: 4/03/2013 12:41 PM
File folder

Figure 21. Subversion Folders

Under the environment create two folders,

- Baseline – will store all the complete code releases and backups
- Releases – will store incremental releases

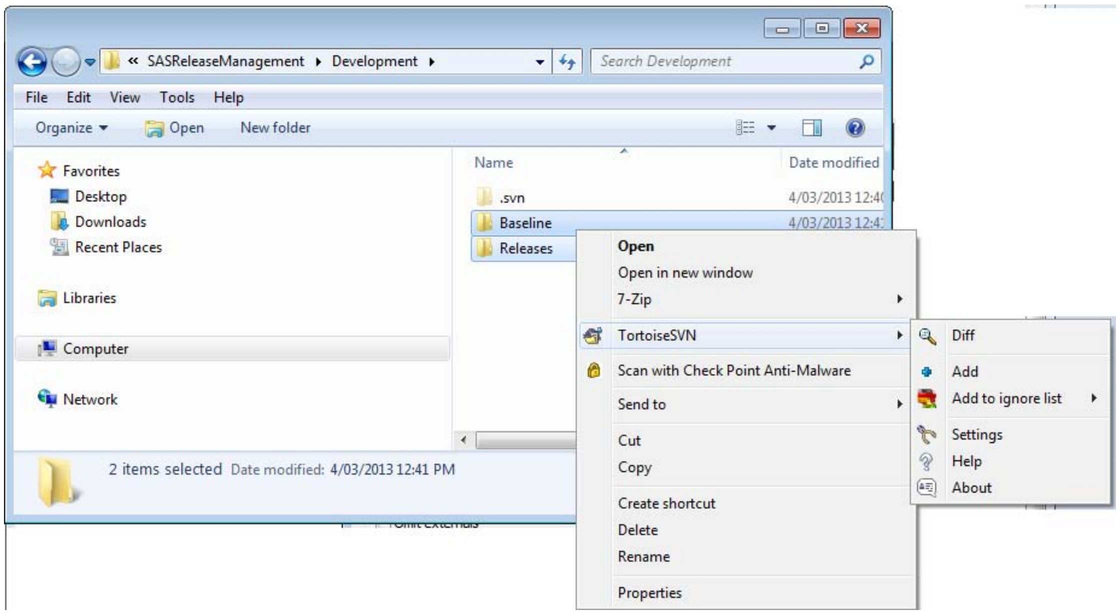


Figure 22. Subversion Add

Select both folders, right click and select TortoiseSVN -> Add.

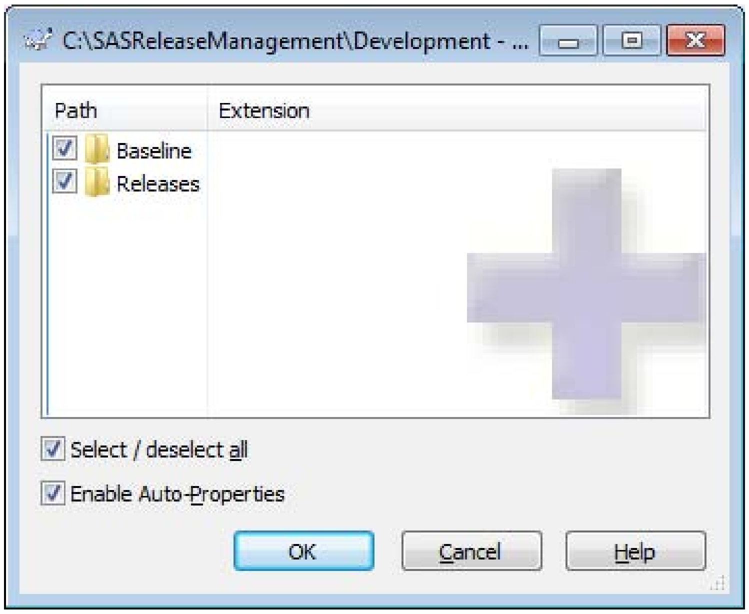
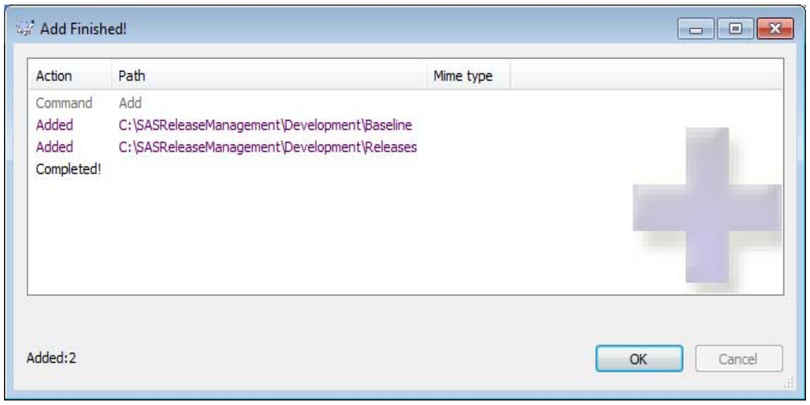


Figure 23~~22~~. Subversion Add

Verify the folders and select Add.



Add Finished!

Action	Path	Mime type
Command	Add	
Added	C:\SASReleaseManagement\Development\Baseline	
Added	C:\SASReleaseManagement\Development\Releases	
Completed!		

Added:2

Figure 24~~22~~. Subversion Add Success

Click Ok.

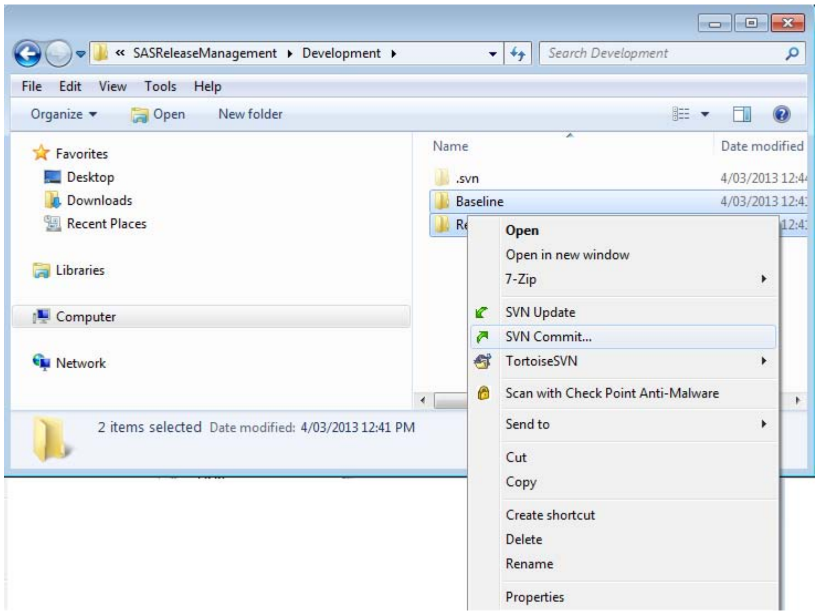


Figure 25. Subversion Commit

Right click and select SVNCommit

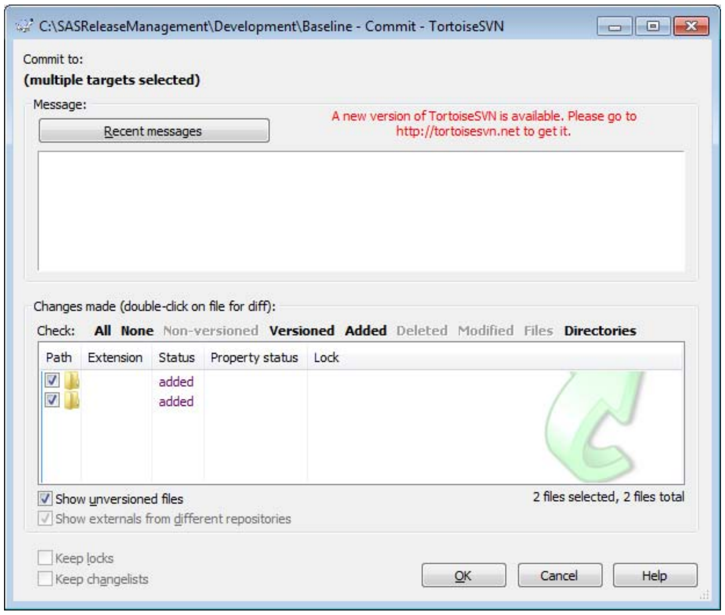
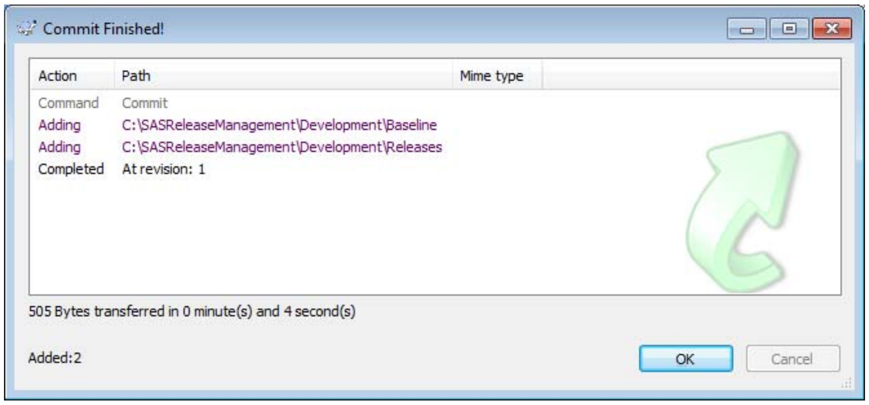

C:\SASReleaseManagement\Development\Baseline - Commit - TortoiseSVN
Commit to:
(multiple targets selected)
Message:
Recent messages
A new version of TortoiseSVN is available. Please go to http://tortoisesvn.net to get it.
Changes made (double-click on file for diff):
Check: All None Non-versioned Versioned Added Deleted Modified Files Directories
Path Extension Status Property status Lock
added
added
Show unversioned files
2 files selected, 2 files total
Show externals from different repositories
Keep locks
Keep changelists
OK
Cancel
Help

Figure 26. Subversion Commit

Verify object and select Ok



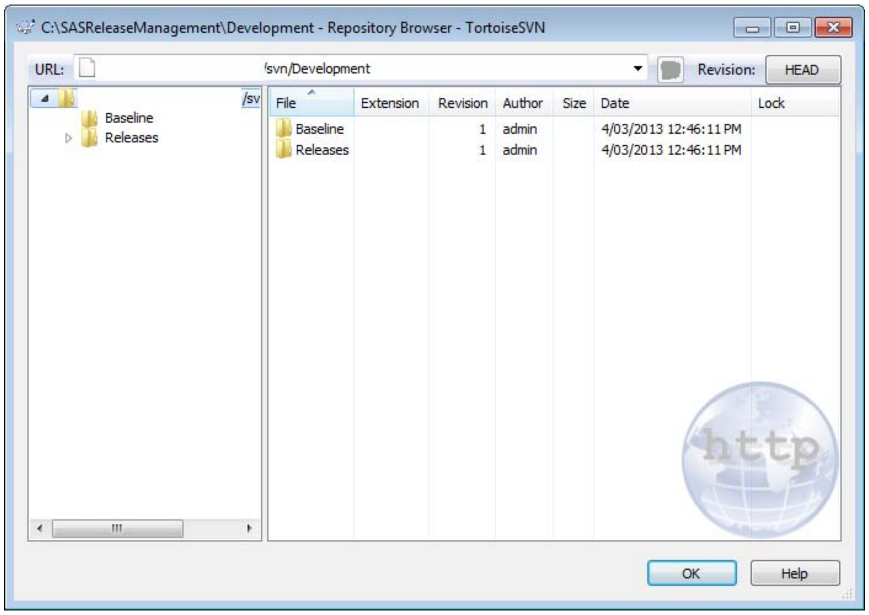
Action	Path	Mime type
Command	Commit	
Adding	C:\SASReleaseManagement\Development\Baseline	
Adding	C:\SASReleaseManagement\Development\Releases	
Completed	At revision: 1	

505 Bytes transferred in 0 minute(s) and 4 second(s)

Added:2

Figure 27. Subversion Commit Complete

Click Ok.


C:\SASReleaseManagement\Development - Repository Browser - TortoiseSVN
URL:
/svn/Development
Revision:
HEAD
Baseline
Releases
OK
Help

File	Extension	Revision	Author	Size	Date	Lock
Baseline		1	admin		4/03/2013 12:46:11 PM	
Releases		1	admin		4/03/2013 12:46:11 PM	

Figure 28. TortoiseSVN Repo Browser

Right click and select TortoiseSVN -> Repo Browser in Windows Explorer. Enter the URL for the machine and repository you have just added the folders to. Enter the username and password of admin when prompted. You will now notice there is two folders under the environment which mirror the folders on your machine.

BATCH EXPORT AND EXPORT TOOLS

To enable the Batch Import and Export Tools there are two configuration files which need to be configured.   Detailed instruction can be found in the SAS® 9.3 Intelligence Platform: System Administration Guide, Using the Promotion Tools (link provided in the references section).

In the .ini files for the batch export and import tools, you can set global options to override the default handling of certain object types. To
set these options:

1. Go to the directory SAS-installation-directory/SASPlatformObjectFramework/9.3 and locate the files ExportPackage.ini and ImportPackage.ini on your server.

2. Make a backup copy of the .ini files ExportPackage.ini and ImportPackage.ini.

3. Open the .ini file in a text editor, and add a new Java argument for each option that you want to specify.

```
JavaArgs_10=-Dsas.promotion.stp.includesourcecode=true
JavaArgs_11=-Dsas.promotion.columns.excludenewcolumns=false
JavaArgs_12=-Dsas.promotion.columns.deletetargetcolumns=false;
```

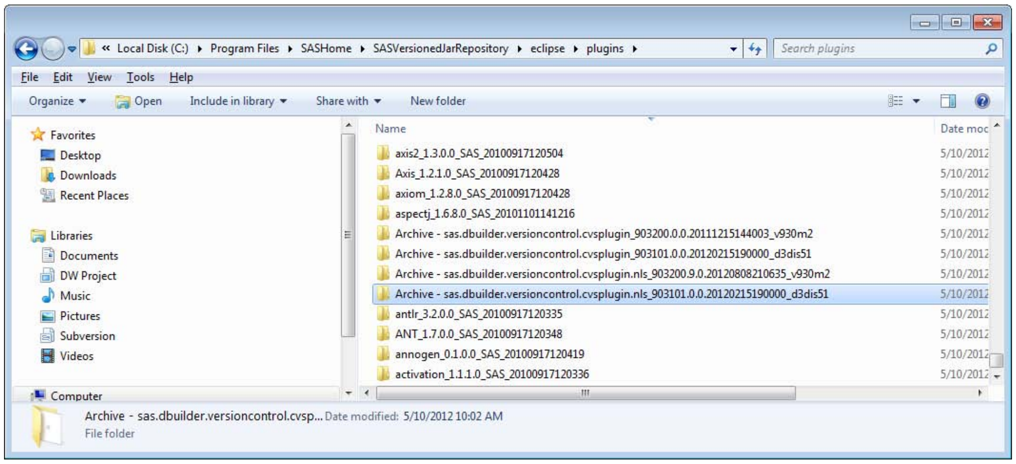
These options are the default settings and you need to validate they will work in your environment.

SAS® DI STUDIO

Navigate to the SAS Home directory for your SAS® DI Studio installation on your client machine and locate the plugins for CVS version control. Note the version numbers may be different depending on the patch level your are on.

- sas.dbuilder.versioncontrol.cvsplugin_903200.0.0.20111215144003_v930m2
- sas.dbuilder.versioncontrol.cvsplugin.nls_903200.9.0.20120808210635_v930m2
- sas.dbuilder.versioncontrol.cvsplugin_903101.0.0.20120215190000_d3dis51
- sas.dbuilder.versioncontrol.cvsplugin.nls_903101.0.0.20120215190000_d3dis51

Rename the directories as Archive – directory name. Notice you may have numerous directories. Be sure to rename them all.



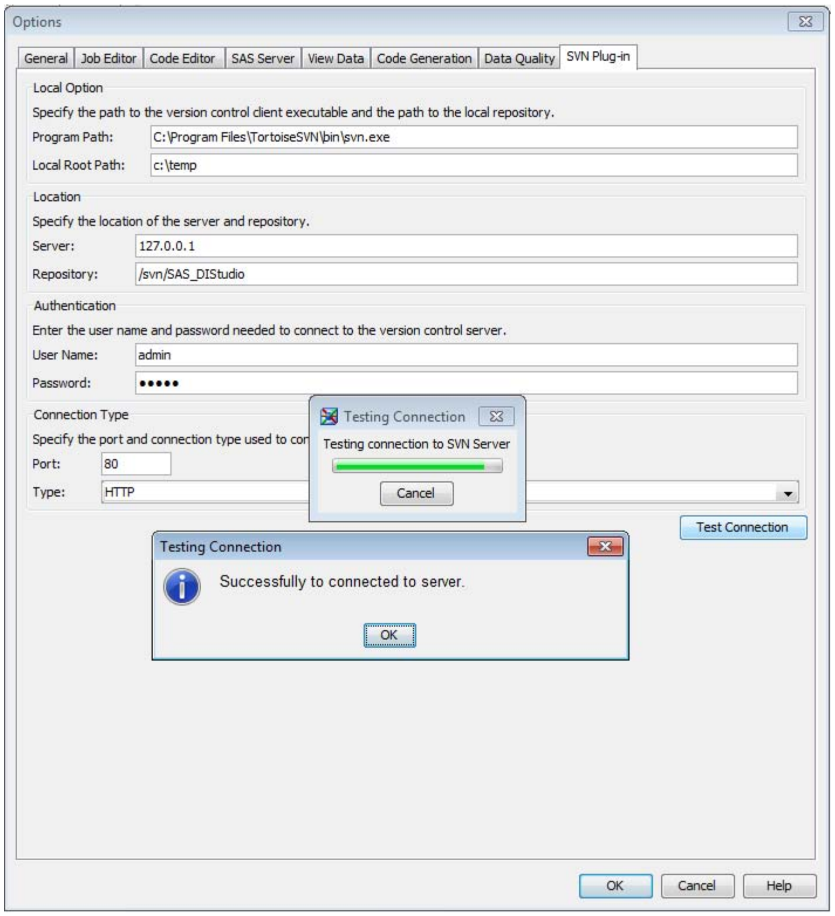
« Local Disk (C:) ▸ Program Files ▸ SASHome ▸ SASVersionedJarRepository ▸ eclipse ▸ plugins ▸

Name	Date modified
axis2_1.3.0.0_SAS_20100917120504	5/10/2012
Axis_1.2.1.0_SAS_20100917120428	5/10/2012
axiom_1.2.8.0_SAS_20100917120428	5/10/2012
aspectj_1.6.8.0_SAS_20101101141216	5/10/2012
Archive - sas.dbuilder.versioncontrol.cvsplugin_903200.0.0.20111215144003_v930m2	5/10/2012
Archive - sas.dbuilder.versioncontrol.cvsplugin_903101.0.0.20120215190000_d3dis51	5/10/2012
Archive - sas.dbuilder.versioncontrol.cvsplugin.nls_903200.9.0.20120808210635_v930m2	5/10/2012
Archive - sas.dbuilder.versioncontrol.cvsplugin.nls_903101.0.0.20120215190000_d3dis51	5/10/2012
antlr_3.2.0.0_SAS_20100917120335	5/10/2012
ANT_1.7.0.0_SAS_20100917120348	5/10/2012
annogen_0.1.0.0_SAS_20100917120419	5/10/2012
activation_1.1.1.0_SAS_20100917120336	5/10/2012

Archive - sas.dbuilder.versioncontrol.cvsp... Date modified: 5/10/2012 10:02 AM
File folder

Figure 29. CVS Plugins renamed

Open SAS® DI Studio and navigate to Tools -> Options -> SVN Plug-in.



Options — SVN Plug-in

Local Option
Specify the path to the version control client executable and the path to the local repository.
Program Path: C:\Program Files\TortoiseSVN\bin\svn.exe
Local Root Path: c:\temp

Location
Specify the location of the server and repository.
Server: 127.0.0.1
Repository: /svn/SAS_DIStudio

Authentication
Enter the user name and password needed to connect to the version control server.
User Name: admin
Password: •••••

Connection Type
Specify the port and connection type used to con
Port: 80
Type: HTTP

Testing Connection
Testing connection to SVN Server

Testing Connection
Successfully to connected to server.

Figure 30. Subversion Plugin Configuration

If the previous step was successful the CVS Plu-in tab will not be present and only the SVN Plug-in will be available.

Enter the details related to your Collabnet Subversion Edge Installation.

1. Program Path – this is the location of the svn.exe installed with TortoiseSVN above.
2. Local Root Path – temporary location for
3. Server – Collabnet Subversion Edge Server
4. Repository – the name of the repository for DI Studio – e.g. SAS_DIStudio
5. Username – admin
6. Password – admin
7. Port – 80
8. Type – HTTP

Click Test Connection and it should be successful.

USING VERSION CONTROL

Collaboration

Version control can be used for two primary purposes, collaboration and versioning. Version control used for collaboration is helpful for the sharing of information and maintaining and a most recent version of the artifact.

In this scenario you may have multiple people working on the same or different pieces of code. Instead of emailing information around, placing them on the shared drive or some other form of centralized solution the code is placed into version control. As the code is maintained the updated information can be replicated and shared amongst the collaborators. SAS products which can benefit from this would be

1. SAS® Enterprise Guide – projects and programs can be saved to Subversion folders and then added to the Subversion repository using TortoiseSVN. The process allows code to be ~~is~~ centrally stored, managed and updated.
2. SAS® DI Studio – before changes are made to objects they can be archived and stored in Subversion. These changes are then available to everyone if they are needed to be rolled back.
3. Custom Code – any custom code which is used in your project and stored as a text file, including macros can be stored in the same way.

Within the a SAS toolset there are minimal products which do not make use of the SAS® Metadata Server and therefore you need to consider if collaboration makes more sense using the SAS® Metadata Server or TortoiseSVN.

Version Control

For version control SAS® DI Studio is the only product which links into SVN natively. You have the capability of archiving most objects at either the object or the folder level. The ability to archive objects at the object level allows you to track changes at a granular level. Note this capability operates independently of Change Management.

In SAS® DI Studio when you create a job before a change you archive the job or object by right clicking and selecting Archive as a SAS Package.

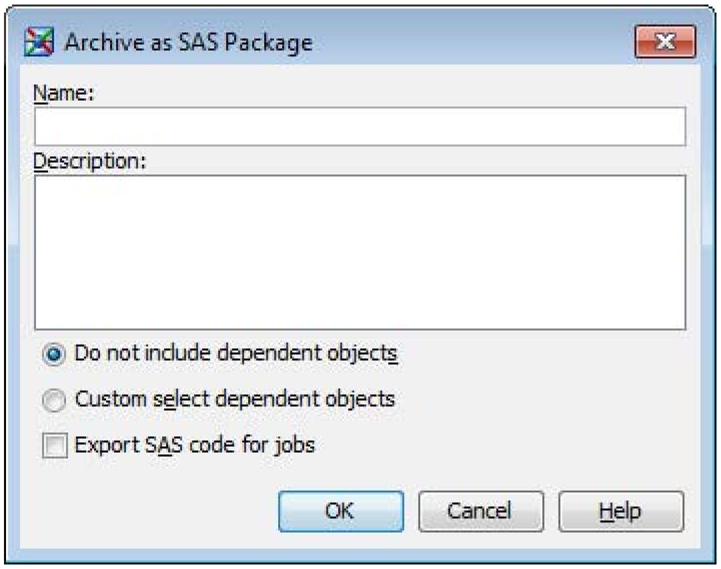
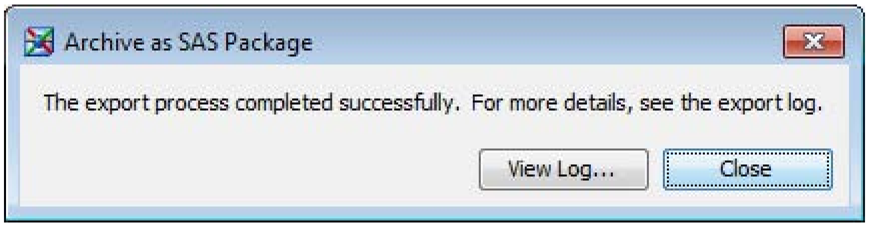


Figure 31. Archive as SAS Package

Supply a name and description and click OK.

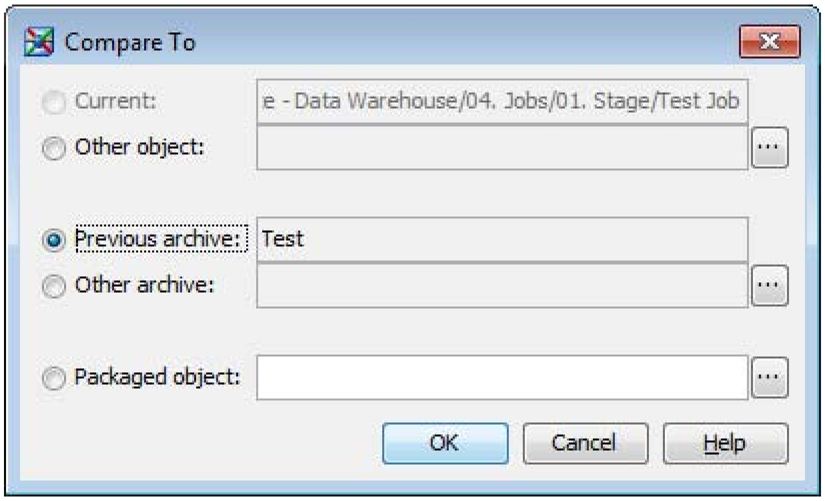


The export process completed successfully. For more details, see the export log.

Figure 32. Archive as SAS Package success

Click close or view log to see results.

When you right click on the object now you can chose compare.



Current: e - Data Warehouse/04. Jobs/01. Stage/Test Job

Other object:

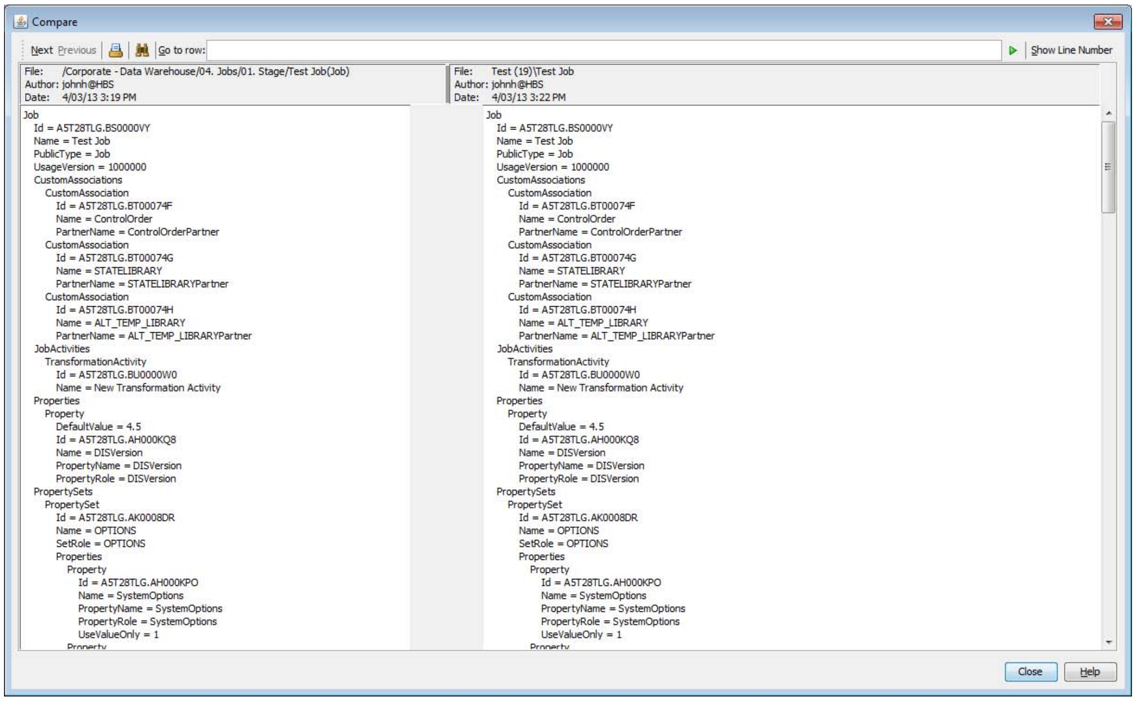
Previous archive: Test

Other archive:

Packaged object:

Figure 33. Compare a job

Select a previous archive and select OK.

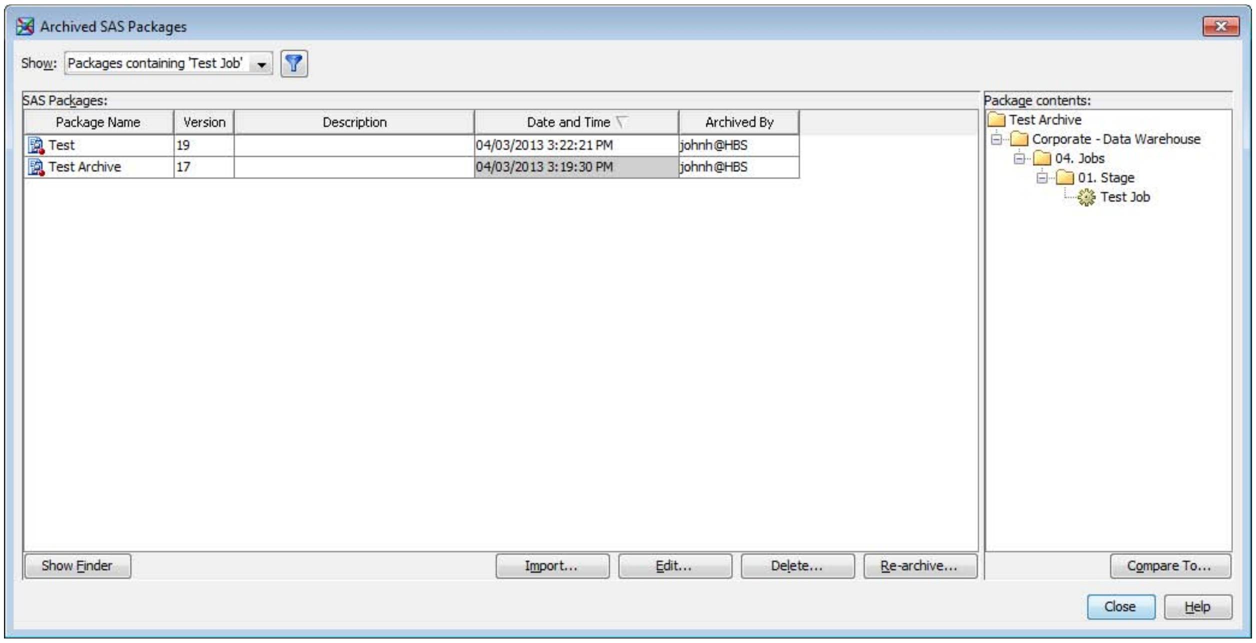


```
File:    /Corporate - Data Warehouse/04. Jobs/01. Stage/Test Job(Job)
Author: johnh@HBS
Date:    4/03/13 3:19 PM
Job
  Id = A5T28TLG.BS0000VY
  Name = Test Job
  PublicType = Job
  UsageVersion = 1000000
  CustomAssociations
    CustomAssociation
      Id = A5T28TLG.BT00074F
      Name = ControlOrder
      PartnerName = ControlOrderPartner
    CustomAssociation
      Id = A5T28TLG.BT00074G
      Name = STATELIBRARY
      PartnerName = STATELIBRARYPartner
    CustomAssociation
      Id = A5T28TLG.BT00074H
      Name = ALT_TEMP_LIBRARY
      PartnerName = ALT_TEMP_LIBRARYPartner
  JobActivities
    TransformationActivity
      Id = A5T28TLG.BU0000W0
      Name = New Transformation Activity
  Properties
    Property
      DefaultValue = 4.5
      Id = A5T28TLG.AH000KQ8
      Name = DISVersion
      PropertyName = DISVersion
      PropertyRole = DISVersion
  PropertySets
    PropertySet
      Id = A5T28TLG.AK0008DR
      Name = OPTIONS
      SetRole = OPTIONS
      Properties
        Property
          Id = A5T28TLG.AH000KPO
          Name = SystemOptions
          PropertyName = SystemOptions
          PropertyRole = SystemOptions
          UseValueOnly = 1
```

```
File:    Test (19)\Test Job
Author: johnh@HBS
Date:    4/03/13 3:22 PM
Job
  Id = A5T28TLG.BS0000VY
  Name = Test Job
  PublicType = Job
  UsageVersion = 1000000
  CustomAssociations
    CustomAssociation
      Id = A5T28TLG.BT00074F
      Name = ControlOrder
      PartnerName = ControlOrderPartner
    CustomAssociation
      Id = A5T28TLG.BT00074G
      Name = STATELIBRARY
      PartnerName = STATELIBRARYPartner
    CustomAssociation
      Id = A5T28TLG.BT00074H
      Name = ALT_TEMP_LIBRARY
      PartnerName = ALT_TEMP_LIBRARYPartner
  JobActivities
    TransformationActivity
      Id = A5T28TLG.BU0000W0
      Name = New Transformation Activity
  Properties
    Property
      DefaultValue = 4.5
      Id = A5T28TLG.AH000KQ8
      Name = DISVersion
      PropertyName = DISVersion
      PropertyRole = DISVersion
  PropertySets
    PropertySet
      Id = A5T28TLG.AK0008DR
      Name = OPTIONS
      SetRole = OPTIONS
      Properties
        Property
          Id = A5T28TLG.AH000KPO
          Name = SystemOptions
          PropertyName = SystemOptions
          PropertyRole = SystemOptions
          UseValueOnly = 1
```

Figure 34. Job diff

Review the comparison. Should you notice any different which is incorrect you can Close. Right click on the job and select Import – Archived SAS Package. These are technical compares and do not provide any facility for a logical compare of the code. For this reason there is limited benefit to detect changes.



Show: Packages containing 'Test Job'

SAS Packages:

Package Name	Version	Description	Date and Time	Archived By
Test	19		04/03/2013 3:22:21 PM	johnh@HBS
Test Archive	17		04/03/2013 3:19:30 PM	johnh@HBS

Package contents:

```
Test Archive
  Corporate - Data Warehouse
    04. Jobs
      01. Stage
        Test Job
```

Figure 35. Archived packages

In this window you can review the different archives for the object and chose the appropriate to import, delete or re – archive. You can also compare the archives to the current. This is where you can roll code back from a change and then re-deploy if necessary.

RELEASE MANAGEMENT PROCESS – INITIAL RELEASE

PROCESS

Once the initial development process is completed and all obsolete objects have been deleted then the code is ready to migrate from your source environment to your target environment. Figure 1 below outlines the process as per the SAS® documentation. Note the environments should be as similar as possible, thus all the same groups, libraries, folders and configurations should exists within the source and target environments before code is migrated.

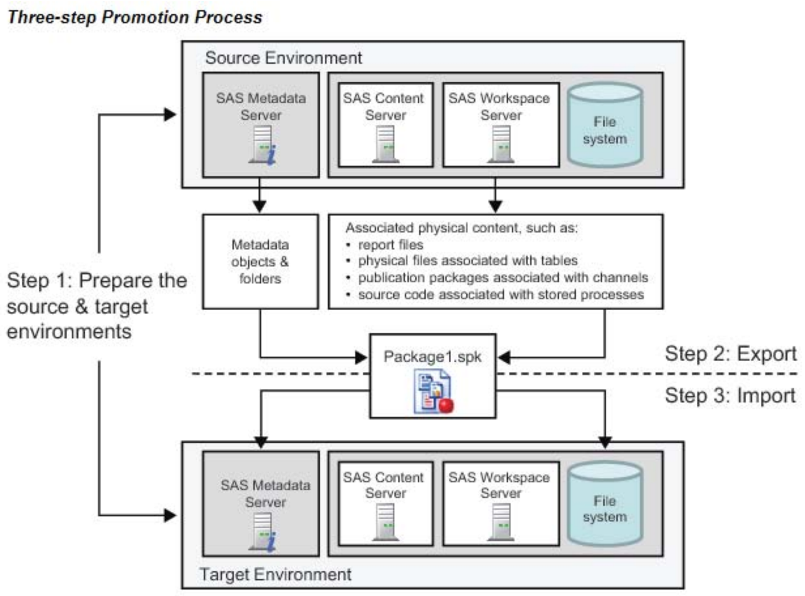

Three-step Promotion Process
Source Environment
SAS Metadata Server
SAS Content Server
SAS Workspace Server
File system
Metadata objects & folders
Associated physical content, such as:
• report files
• physical files associated with tables
• publication packages associated with channels
• source code associated with stored processes
Step 1: Prepare the source & target environments
Package1.spk
Step 2: Export
Step 3: Import
SAS Metadata Server
SAS Content Server
SAS Workspace Server
File system
Target Environment

Figure 36. SAS® Release Process

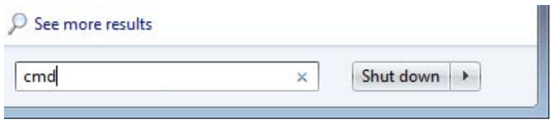
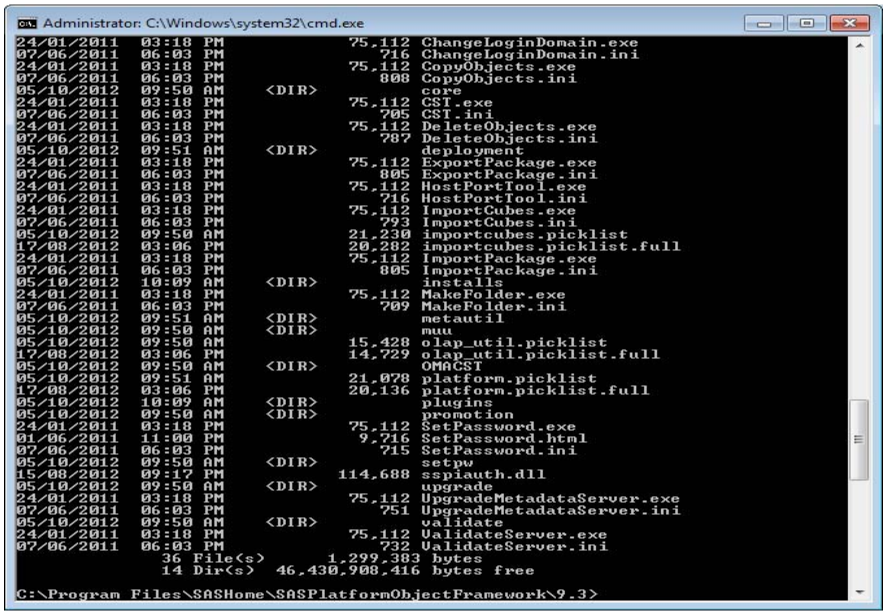


Figure ~~2~~37. Command Line access

On your client you can open a command window from Start and type in cmd then press enter.


Administrator: C:\Windows\system32\cmd.exe

```
24/01/2011  03:18 PM            75,112 ChangeLoginDomain.exe
07/06/2011  06:03 PM               716 ChangeLoginDomain.ini
24/01/2011  03:18 PM            75,112 CopyObjects.exe
07/06/2011  06:03 PM               808 CopyObjects.ini
05/10/2012  09:50 AM    <DIR>          core
24/01/2011  03:18 PM            75,112 CST.exe
07/06/2011  06:03 PM               705 CST.ini
24/01/2011  03:18 PM            75,112 DeleteObjects.exe
07/06/2011  06:03 PM               787 DeleteObjects.ini
05/10/2012  09:51 AM    <DIR>          deployment
24/01/2011  03:18 PM            75,112 ExportPackage.exe
07/06/2011  06:03 PM               805 ExportPackage.ini
24/01/2011  03:18 PM            75,112 HostPortTool.exe
07/06/2011  06:03 PM               716 HostPortTool.ini
24/01/2011  03:18 PM            75,112 ImportCubes.exe
07/06/2011  06:03 PM               793 ImportCubes.ini
05/10/2012  09:50 AM            21,230 importcubes.picklist
17/08/2012  03:06 PM            20,282 importcubes.picklist.full
24/01/2011  03:18 PM            75,112 ImportPackage.exe
07/06/2011  06:03 PM               805 ImportPackage.ini
05/10/2012  10:09 AM    <DIR>          installs
24/01/2011  03:18 PM            75,112 MakeFolder.exe
07/06/2011  06:03 PM               709 MakeFolder.ini
05/10/2012  09:51 AM    <DIR>          metautil
05/10/2012  09:50 AM    <DIR>          muu
05/10/2012  09:50 AM            15,428 olap_util.picklist
17/08/2012  03:06 PM            14,729 olap_util.picklist.full
05/10/2012  09:50 AM    <DIR>          OMACST
05/10/2012  09:51 AM            21,078 platform.picklist
17/08/2012  03:06 PM            20,136 platform.picklist.full
05/10/2012  10:09 AM    <DIR>          plugins
05/10/2012  09:50 AM    <DIR>          promotion
24/01/2011  03:18 PM            75,112 SetPassword.exe
01/06/2011  11:00 PM             9,716 SetPassword.html
07/06/2011  06:03 PM               715 SetPassword.ini
05/10/2012  09:50 AM    <DIR>          setpw
15/08/2012  09:17 PM           114,688 sspiauth.dll
05/10/2012  09:50 AM    <DIR>          upgrade
24/01/2011  03:18 PM            75,112 UpgradeMetadataServer.exe
07/06/2011  06:03 PM               751 UpgradeMetadataServer.ini
05/10/2012  09:50 AM    <DIR>          validate
24/01/2011  03:18 PM            75,112 ValidateServer.exe
07/06/2011  06:03 PM               732 ValidateServer.ini
              36 File(s)      1,299,383 bytes
              14 Dir(s)  46,430,908,416 bytes free

C:\Program Files\SASHome\SASPlatformObjectFramework\9.3>
```

Figure 38. SAS installation directory

Once a command window ~~has~~ is opened, navigate to your SAS installation directory located on your client PC under the SASPlatformObjectFramework directory. In the 9.3 folder you will find the ExportPackage.exe and ImportPackage.exe.

In the command window you can enter your export command as follows:

```
C:\Program Files\SASHome\SASPlatformObjectFramework\9.3>ExportPackage -host
"machinename" -port "8561" -user "username" -password "password" -package
"c:\temp\full.spk" -objects "/RootFolder" -log "c:\temp\full.log" -includeDep
```

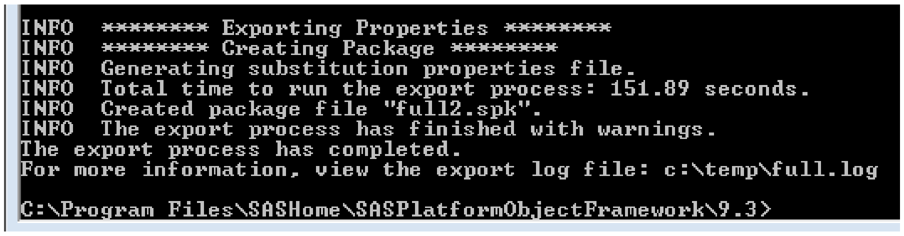
In the documentation there are many different options to choose from, an approach is to export a complete folder location:

1. RootFolder – this is the top of the folder structure within SAS® Management Console you wish to export all the objects from.
2. machinename – the SAS® Metadata server name. Can be found in the SAS® Management Console under Server Manager -> SASMeta -> SASMeta – Logical Metadata Server.
3. username – a user who has access to the entire structure and all the objects
4. password – the password of the user from above. This can also be encrypted using the following code in SAS Enterprise Guide. Replace password with the result.

   ```
   proc pwencode in="password" method=sas002;
   run;
   Result
   {SAS002}155E1222579A284D2DF718543CE8FFAB3FD8AA67
   ```

5. includeDep – This will include all the dependant objects within the export.

~~After the code has run a~~ A log file will be ~~produced~~ generated after the code is run and the output will also be sent to the screen. Review the log file for any errors or warnings.



```
INFO   ******** Exporting Properties ********
INFO   ******** Creating Package ********
INFO   Generating substitution properties file.
INFO   Total time to run the export process: 151.89 seconds.
INFO   Created package file "full2.spk".
INFO   The export process has finished with warnings.
The export process has completed.
For more information, view the export log file: c:\temp\full.log

C:\Program Files\SASHome\SASPlatformObjectFramework\9.3>
```

Figure 39. SAS Export output

In the same command window you can now test your import in your target environment.

```
C:\Program Files\SASHome\SASPlatformObjectFramework\9.3>ImportPackage -host
"machinename" -port "8561" -user "username" -password "password" -package
"c:\temp\full.spk" -preservePaths -log "c:\temp\full.log" -target "/RootFolder"  -
noexecute
```

1. machinename – the SAS® Metadata server name. Can be found in the SAS® Management Console under Server Manager -> SASMeta -> SASMeta – Logical Metadata Server. This is the target server
2. username – a user who has access to the entire structure and all the objects
3. password – the password of the user from above. This can also be encrypted using the following code in SAS Enterprise Guide. Replace password with the result.
4. preservePaths – ensures the folder structure is maintained in the target.
5. Target – target location for the objects
6. Noexecute – this option allows you to run the command and produce the log. This does not execute the import but allows you to validate the command and see what will be imported. This is advisable to reviewing before executing the import.

Once the test log has been reviewed the import can be executed.

```
C:\Program Files\SASHome\SASPlatformObjectFramework\9.3>ImportPackage -host
"machinename" -port "8561" -user "username" -password "password" -package
"c:\temp\full.spk" -target "/RootFolder"  -preservePaths -log "c:\temp\full.log"
```

SUMMARY

For the initial release a full export can be executed from the root folder within your source environment and then imported into your target environment. Some tips for the export and import are as follows:

1. Create a root folder which contains all your code within SAS® Management Console
2. Ensure your environments are setup as identical as possible. Libraries, folders, paths etc. This is important as it can cause a lot of problems during the release process.
3. Determine if you need to replicate your security structure in the source environment. If not then purely configure your target environment with your access control lists etc.
4. Post release, remember to setup your security for the objects and re-deploy any jobs necessary.

To summarize the release process steps are as follows:

1. Export code from the source environment
2. Check the code and log into the version control repository using Tortoise under the source environment.
3. Validate the export package using the import noexecute option.
4. Import code into the target environment and target folder
5. Check the code and log into the version control repository using Tortoise under the target environment.
6. Perform post import tasks
    a. Security
    b. Moving of objects to necessary folders (if required)
    c. Deployment of code
    d. Configuration of Libraries if necessary.
7. Perform post release tests

RELEASE MANAGEMENT PROCESS – PATCH RELEASE

PROCESS

For an effective release management process changes and code needs to be captured at the lowest practical level. Code should be placed into a version control repository at the object level (table, job, view etc) within the same folder structure. The import export process from SAS does not support this as the process creates a single file which encompasses all the objects.

Programmatically the metadata required for release management is also very difficult to transverse or is not exposed. For this reason and some of the limitations of the import tool (requiring a target) a robust automated export, import script is not feasible without investing significant amount of time, effort and resources to develop and maintain. A more practical approach release process is recommended for incremental or patch releases.

To enable patch or incremental releases it is recommend that you enhance your folder structure within your source and target environments to support an incremental release process.

In SAS® Management Console create an additional folder called Releases as a Root folder.

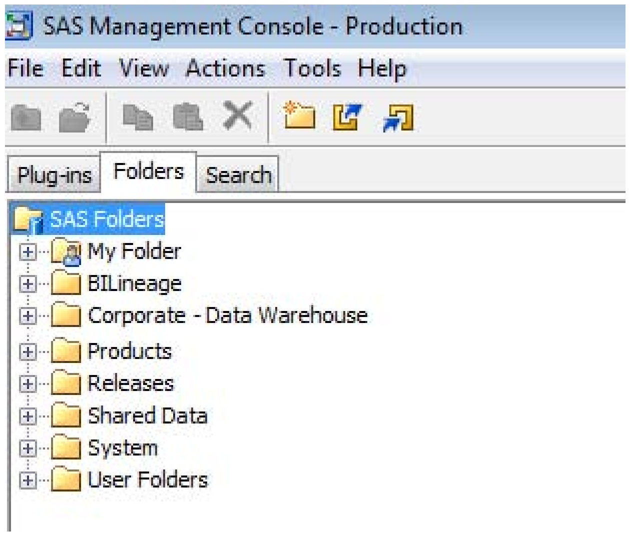


Figure 40. SAS Releases Folder

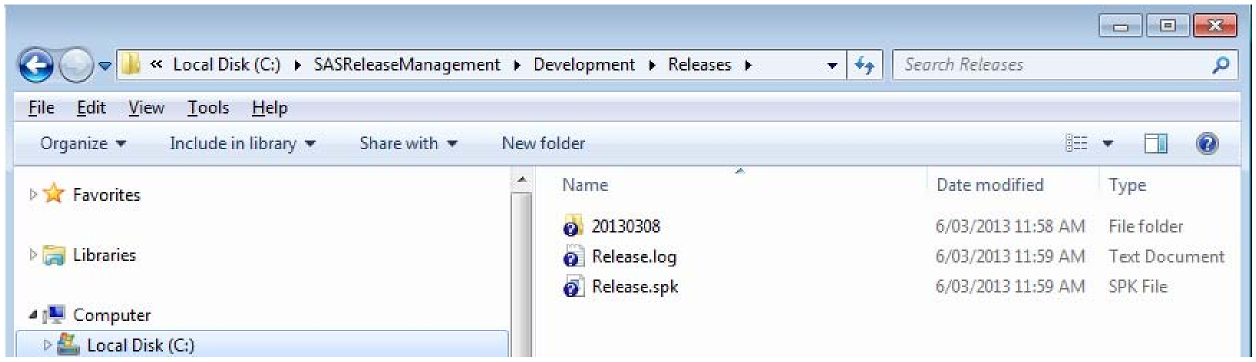
For planning purposes it is recommended that you plan your releases into the future on a regular timeframe. For example your release code every month, two months, quarter, six months etc. Each release should be assigned a release number for easy identification and tracking. Release numbers can normally be very complex, a simplier option is create a release number from a date for example 20130308 would be the release for Mar 8th 2013. Note releases are specific to each target environment. So if you are migrating code from Dev to Test and then from Test to Production you may have different release numbers for Dev to Test and then for Test to Prod.

During the development lifecycle once a component is completed and tested it can be moved into the Releases folder. This way multiple developers can contribute to an upcoming release. This release folder makes it simple to create a standard script to export all the components within the folder.

To export the code the batch export tool can be used as follows:

```
C:\Program Files\SASHome\SASPlatformObjectFramework\9.3>ExportPackage -host
"machinename" -port "8561" -user "username" -password "password" -package "
C:\SASReleaseManagement\Development\Releases\Release.spk" -objects "/Releases" -log
"C:\SASReleaseManagement\Development\Releases\Release.log"
```

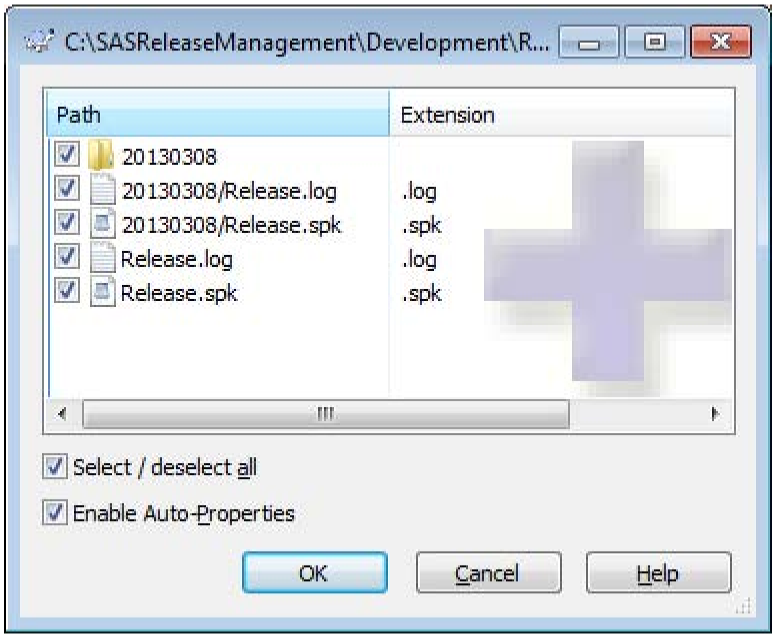
Note this time the command has been changed to include the release management folder used to store the code under version control. Also the package name is a standard name. Under your directory structure for version control you should create a folder for the upcoming release. The includeDep option has been removed so as not to export dependant objects. This is to preserve the objects in the target environment. Be careful should you include this as duplicate objects will be created such as libraries, tables etc. It is a better practice to place all the objects which have changed in the release folder for the release.


Local Disk (C:) ▸ SASReleaseManagement ▸ Development ▸ Releases ▸
Search Releases
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ New folder
Favorites
Libraries
Computer
Local Disk (C:)

Name	Date modified	Type
20130308	6/03/2013 11:58 AM	File folder
Release.log	6/03/2013 11:59 AM	Text Document
Release.spk	6/03/2013 11:59 AM	SPK File

Figure 41. Version Control folder

Notice from the export the package and the log have been created as well as the version control folder. You should copy the package and the log into the Release folder to ensure a copy is keep for the specific release. In Windows Explorer you need to multi select the folder, package and log. Right click and select TortoiseSVN -> Add.



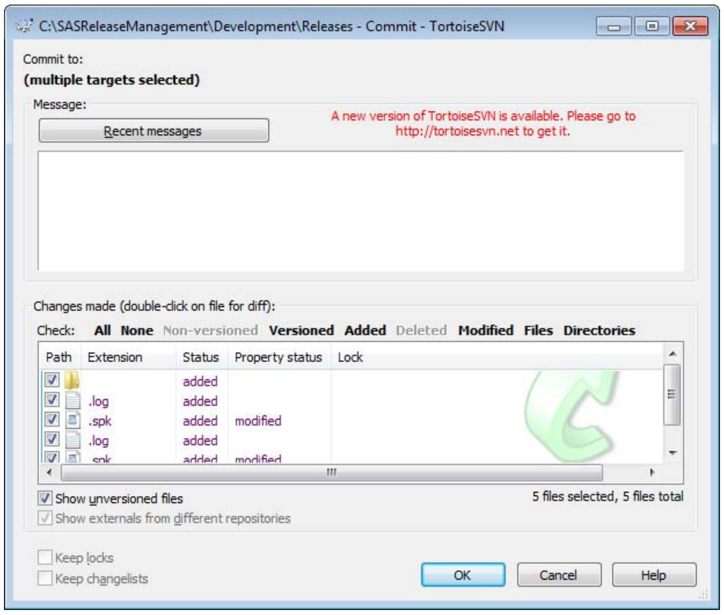
Path	Extension
☑ 20130308	
☑ 20130308/Release.log	.log
☑ 20130308/Release.spk	.spk
☑ Release.log	.log
☑ Release.spk	.spk

☑ Select / deselect all

☑ Enable Auto-Properties

Figure 42. Add new version

Select OK to add the code to Version Control. Once added right click and select SVN Commit.


C:\SASReleaseManagement\Development\Releases - Commit - TortoiseSVN

Commit to:
(multiple targets selected)

Message:

A new version of TortoiseSVN is available. Please go to http://tortoisesvn.net to get it.

Changes made (double-click on file for diff):

Path	Extension	Status	Property status	Lock
☑		added		
☑	.log	added		
☑	.spk	added	modified	
☑	.log	added		
☑	.spk	added	modified	

☑ Show unversioned files

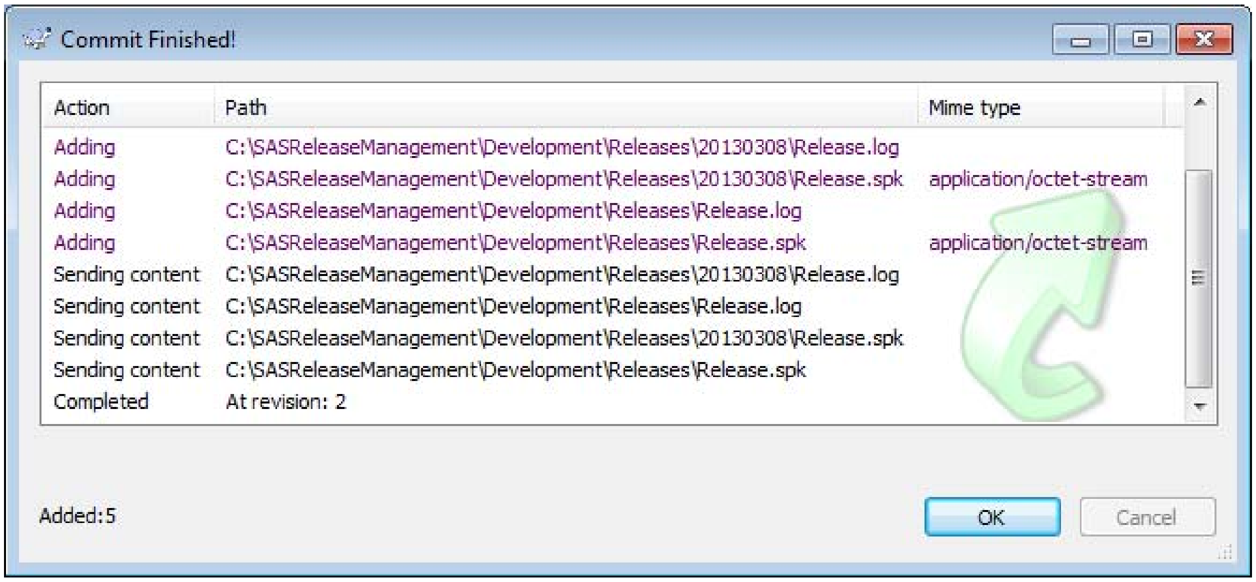
5 files selected, 5 files total

☐ Keep locks
☐ Keep changelists

Recent messages
Check: All None Non-versioned Versioned Added Deleted Modified Files Directories
Show externals from different repositories
OK
Cancel
Help

Figure 43. Commit new version

Click OK.


Commit Finished!

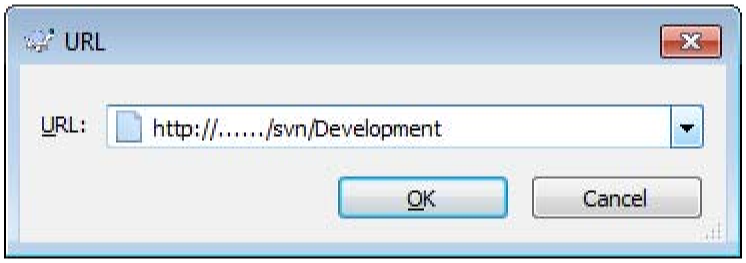
Action	Path	Mime type
Adding	C:\SASReleaseManagement\Development\Releases\20130308\Release.log	
Adding	C:\SASReleaseManagement\Development\Releases\20130308\Release.spk	application/octet-stream
Adding	C:\SASReleaseManagement\Development\Releases\Release.log	
Adding	C:\SASReleaseManagement\Development\Releases\Release.spk	application/octet-stream
Sending content	C:\SASReleaseManagement\Development\Releases\20130308\Release.log	
Sending content	C:\SASReleaseManagement\Development\Releases\Release.log	
Sending content	C:\SASReleaseManagement\Development\Releases\20130308\Release.spk	
Sending content	C:\SASReleaseManagement\Development\Releases\Release.spk	
Completed	At revision: 2	

Added:5

Figure 44. Commit confirmation

Click OK. The code now has been committed to the SVN repository. This is now available for anyone to use if necessary. Development in the existing environment can continue as a static copy of the code is available for the release into the next environment.

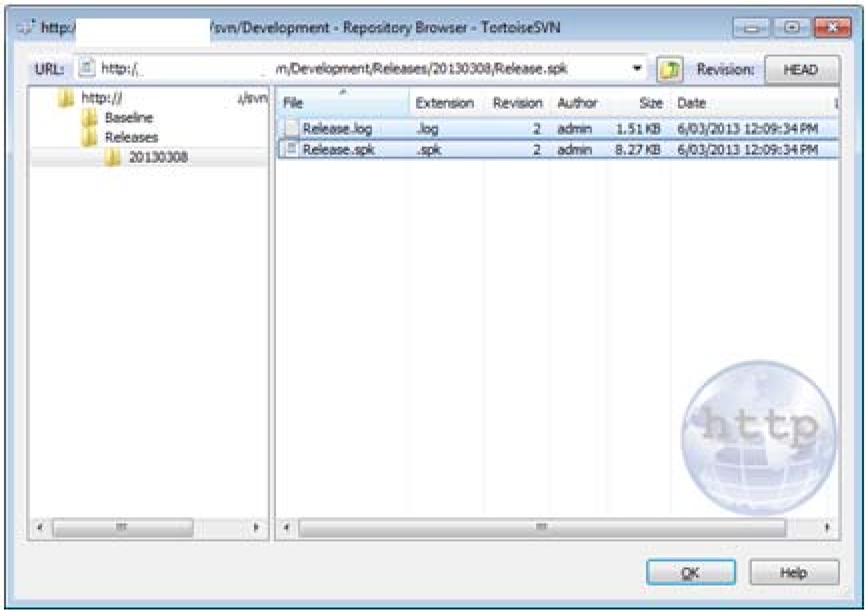
In order to promote the code to the new environment we will assume that another team member is responsible to perform the actions. From Windows Explorer you would right click and select TortoiseSVN -> Repo-Browser.



URL: http://....../svn/Development

Figure 45. Repository connect

Select the source environment location and repository. Click on OK. Enter username and password when prompted.



http:/ /svn/Development - Repository Browser - TortoiseSVN

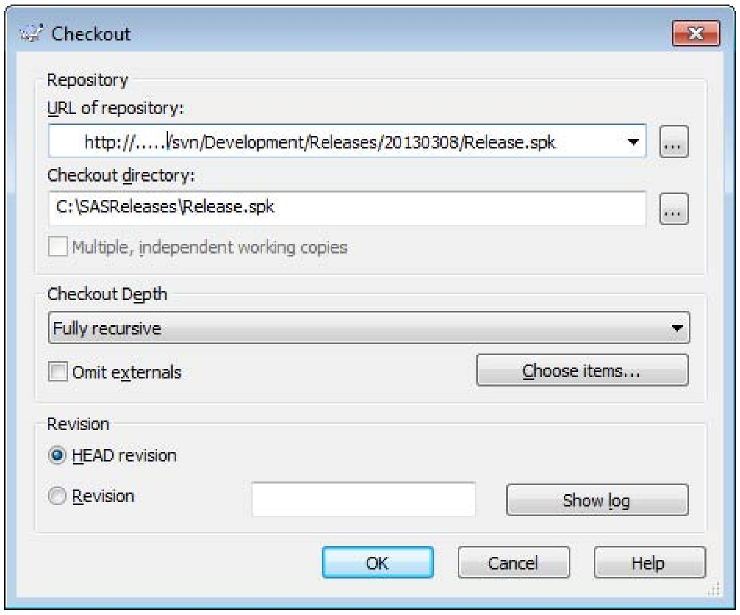
URL: http:/ m/Development/Releases/20130308/Release.spk

- http:// .../svn
  - Baseline
  - Releases
    - 20130308

File	Extension	Revision	Author	Size	Date
Release.log	.log	2	admin	1.51 KB	6/03/2013 12:09:34 PM
Release.spk	.spk	2	admin	8.27 KB	6/03/2013 12:09:34 PM

Figure 46. Repository browser

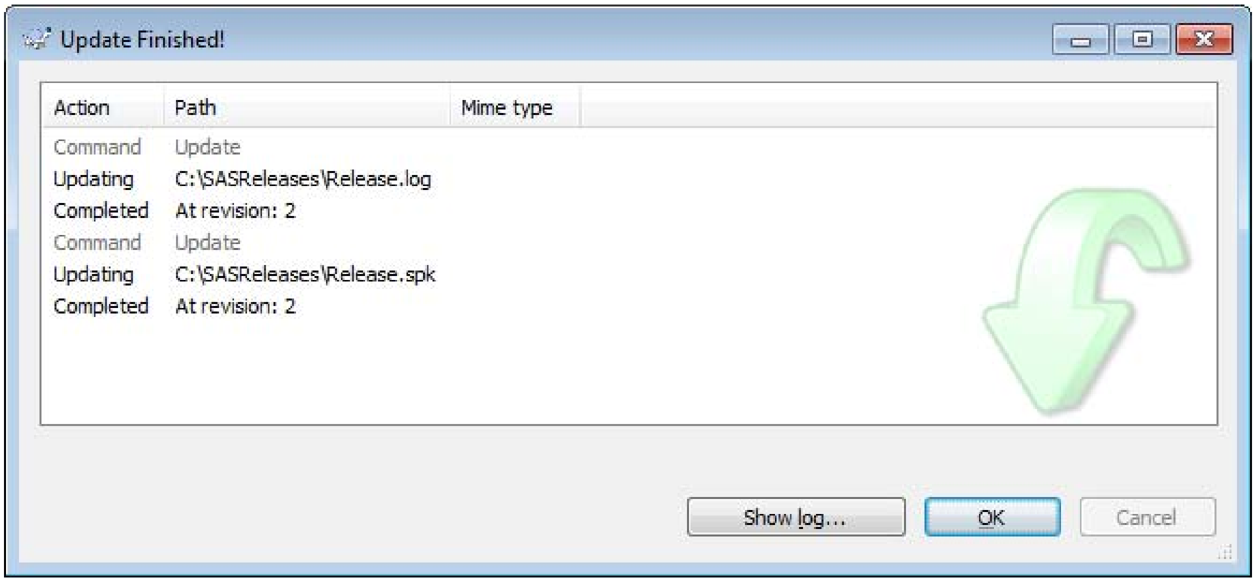
Navigate to the release folder and select the package. Right click and select Checkout



URL of repository: http://...../svn/Development/Releases/20130308/Release.spk
Checkout directory: C:\SASReleases\Release.spk
Checkout Depth: Fully recursive

Figure 47. Repository Checkout

Select the folder you wish to check the code out to and select OK.



Action	Path	Mime type
Command	Update	
Updating	C:\SASReleases\Release.log	
Completed	At revision: 2	
Command	Update	
Updating	C:\SASReleases\Release.spk	
Completed	At revision: 2	

Figure 48. Repository Checkout confirmation

Validate the checkout was successful and click OK.

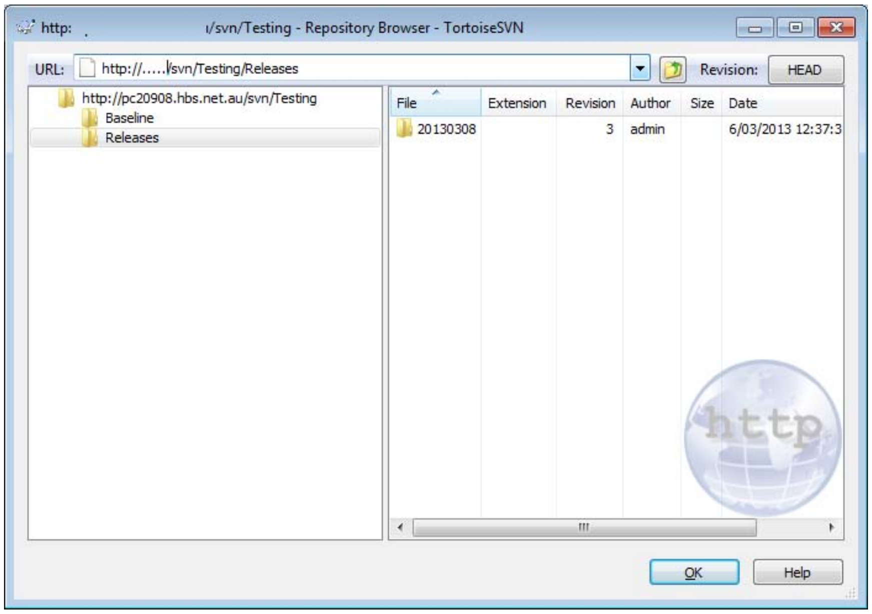
Open a command prompt and navigate to your batch import tool.

```
C:\Program Files\SASHome\SASPlatformObjectFramework\9.3>ImportPackage -host
"machinename" -port "8561" -user "username" -password "password" -package
"C:\SASReleases\Release.spk" -target "/Releases" -preservePaths -log
"C:\SASReleases\Promote.log"
```

Upon successful import of the new objects you can now perform any post migration activities such as:

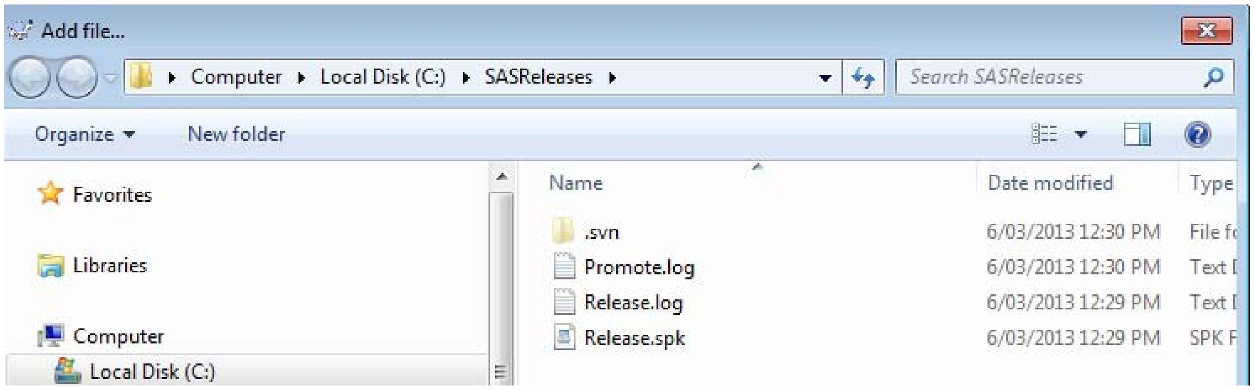
1. The removal of old objects with the same name.
2. The movement of the objects to the correct folders
3. Updates to any dependant objects (Information maps etc).
4. Redeployment of jobs.
5. Updates to job processes
6. Test and validate code

Once completed the release code needs to be merged into the release for the target environment. In Windows Explorer Right Click and open the Repo-Browser under TortoiseSVN connecting to the target repository.


http: ı/svn/Testing - Repository Browser - TortoiseSVN
URL: http://.....\svn/Testing/Releases
Revision: HEAD
http://pc20908.hbs.net.au/svn/Testing
Baseline
Releases
OK
Help

File	Extension	Revision	Author	Size	Date
20130308		3	admin		6/03/2013 12:37:3

Right Click under releases and select Create Folder. Enter the Release Number for the folder and then navigate into the folder. Right Click and select Add File.



Name	Date modified	Type
.svn	6/03/2013 12:30 PM	File fo
Promote.log	6/03/2013 12:30 PM	Text D
Release.log	6/03/2013 12:29 PM	Text D
Release.spk	6/03/2013 12:29 PM	SPK F

Select the Release Package and the Promote Log. These need to be done individually. Once completed the code will be stored in your repository. A full copy of the code should be exported using the batch export tool and placed into the repository under the baseline folder. After every successful release a new baseline should be created to ensure that there is also a complete copy of the working code which can be used to restore and environment or component of an environment.

SUMMARY

Patch releases are a little more complicated than a full release. Using a incremental release processes within a data warehouse environment is highly recommended as predefined release dates communicates a commitment to your stakeholders that new requirements and functionality will be released. The requirements and functionality within each release can be negotiated and prioritized with the stakeholders. Each release should be assigned a number per target environment and each environment should have a SVN repository created with a baseline and a releases folder. In SAS you need to create a Releases folder at the root within each environment to support incremental releases. Once completed incremental releases can be achieved as follows

1. Export your release to a local folder monitored by SVN using the Batch Export tool into a common location.
2. Copy the release to a specific release directory with the SVN directory.
3. Add and commit the code to the SVN repository.
4. Checkout the release package from the Source repository.
5. Import the release package into the Target repository using the Batch Import tool.
6. Perform any post import steps.
7. Test your release.
8. Create a release directory within the Target environment
9. Import you Release package and logs into the target release directory
10. Create a complete code export using the batch export
11. Import the complete code export into the Baseline directory for the Target environment

CONCLUSION

SAS® DI Studio supports version control, but no SAS® products support the use of version control within a Release Management process. This is a very manual and error prone process which is time consuming and potentially disastrous should errors occur. Release Management needs to be considered at the beginning of the project so that the appropriate decisions can be made on how code will be controlled, managed and the environment configured to support a release process. SAS® DI Studio should be used to archive changes before and after any changes to the environment is made. This way code can be rolled back or forward as necessary insuring the quality and integrity of the release.

Using the Batch Import and Export tools an effective release management process can be constructed utilizing Subversion for version control. These scripts can be automated and included into the release process of most projects.

REFERENCES

SAS®. SAS(R) 9.3 Intelligence Platform: System Administration Guide, Second Edition. Using the Promotion Tools SAS® Knowledge Base. 2013.
http://support.sas.com/documentation/cdl/en/bisag/65422/HTML/default/viewer.htm#titlepage.htm

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: John Heaton
Enterprise: Heritage Bank
Address: 400 Ruthven Drive
City, State ZIP: Toowoomba, Queensland, Australia, 4350
Work Phone: +61 4694 9129
Fax: +61 4694 9785
E-mail: Heaton.j@heritage.com.au
Web: http://www.heritage.com.au