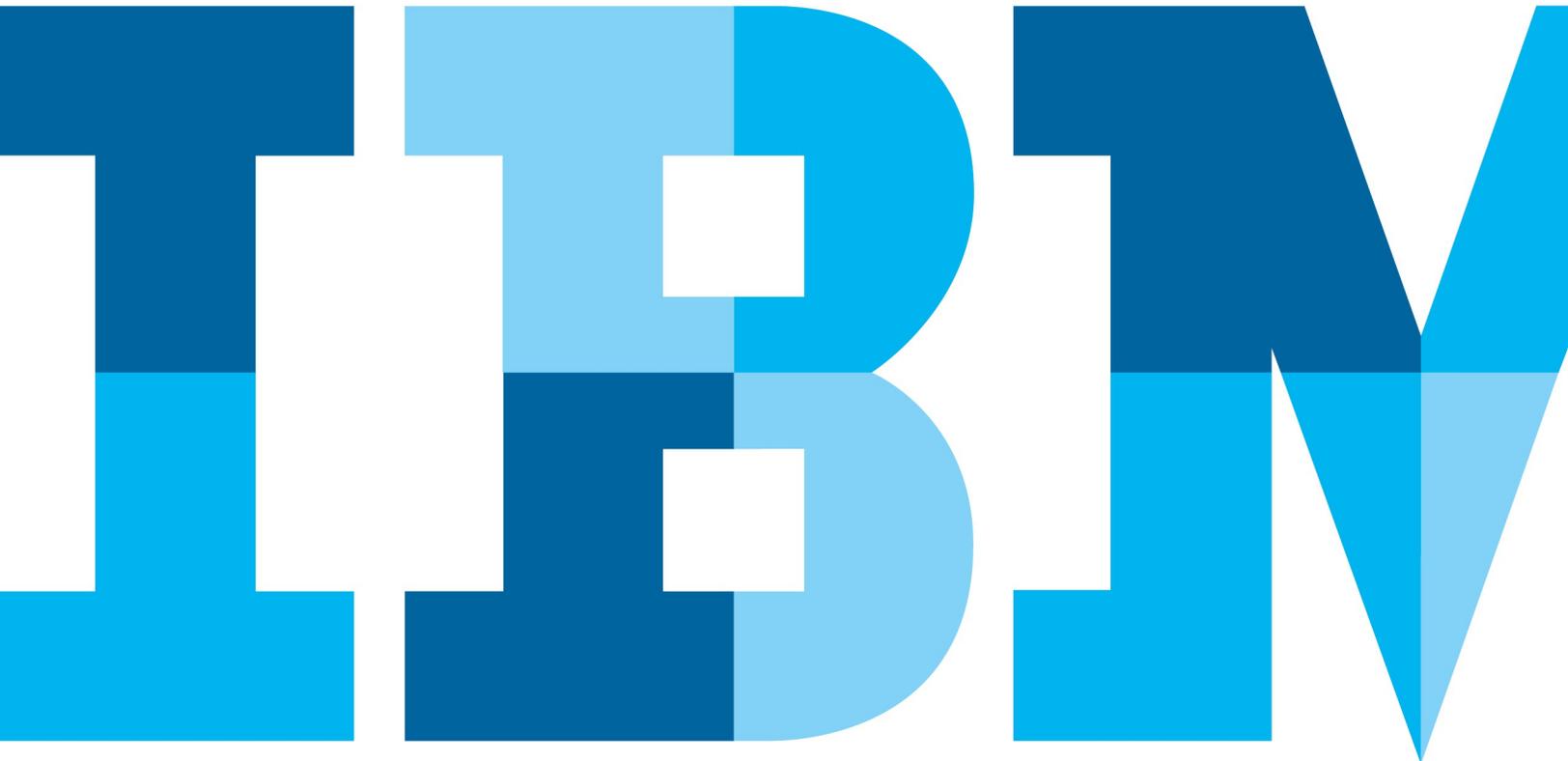


IBM Rational Team Concert Workshop – Distributed SCM and Shipping RTC SCM Change Sets Across Disconnected Networks

Lab Exercises



Contents

LAB 1 SOLUTION OVERVIEW.....	3
1.1 SUGGESTED READING.....	3
1.2 SCENARIO DESCRIPTION.....	4
1.3 SOLUTION OUTLINE.....	4
1.4 IMPORTANT CONSIDERATIONS.....	6
1.4.1 SECURITY.....	6
1.4.2 LICENSING.....	6
1.4.3 HARDWARE.....	6
1.4.4 HOST NAMES.....	6
1.4.5 APPLICATION SERVER PORTS.....	7
1.4.6 RTC VERSION DEPENDENCIES.....	7
1.4.7 BIDIRECTIONAL SHIPPING.....	7
1.4.8 AVOIDING SHIPPING ERRORS.....	7
1.4.9 INDEPENDENT SHIPPING.....	8
1.4.10 SHIPPING INDEPENDENTLY BETWEEN MULTIPLE SITES.....	8
1.4.11 HANDLE MULTIPLE SHIPPING SERVERS ON ONE SITE.....	9
1.4.12 SHIPPING USERS.....	9
1.4.13 BACKUP.....	9
1.4.14 BACKUP FOOTPRINT.....	9
1.4.15 DATABASE.....	10
1.4.16 AUTOMATION.....	10
1.4.17 WORK ITEM VISIBILITY.....	10
LAB 2 PREPARING A SHIPPING SERVER.....	11
2.1 CREATE A SHIPPING SERVER TEMPLATE.....	11
2.1.1 DOWNLOAD AND INSTALL THE SOFTWARE.....	11
2.2 CONFIGURE THE APPLICATION SERVER.....	13
2.3 SET UP THE SERVER.....	15
2.3.1 CREATE USER ACCOUNT.....	18
2.3.2 CONFIGURE THE SERVER FOR DISTRIBUTED SCM.....	19
2.3.3 CREATE A PROJECT AREA.....	19
2.3.4 CONFIGURE THE PROJECT AREA FOR DISTRIBUTED SCM.....	20
2.4 DOWNLOAD AND INSTALL THE ECLIPSE CLIENT USED FOR SHIPPING.....	22
2.4.1 CONNECT THE ECLIPSE CLIENT TO THE SHIPPING SERVER.....	23
2.5 BACK UP YOUR SHIPPING SERVER TEMPLATE.....	25
2.6 CREATE A DEDICATED SHIPPING SERVER INSTANCE.....	27
LAB 3 USING DISTRIBUTED SCM FOR SHIPPING.....	29
3.1 CONFIGURE THE PRODUCTION SERVER FOR DISTRIBUTED SCM.....	29
3.1.1 CREATE AN EXAMPLE PRODUCTION SERVER FOR SITE A.....	29
3.1.2 ENABLE DISTRIBUTED SCM ON THE PRODUCTION SERVER SITE A.....	30
3.1.3 CREATE A SHIPPING USER AND CONFIGURE THE PROJECT FOR SHIPPING.....	30
3.2 CREATE A DEDICATED SHIPPING SERVER INSTANCE.....	31
3.3 TRANSFER THE SCM CHANGES TO THE SHIPPING SERVER.....	33

- 3.3.1 START YOUR SHIPPING SERVER.....33**
- 3.3.2 CONNECT THE ECLIPSE CLIENT TO THE PRODUCTION SERVER.....33**
- 3.3.3 CREATE A SHIPPING STREAM.....35**
- 3.3.4 ACCEPT SCM CHANGE SETS TO THE SHIPPING SERVER.....36**
- 3.4 BACKUP THE SHIPPING SERVER FOR SHIPPING.....40
 - 3.4.1 CREATE A PARTIAL BACKUP OF THE SHIPPING SERVER.....40**
- LAB 4 DISTRIBUTED SCM AND SHIPPING CHANGES TO A SERVER.....43**
 - 4.1 SETUP THE SHIPPING SERVER FORM A PARTIAL BACKUP.....43
 - 4.2 CONFIGURE THE SITE B PRODUCTION SERVER FOR DISTRIBUTED SCM.....43
 - 4.2.1 CREATE AN EXAMPLE PRODUCTION SERVER FOR SITE B.....44**
 - 4.2.2 ENABLE DISTRIBUTED SCM ON THE PRODUCTION SERVER SITE B.....45**
 - 4.2.3 CREATE A SHIPPING USER AND CONFIGURE THE PROJECT FOR SHIPPING.....45**
 - 4.3 DELIVER YOUR SHIPPED CHANGES INTO THE PRODUCTION SERVER.....46
 - 4.3.1 CONNECT THE ECLIPSE CLIENT TO THE PRODUCTION SERVER AND THE SHIPPING SERVER.....46**
 - 4.3.2 REPLICATE YOUR SHIPPED CHANGES FROM THE SHIPPING REPOSITORY WORKSPACE INTO THE PRODUCTION SERVER STREAM.....47**
 - 4.3.3 SUBSEQUENT SHIPPING51**
 - 4.3.4 PREPARE PROJECTS FOR SHIPPING52**
- APPENDIX A BACKUP.....53**
- APPENDIX B TROUBLESHOOTING.....54**
- APPENDIX C NOTICES.....55**
- APPENDIX D TRADEMARKS AND COPYRIGHTS.....57**

Lab 1 Solution Overview

Sometimes it is necessary to contain development data in disconnected networks. However it is still necessary to be able to share work products with other teams working in other networks. Some products such as IBM Rational Clear Case provide capabilities to achieve this by shipping the change data in some format back and forth and replicate them in and out of a system. Rational Team Concert, with Version 4.x today, does not provide this kind of export mechanism. It is however, still possible to use the built in distributed SCM Capability and backup capabilities to achieve the same goal.

This workshop provides a detailed description how to achieve this goal. This workshop also explains the steps necessary to set up distributed SCM in general and how it can be used in situations with connected networks.

This lab provides with an overview of the solution used in the workshop. It also provides information about considerations for planning to implement this solution.

Hardware Requirements

The scenario in this workshop assumes, that you have two server machines that can work as production servers to ship between. In addition you would have a machine used to install shipping servers.



The workshop provides guidance how to run it on a single hardware, with the production servers and shipping servers set up on the same machine.

If you perform the workshop on one machine, you have to make sure only to run one of the production servers at any one time. Alternatively you can configure each server to use its own unique ports to allow all of them to run on one hardware without conflicting ports.

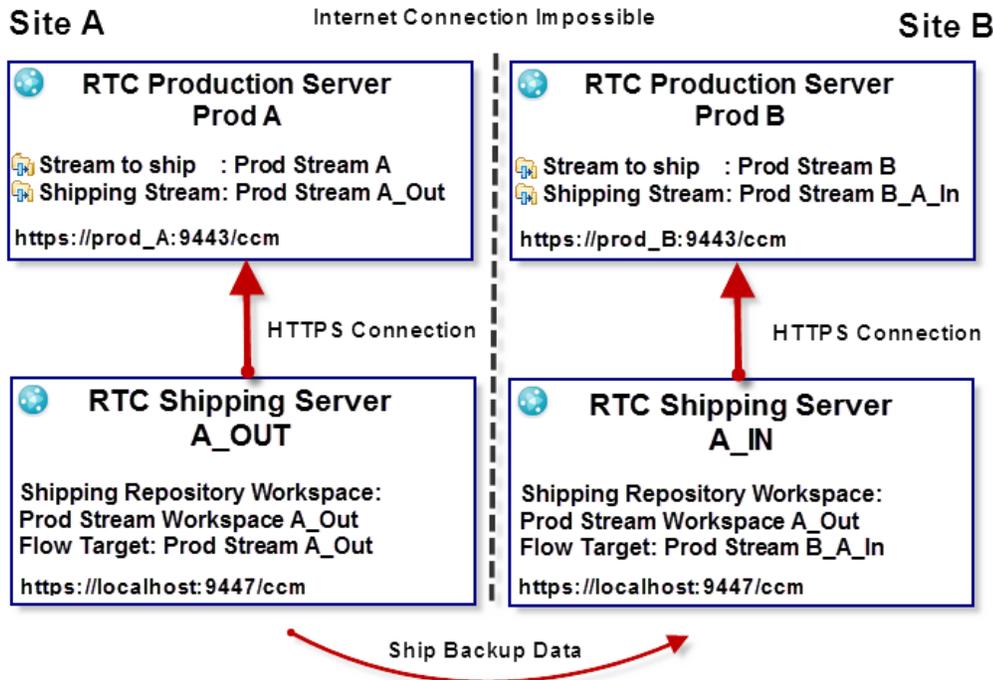
1.1 Suggested reading

Please consider reading this material as well.

- [Flow changes cross repositories with Rational Team Concert](#)
- [Backup the Rational Solution for Collaborative Lifecycle Management](#)
- [Concerns when configuring Apache Tomcat](#)

1.2 Scenario Description

The scenario assumes the topology of the picture below.



RTC Production servers are running on two sites, Site A and Site B. The networks of Site A and Site B are physically disconnected.

On each site a shipping server is set up that can connect to the production server on its site.

User accounts are available with access to the relevant SCM data and the permission to replicate change sets on the Production servers Prod A and Prod B.

1.3 Solution Outline

The Production Servers on Site A and Site B are enabled for Distributed SCM.

On each site a Shipping Server is set up. The shipping servers on both sites will operate with identical settings, including the public URI. The shipping servers are enabled for Distributed SCM.

An RTC Eclipse Shipping Client is available on the machine that hosts the Shipping Server A and Shipping Server B respectively.

The strategy to bring SCM data from Site A to Site B, works as follows.

1. Get the SCM Changes from Production Server A on Site A
 - 1.1. Make sure the Shipping Server A_OUT on Site A is started.

- 1.2. Start the Shipping RTC Eclipse Client co-located to the Shipping Server A_OUT and connect to Production Server A.
- 1.3. For a Stream Prod Stream A you want to ship to Production Server B, create a Shipping Stream Prod Stream A_Out.
 - 1.3.1. Create a Shipping Repository Workspace Prod Stream Workspace A_Out stored in the the Shipping Server A_OUT repository on site A to get the current changes for the stream.
 - 1.3.2. Set the flow target of the Shipping Repository Workspace Prod Stream Workspace A_Out to the stream Prod Stream A_Out located on Production Server A.
 - 1.3.3. Accept all incoming changes and component additions to bring your Shipping Repository Workspace Prod Stream Workspace A_Out into the desired state of the shipping stream. It is not necessary to load the repository workspaces to disk in the Shipping RTC Client.
- 1.4. Stop the Shipping Server A_OUT.
- 1.5. Backup the Shipping Server A_OUT.
2. Send the backup data of Shipping Server A_OUT to Site B.
3. Restore the Shipping Server A_IN on site B, from the backup of Shipping Server A_OUT while the Shipping Server A_IN is down.
4. Deliver the SCM changes Shipping Server A_IN to the Production Server B
 - 4.1. Start the restored Shipping Server A_IN on site B
 - 4.2. Start the Shipping RTC Eclipse Client co-located to the Shipping Server A_In and connect to Production Server A.
 - 4.3. For each Shipping Repository Workspace Prod Stream A_Out create a Shipping Stream Prod Stream A_In if it is not yet available..
 - 4.3.1. Set the flow target of the Shipping Repository Workspace Prod Stream A_Out to the corresponding Shipping Stream Prod Stream A_In on Production Server B.
 - 4.3.2. Deliver the outgoing changes and component additions to the stream
 - 4.4. Stop the Shipping Server A_IN on site B

The changes from streams on Production Server A have been successfully shipped to streams on Production Server B. Changes from and to the affected Stream on the Production Server B can now be integrated.

A similar scenario involving another pair of shipping servers can be used to ship changes from the Production Server B back to the Production Server A. See Important Considerations below for more information.

1.4 Important Considerations



Important Considerations!

The following information should be carefully considered when implementing this strategy.

1.4.1 Security

When shipping the backup data back and forth you should consider to encrypt the data. Otherwise a capable person could recreate the shipped data on any machine and access it. Use a company encryption solution. There are also free open source solutions available such as [TrueCrypt](#).

1.4.2 Licensing

In order to use the distributed SCM capability

- A Rational Team Concert **Developer** or **Developer for IBM Enterprise Platforms** license is required on the production servers and the shipping server.
- It is not possible to mix the licenses on any of the servers with **Developer for Workgroup** or **10 Free Developer** licenses. This would disable the distributed SCM capabilities. See the main download page for your release of your version of RTC. As an example see the information [for RTC 4.0.1](#).

1.4.3 Hardware

The shipping servers can be installed on a dedicated hardware or on the same hardware used by the production server. In general it would be possible to install a shipping server and a shipping client on a single machine, or on different machines. The approach described in this document requires the shipping server and client to be installed on the same machine.

1.4.4 Host Names

Any CLM server such as the Production and Shipping Servers, require a stable public URI. The solution described in this document uses **localhost** for the shipping servers to simplify network management. Due to using **localhost** in the public URI, it is impossible to contact the Shipping Servers from any other machine in the network of its site. The solution works, because the Shipping Servers only need to be contacted by the Shipping Client and all other network activity is against a DNS based public URI for the Production Servers. If this is not feasible, and the shipping client needs to be located on a different machine, it is necessary to provide a public URI with a host name that is the same on each site. This can be done using DNS e.g. providing a machine independent host name that is the same on every site, or using the hosts file of the client machines, using a local host entry to the IP of the shipping server.

1.4.5 Application Server Ports

When using the existing production hardware for the shipping server it is necessary to plan the install of the shipping server.

Make sure to choose the ports to be used by the shipping server that don't create a conflict with any of the production servers you want to ship to. The instructions below assume that the production servers Prod A and Prod B use the same kind of setup using the same ports and thus don't conflict with the port of the shipping servers.

1.4.6 RTC Version Dependencies

Please note, distributed SCM requires the RTC Applications Version to be compatible for Distributed SCM to work.

Server Versions must be compatible



Distributed SCM requires the RTC Applications Version to be compatible for Distributed SCM to work.

1.4.7 Bidirectional Shipping

It is possible to ship changes between two sites.

For clarity and automation the suggestion is to have a dedicated **Shipping Stream** and a related **Shipping Repository Workspace** for **each Stream and each direction** on the Production and Shipping Servers.

For example for shipping from Production Server A to Production Server B provide a stream **Shipping <OriginalStreamNameProductionServerA> A_Out** as Shipping OUT Stream on Production Server A. Use a dedicated repository workspace **Shipping <OriginalStreamNameProductionServerA> Workspace A_Out** as Shipping OUT Repository Workspace for this direction. On site B provide a stream **Shipping <OriginalStreamNameProductionServerB> A_In** as Shipping IN Stream to deliver the changes from site A to the Production Server B.

For shipping from Production Server B provide a stream **Shipping <OriginalStreamNameProductionServerB> B_Out** as Shipping OUT Stream on Production Server B. Use a dedicated repository workspace **Shipping <OriginalStreamNameProductionServerB> Workspace B_Out** as Shipping OUT Repository Workspace for this direction. On site A provide a stream **Shipping <OriginalStreamNameProductionServerA> B_In** as Shipping IN Stream to deliver the changes from site B back to the Production Server A.

1.4.8 Avoiding Shipping Errors

When using the strategy described in 1.4.7 Bidirectional Shipping shipping errors can easily be avoided. As a rule of thumbs, there should never be any conflicting changes during shipment.

The shipping changes to other sites would only accept incoming changes from the Shipping **Out** Stream into the Shipping **OUT** Repository Workspace. No conflicting change should ever happen on accept. If this happens, the Shipping **OUT** Repository Workspace has been manipulated. In this case

- Delete the Shipping OUT Repository Workspace
- Create a new Shipping OUT Repository Workspace
- Connect it to the corresponding Shipping OUT Stream and accept the changes.

The user receiving changes shipped from other sites would only deliver changes from the Shipping Out Repository Workspace into the Shipping In Stream for the remote site. No conflicting change should ever happen during delivery. If this happens, the Shipping IN Stream has been manipulated. In this case

- Delete the Shipping IN Stream
- Create a new Shipping IN Stream
- Connect the Shipping OUT Repository Workspace back to the Shipping IN Stream and deliver the changes.

1.4.9 Independent Shipping

It is desirable to have independent shipping from a site to another site: allow to ship changes from the outgoing site to any other site without having to synchronize and wait for changes to come back first.

To implement this, each site that ships data to other sites should have a dedicated Shipping Server to ship data out to other sites. Each site that needs to receive data from another site should have a dedicated shipping server that is only used to receive data from this site.

Example: Site A ships to other sites and uses the **Shipping Server A_OUT** as Shipping OUT Server for this purpose. This server is used to accept changes from Production Server A that need to be shipped somewhere else.

Site B that needs changes from Production Server A should have its own Shipping IN Server **Shipping Server A_IN** used to restore backup data from Shipping Server A_Out and to deliver data to site B's Production Server B.

As described in 1.4.7 Bidirectional Shipping page 7 above, use dedicated streams for shipping out and in. The outgoing changes are shipped with the Shipping OUT Server. The incoming changes are received with the Shipping IN Server.

1.4.10 Shipping Independently Between Multiple Sites

It is possible to ship changes to multiple sites and receive changes from multiple sites. To reduce complexity and synchronization when shipping data, the schema as described in 1.4.9 Independent Shipping on page 8 above should be used for shipping between multiple sites.

Each site that ships out should have its own **Shipping OUT Server** to ship changes out to other sites.

Each site that needs to ship data in from another site should have a dedicated **Shipping IN Server** for each site from which it needs to receive data.

This allows to send backup data for shipping independent from all the other sites.

1.4.11 Handle Multiple Shipping Servers On One Site

It is possible to have multiple Shipping Servers on a single machine as described in 1.4.3 Hardware page 6 above on each site and still use the same hostname as described in 1.4.4 Host Names on page 6 above and port number as described in 1.4.5 Application Server Ports page 7 above. The restriction is that only one of the shipping servers can be up and running at any time.

1.4.12 Shipping Users

To be able to ship changes between servers on one site, it is necessary to have a user ID on the production server and a user ID on the shipping server. The user ID's need to have a Team Concert Developer License and at least the repository role JazzUsers to access their server's repository. The users need to be added to the Process Areas with a role that allows to perform the necessary SCM operations, including Replicating change sets.

It is not necessary to use the same ID or password on the servers.

1.4.13 Backup

There are various possible ways to create the backup of the Source Shipping Server.



Backup and Restore

See Appendix A Backup for additional information.

The description below uses Derby as the database.

The document below assumes that a full backup of the database and other required data is taken and shipped from the Source Shipping Server A to the Target Shipping Server B.

1.4.14 Backup Footprint

Shipping always a complete backup can require a huge amount of data that needs shipping. It is possible to use incremental backups for shipping. This reduces the size of the backup data that needs shipping. See Appendix A Backup for additional information about backup.

Incremental Backup is only available for the database. It is still required to do a complete backup of the index files and the Application Server and the Jazz Configuration files in case any of these they changed.

If using incremental backup it is only required to sent the increment of the backup to the other sites. In case of shipping errors such as missing increments the database restore should detect the error and prevent a restore until the issue is fixed.

1.4.15 Database

The workshop below uses a standard Tomcat/Derby setup. For large amounts of files you might have to use an enterprise database such as DB2 for the shipping server. This would impact the backup process.

1.4.16 Automation

It is possible to automate accepting the changes into the Shipping Repository Workspace, server shutdown, backup, packaging, encrypting, decrypting, unpacking, restoring the backup, server start up, delivery of the changes to the Shipping IN Stream. The mechanisms described in 1.4.8 Avoiding Shipping Errors on page 7 above and 1.4.14 Backup Footprint would detect inconsistencies and cause an error that can be fixed by the shipping user.

1.4.17 Work Item Visibility

If change sets are linked to work items, and the user should be able to follow the artifact link, it is necessary to make the servers involved friends. See [Flow changes cross repositories with Rational Team Concert](#) for a description on how to do this. If there is no connectivity to the remote server, that hosts the work item, it is not possible to follow the link.

Lab 2 Preparing a Shipping Server

This lab explains how to install and set up a Rational Team Concert server that can be used as shipping server. Instead of always having to do a fresh install and setup, the procedure below allows to back up the set up and to create a template for a shipping server that can be restored when needed.

2.1 Create A Shipping Server Template

The following Instructions show the step by step process to setup shipping servers to be used to ship the SCM data across two disconnected production servers. The method described below allows to create a shipping server template install, that can be reused as often as a new shipping server is needed on any site. If this is not desired, you can follow these steps to manually install and set up a shipping server.

RTC Install Flavor Used



The instructions will use a Tomcat/Derby ZIP install for the shipping servers. This allows to ship the whole install of client and server. The strategy works for other setups as well, but may then require to only backup the repository data.

Operating systems other than Windows



The instructions are for Windows, however, you can follow the instructions on a different operating system by adjusting to the different behavior and operating system paths.

2.1.1 Download and Install the Software

First download and install the necessary software. The downloads will be reused across the workshop.

- __1. Download and install the server software.
 - __a. Open the [RTC Release Download page on Jazz.net](#).
 - __b. Open the **All Downloads Tab**.

- __c. Download the ZIP version of the Jazz Team Server and the CCM Application for your release from the Rational Team Concert All Downloads Page

Plain .zip Files

Extract these .zip files to quickly install specific IDE-based clients or other tools. The Client for Microsoft Visual Studio IDE can only be installed by using the web installer or by installing locally using the IBM Installation Manager repository. Compressed (.zip) files are used for installing the server and license keys on z/OS and IBM i platforms for milestone builds. LICPGM (IBM i) and SMP/E (zOS) packages will be available for the final release.

Description	Platform
Compressed File (.zip)	
Jazz Team Server and the CCM Application, and Trial licenses for Rational Team Concert	Windows x86 (623.74 MB)
	Windows x86-64 (636.45 MB)
	Linux x86 (609.95 MB)
	Linux x86-64 (616.08 MB)
Client for Eclipse IDE	Windows x86 (404.84 MB)
	Linux x86 (392.57 MB)
	Mac OS X (321.97 MB)

ZIP Version not available or Installation Manager is preferred

If the zip version of JTS and CCM is not available for your version or you prefer to use Installation Manager perform the install using the available media, you can Adjust the setup to the paths given in the instructions.



Installation Manager uses shared data across install on the same machine. Therefore it is not proven that you can simply zip the shipping server template data.

A quick test did not reveal any issues. If you have trouble or don't want to take the chance, you can repeat the install and setup described in this section for each shipping server instance. Use the desired shipping server name to create the shipping server root folder instead.

- __d. Create a folder [C:\ShippingServer<Name>](#) on the machine acting as shipping server.

Replace the place holder suffix **<Name>** by **Template** to install a shipping server template. We will refer to the path as **<ShippingServerFolder>** in the following document. If you use different paths, Adjust the path as required.

Install without using a shipping server template

Replace the place holder suffix <Name> by a meaningful name in your context. The suffix Template will be replaced by the real name of the shipping server when using the template to create the real shipping server.

if installing the shipping servers individually, replace <Name> by <Site>_<Direction>, where <Site> is the site of the repository and <Direction> is OUT or IN. For a Site A and the shipping direction OUT replace <Name> with **A_OUT**.

- __e. Unzip the downloaded Zip version of JTS and CCM to the folder **<ShippingServerFolder>\JazzTeamServer**.

Compatible ZIP Tools

The standard zip tool on Windows has issues with long path names. To avoid these issues use [7Zip](#) to unzip.

2.2 Configure the Application Server

You will now configure the shipping server to be able to run on a set of ports different from the ports used by a default install. This allows to run the shipping server on the same machine that hosts the production server.

- __2. Prepare Tomcat for Running JTS and CCM. See also the [Deployment Wiki](#), especially the entry on [Configuring Tomcat](#):

Plan the Ports of the Shipping Server



This step is essential if the shipping server is located on the same machine as the production server or any other Jazz based server. The default install uses the same ports for Tomcat and that can cause problems running the servers.

The ports used below work on a standard system for a second Tomcat. On windows you can use **netstat -a** on the console to make sure you pick unused ports. Other operating systems have similar capabilities.

If you run your shipping servers on a dedicated machine, you can skip this step and use the default ports.

- __a. Open the **<ShippingServerFolder>\JazzTeamServer\server\tomcat\conf\server.xml** file for editing.

- __b. Locate the line for the **shutdown** server port

```
<Server port="9005" shutdown="SHUTDOWN">
```

and change the port to 9007. The line should look like below:

```
<Server port="9007" shutdown="SHUTDOWN">
```

- __c. Locate the lines to define the connector port

```
<Connector port="9080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="9443" />
```

Change the Connector port to **9087** and the redirect port to **9447**. The lines should look like below:

```
<Connector port="9087" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="9447" />
```

- __d. Locate the next uncommented line for the second connector port definition

```
<Connector port="9443"
```

Change the Connector port from 9443 to **9447**. the line should look like

```
<Connector port="9447"
```

- __e. Locate the line defining the AJP Connector

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="9009" protocol="AJP/1.3" redirectPort="9443"
```

Change the AJP Connector port from 9009 to **9010** and the redirect port to **9447**. The line should look as below.

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="9010" protocol="AJP/1.3" redirectPort="9447"
```

- __f. Save the changes to the server.xml file.

- __3. You can adjust other settings as usual, for example language and regional settings.

2.3 Set Up the Server

This section explains how to set up the server to use it later.

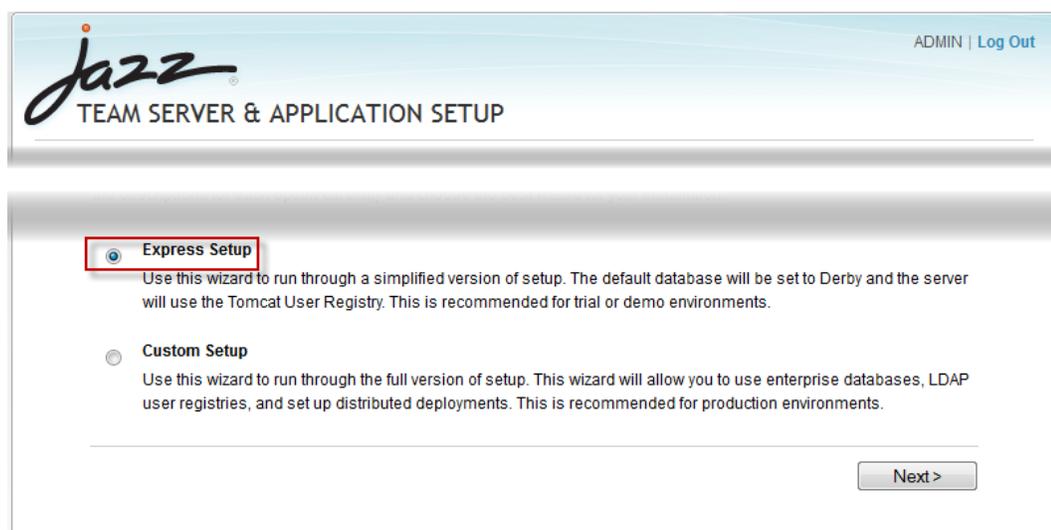
- __4. Use `<ShippingServerFolder>\JazzTeamServer\server\server.startup.bat` to start the Shipping Server.

- __a. Monitor the startup for conflicting ports for conflicts. Adjust ports in case you see a conflict.

- __5. Run the Jazz Server setup

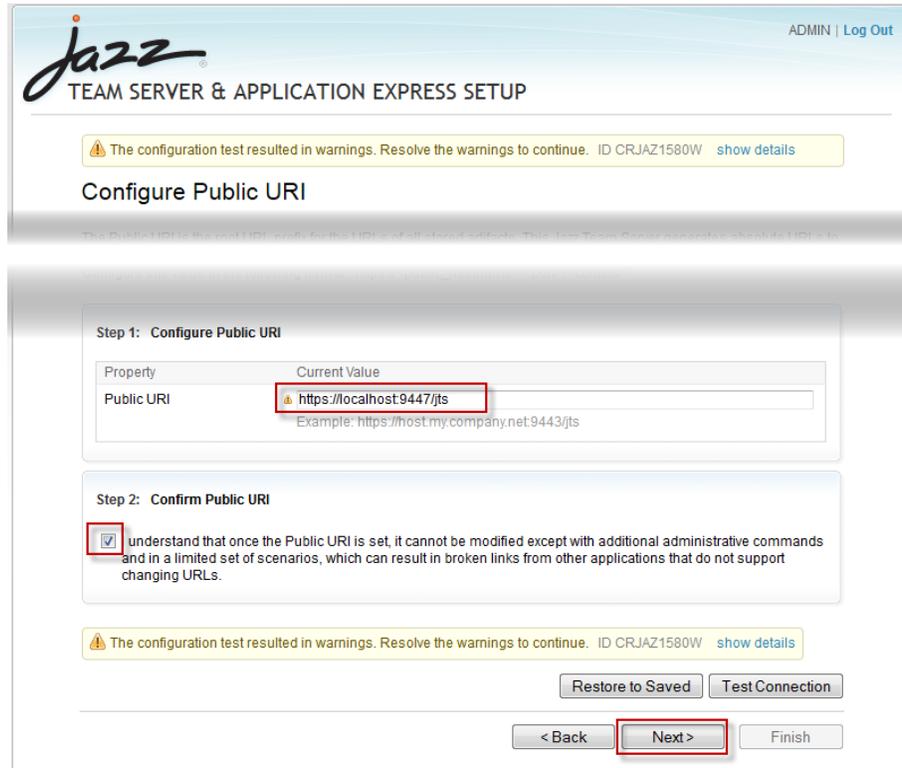
- __a. Start a compatible browser on the machine hosting the shipping server. Open the URL <https://localhost:9447/jts/setup>.

- __b. Login with the user **ADMIN** with password **ADMIN**



- __c. Select **Express Setup** and click **Next** to continue.

- __d. Check and confirm the public URI <https://localhost:9447/jts/>. It is safe to use this unusual public URI in our case, since the RTC client runs on the same machine.



- __e. Press **Next** to continue.
- __f. Enter the information for the administrator user.

Use your own administrator user.

Use your own administrator and shipping user. replace the values in the description with your values.

A good choice would be a user that has administrative rights on the production servers too, or at least access to the streams that need to be synchronized.

- __g. For this scenario we use **shippingadmin** as *User ID, Name, Password* and **shippingadmin@example.com** as *e-mail*. You can use a more meaningful user Id and Password. In this case replace the information in this instruction with your values. Press **Next** to continue.
- __h. Let the setup run and make sure to check it finalizes successful. Press **Next** to continue.

- __i. Assign one of the **Team Concert Developer** Trial licenses to your administrator. Press Finish to finish the setup.
- __j. Check the Diagnostics to verify the setup performed without errors. You might see the typical error that the system clock is not synchronized. You can ignore this in the current setup.
- __k. Optional: Go to the License Key Management administration page <https://localhost:9447/jts/admin#action=com.ibm.team.repository.admin.manageLicenses> and upload valid Rational Team Concert Developer licenses.

For a test you can run with the Evaluation Licenses



If you run a test or evaluation, you can keep working with the evaluation licenses until they expire.

Please note: You can not use the 10 Free or Developer for Workgroup licenses, as they disable the distributed SCM capabilities.

- __6. You have just successfully set up your shipping server.

2.3.1 Create User Account

You will now create the shipping user account used for distributed SCM on this server.

Shipping User Account

 You can use different user accounts for the shipping server and the production server, as long as the users have the necessary permissions on the server they are used.

Reuse Shippingadmin as Shipping User

 You can also use the user **shippingadmin** instead of a dedicated shipping user e.g. to save licenses. In this case replace the information for your shipping user in the instructions below.

LDAP

 If possible, use Tomcat Authentication instead of LDAP for the shipping server.

The production servers might use LDAP for Authentication. It is possible to configure the shipping server for LDAP, however this makes things more complicated especially with different LDAP servers on different sites. It would also be necessary to adjust the backup and restore strategy to for the server.xml and the web.xml files of the applications.

Floating Licenses

 It is possible to use floating licenses for the shipping users. In this case you need to manage the floating server on each site.

- __7. Start your shipping server, if it is not already started.
- __8. Open <https://localhost:9447/jts/admin#action=com.ibm.team.repository.manageUsers> and log in with your administrative account. In the example use **shippingadmin** as user ID and password.
- __9. Select [Create User](#) to create a new user.

- __a. Provide the user Name, ID, e-mail and Password for your **site A** shipping user.
 - __b. In the example use **shippinguser** as *User ID, Name, Password* and **shippinguser@example.com** as *e-mail*.
 - __c. Assign a **Rational Team Concert Developer License** to the user.
 - __d. Save your new user.
- __10. Open <https://localhost:9447/jts/admin#action=com.ibm.team.repository.manageUsers> and check if you see the new user.
- __11. You have now created a shipping user to be used with the shipping server.

2.3.2 Configure the Server for Distributed SCM

By default distributed SCM is disabled on the servers, you need to enable it on all servers, before it can be used. This is only needed to be done once.

- __12. Start your shipping server, if it is not already started.
- __13. Open the CCM Application Administration page <https://localhost:9447/ccm/admin#action=jazz.viewPage&id=com.ibm.team.repository.server>. Log in with your administrative account. In the example use **shippingadmin** as user ID and password.
- __14. Navigate to the Advanced Properties of the server. They can be found at this link: <https://localhost:9447/ccm/admin#action=com.ibm.team.repository.admin.configureAdvanced>
- __15. Search the section `com.ibm.team.scm.service.internal.tasks.InternalScmService`. Scroll down in this section and locate “Enable Distributed SCM”.
 - __a. Set the value for Enable Distributed SCM to True

com.ibm.team.scm.service.internal.tasks.InternalScmService	
Property	Current Value
Default size of era in change history	20
Enable Distributed SCM	true
Enable SCM service logging	false
Enable change history validation per transaction	true

- __b. **Save** your change

2.3.3 Create a Project Area

To be able to ship changes it is necessary to have at least one project on the shipping server that hosts the repository workspaces or streams used to ship. This is only needed to be done once. It is also possible to use several project areas for this purpose, in which case you have to perform step Create a Project Area and Configure the Project Area for Distributed SCM for all those projects.

- __16. Start your shipping server, if it is not already started.
- __17. Open the CCM Application Project Area Administration Page <https://localhost:9447/ccm/admin#action=com.ibm.team.process.manageProjectAreas> and Log in with your administrative account if required. In the example use **shippingadmin** as user ID and password.
- __18. Select [Create Project Area](#).
 - __a. Deploy the process templates. This is only necessary the first time you create a project area.
 - __b. Name the Project Area **Shipping**.
 - __c. Select **Scrum** as the process. You can use any process except the unconfigured process, since we will only use this project area to store SCM data. If you choose a different process, you might have to do additional steps to configure roles and permissions.
 - __d. Click **Save** to create the project area

2.3.4 Configure the Project Area for Distributed SCM

To be able to ship changes a special permission is required. The best approach is to configure a special role to give the required permission to the users that are supposed to ship changes between servers.

- __19. Start your shipping server, if it is not already started.
- __20. Open the CCM Application Project Area Administration Page <https://localhost:9447/ccm/admin#action=com.ibm.team.process.manageProjectAreas> and if asked to, log in with your administrative account if required, using **shippingadmin** as user ID and password.

- __21. Select and open the project area **Shipping** for administration.
- __a. In the left column select the **Roles** administration section

The screenshot shows the 'Active Project Areas >' section with a search bar containing 'Shipping'. Below the search bar is a navigation menu with the following items: Overview, Timelines, Roles (highlighted with a red box), Permissions, Iteration Types, Access Control, Categories, Mail Templates, and Process Description. The main content area is titled 'Roles' and contains a description: 'Each project area and each team area can define a set of role declared and in all child areas. Roles defined in the project area can be assigned in any team area. Roles defined in a team area can be assigned in any child team. The ordering of roles in this section determines how roles are assigned to team members. The ordering of roles does not affect the process runtime.' Below the description is a 'Defined Roles' list with a '+' icon (highlighted with a red box) and a '-' icon. The list contains: Product Owner, Scrum Master, Team Member, and Stakeholder.

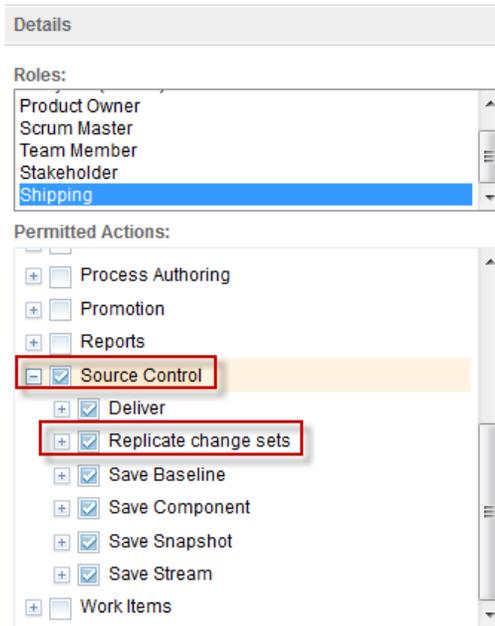
- __b. Use the **Role Add** Button to add a new role.
- __i. Provide **Shipping** as *Identifier* and *Name* for the new Role.
- __ii. Save the change to the process area configuration.
- __22. Select the **Permissions** administration section.

- __a. Under Permissions select **Team Configuration**.
- __b. In the details select the new role **Shipping**.

The screenshot shows the 'Shipping' project area with a 'Save' button. The navigation menu includes: Overview, Timelines, Roles, Permissions (highlighted with a red box), Iteration Types, Access Control, Categories, Mail Templates, and Process Description. The main content area is titled 'Permissions' and is divided into 'Configuration' and 'Details' tabs. The 'Configuration' tab shows a tree view with 'Team Configuration' (highlighted with a red box) selected. The 'Details' tab shows a list of roles: Product Owner, Scrum Master, Team Member, Stakeholder, and Shipping (highlighted with a red box). Below the roles list is a 'Permitted Actions' section.

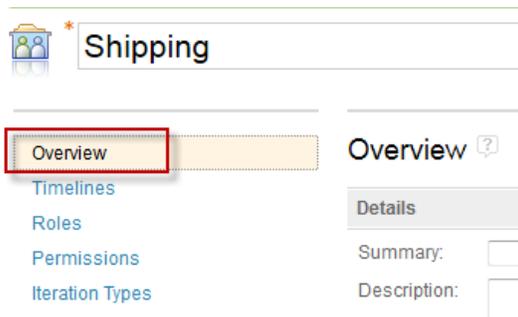
- __c. In the Permitted Actions editor below, scroll down to **Source Control** and expand that section.

- __d. The shipping user needs to have the Permission to **Replicate Change Sets** to use distributed SCM. Select the **Source Control checkbox** to provide this permission along with all other source control permissions to the Shipping role. If necessary you can only permit the required actions.



- __e. Save the change to the process area configuration.

- __23. Select the **Overview** administration section.



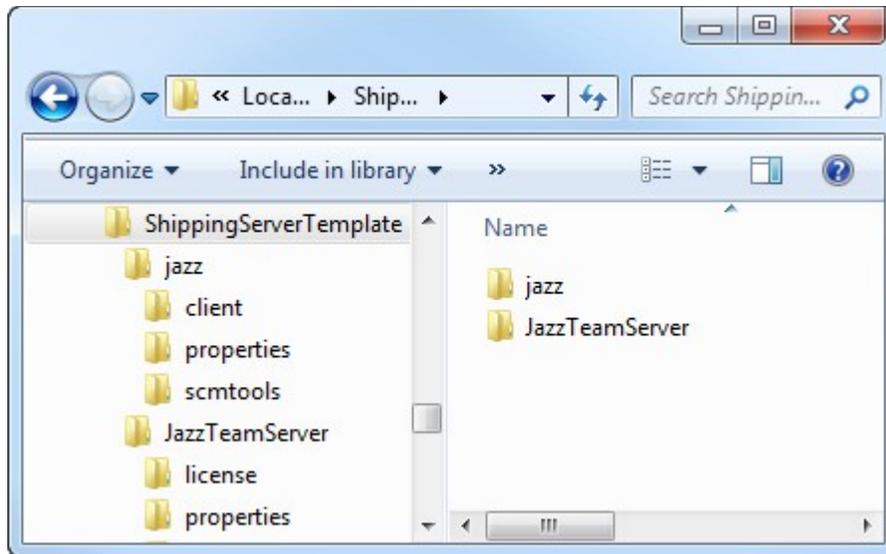
- __a. Add the user **shippinguser** to the **Members** of the project area and assign the roles **Shipping** and **Team Member** to the user.
- __b. Save the change to the project area configuration.

- __24. You have now configured your Shipping project for distributed SCM.

2.4 Download and install the Eclipse client used for shipping

The next steps describe how to install the Eclipse client used for shipping.

- __25. Download the ZIP Version of the Client for Eclipse IDE from the Rational Team Concert All Downloads Page
- __26. Unzip the downloaded Zip version of the Client for Eclipse IDE to the folder **<ShippingServerFolder>**
- __27. Review the **<ShippingServerFolder>** structure.
 - __a. Your **<ShippingServerFolder>** should now look like below.

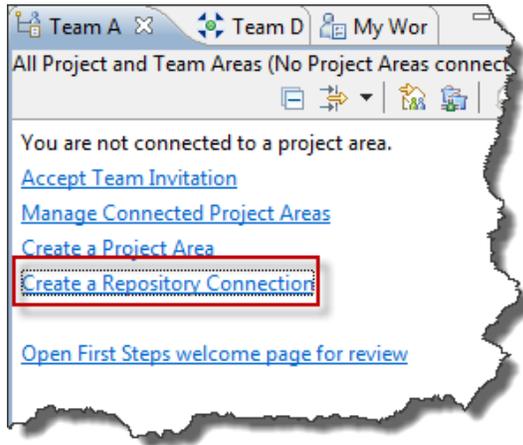


- __b. The RTC Eclipse client is installed in the **<ShippingServerFolder>jazz** folder and its subfolders. The Server is installed in **<ShippingServerFolder>JazzTeamServer**.

2.4.1 Connect the Eclipse Client to the Shipping Server

- __28. Start your shipping server, if it is not already started.
- __29. Start the Eclipse Client in **<ShippingServerFolder>jazz\client\eclipse\eclipse.exe**.
 - __a. If prompted for a workspace choose **<ShippingServerFolder>Workspaces\shippinguser**
- __30. Close the entry page.
- __31. Open the Team Artifacts View
- __32. To create a Repository Connection to your **shipping server**.
 - __a. On the Team Artifacts View

- __i. Click on the Link **Create a Repository connection** or (Right click on Repository connections and select New>Repository Connection).



- __ii. Use `https://localhost:9447/ccm` as connection URI.
 - __iii. Enter the *user id* and the *password* for the **shippinguser**.
 - __iv. Click Apply and connect to the server.
- __b. Right click on the connection and select Manage Connected Project Areas
 - __i. Connect to the project area **Shipping** on the shipping server.
- __33. You are now connected to your shipping server.

2.5 Back Up your Shipping Server Template

To be able to use the shipping server template later for creation of new shipping servers, the best option is to store the template in a compressed format. The way everything was set up allows you to basically put everything in an archive and later recreate all tools as needed for a shipping server.

Installation Manager Setup



If you use Installation Manager instead of the ZIP version of RTC it is not proven that you can zip and reuse the shipping server template, because Installation Manager uses shared data across install on the same machine. See information at the beginning of this chapter.

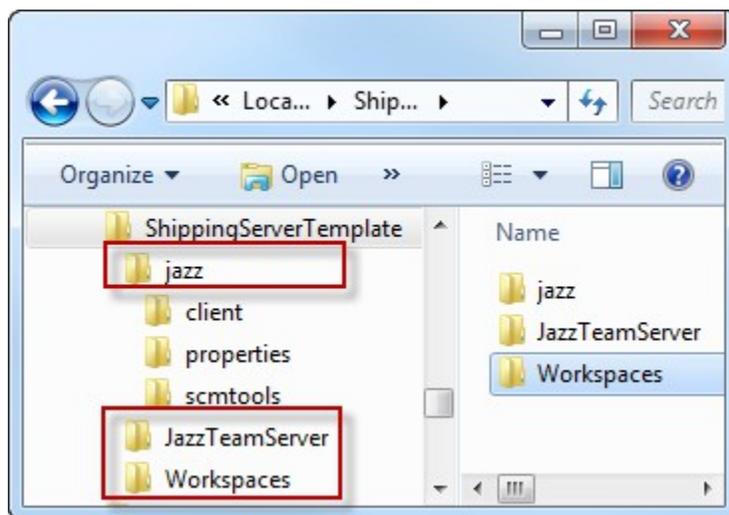
- __34. Close the Shipping Server Eclipse Client.
- __35. Shut down the shipping server by using the **<ShippingServerFolder>\JazzTeamServer\server\server.shutdown.bat** file.

Make sure the server is successfully shut down!



Before continuing, you have to make sure the server processes and the client are successfully shut down. Check the running processes. If you fail to, your result of this step could end up corrupted.

- __36. Your Shipping Server template install folder **<ShippingServerFolder>** now looks like below:



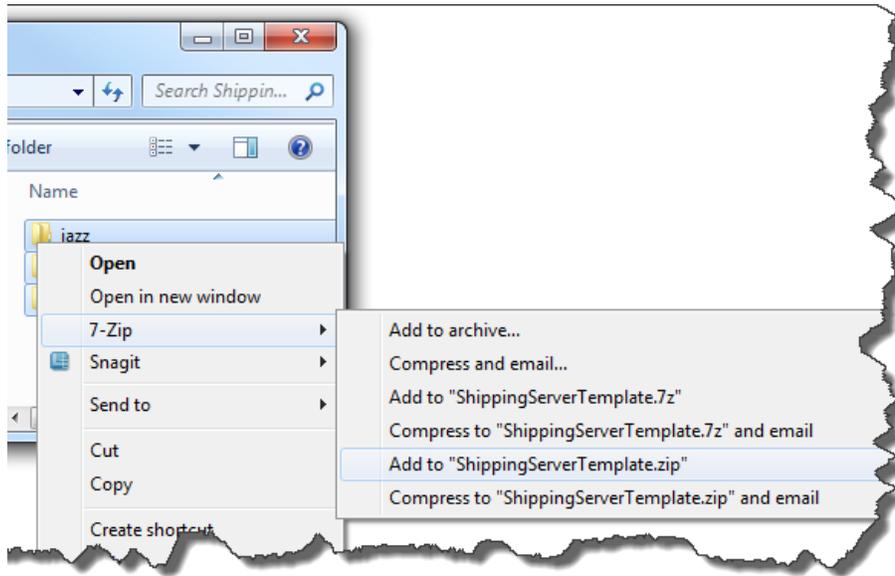
- __a. The folder jazz contains the shipping client instance.

__b. The folder JazzTeamServer contains the RTC shipping server instance.

__c. The Workspaces Folder contains the shipping user workspace instance

__37. Create a ZIP file containing these files.

__a. If you use 7zip, select all files contained in the **<ShippingServerFolder>** and add them to a new archive. Name the archive ShippingServerTemplate.zip



__38. Back up your shipping server template archive ShippingServerTemplate.zip.

Create a New Shipping Server With Just One Unzip



You can now create a new shipping server instance by just unzipping the archive ShippingServerTemplate.zip into an empty folder.

You can ship the shipping server template archive to other sites and create new shipping server instances there using the same method.

__39. You have now successfully created a shipping server template.

__40. You can delete the Shipping Server Template folder, once you moved the template out of it.

2.6 Create a Dedicated Shipping Server Instance

To ship changes from the site A production server Prod_A to the site B production server Prod_B we need a shipping OUT server at the site (e.g. A). The following steps describe how to create a shipping server. We will refer to the folder again as **<ShippingServerFolder>**.



Adjust Steps for the Shipping Server Instance

If using the instructions below to create an arbitrary shipping server replace **<ShippingServerFolder>** with the desired name of the shipping server.

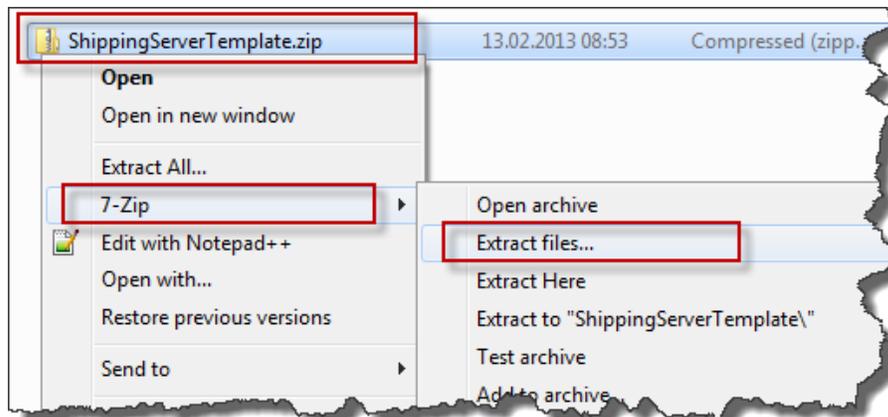
- __41. On the machine used to host the shipping server create a new **<ShippingServerFolder>** folder for the shipping server instance. To create the Shipping OUT Server for Prod_A name the folder for example **C:\ShippingServer_A_OUT**.
- __42. Unzip the Shipping Server Template archive into the Shipping Server Folder.



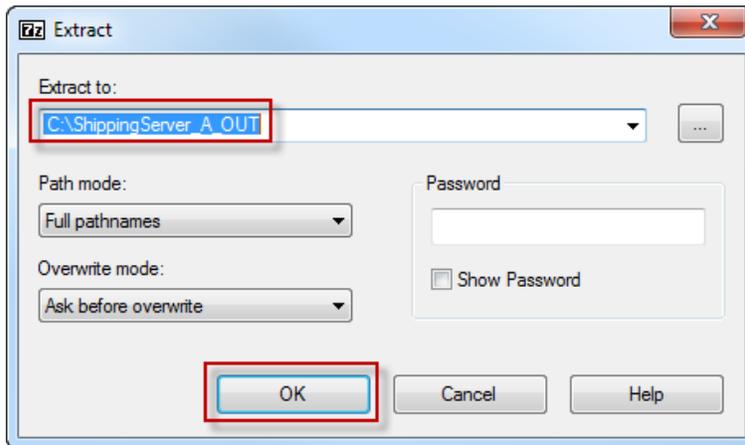
Alternatively use the Commandline

It is also possible to use the commandline to do the following steps.

- __43. Select the Shipping Server Template Archive, in the context menu select **Extract Files**.



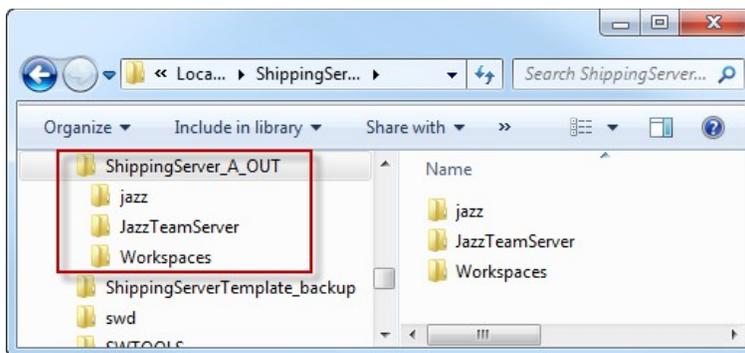
__44. Select or enter the **<ShippingServerFolder>** as extract destination.



__45. Press **OK** to start the extraction.

__46. Wait for the extraction process to finish.

__47. Open the **<ShippingServerFolder>**. The folder should look like below:



You have now a new Shipping Server Instance.

Lab 3 Using Distributed SCM for Shipping

This Lab explains, how you can use distributed SCM to replicate change sets to another server. The replicated data can then be used with a shipping server to bring it over to another, disconnected site. If you are only interested in how distributed SCM works, you can stop after replicating the change sets. However, you probably want to look at the article [Flow changes cross repositories with Rational Team Concert](#) for a description on how to be able to navigate to the work items associated to replicated change sets.

3.1 Configure the Production Server For Distributed SCM

You need to configure your production server for distributed SCM like you configured the shipping server. This section describes the steps for production server **Prod-A**. The same steps will be performed for the production server Prod-B in **Lab 4**.

3.1.1 Create an Example Production Server For Site A

The workshop assumes you have a production server set up already. If you don't have a production server, you can set up an example production.

- __48. Follow **2.1 Create A Shipping Server Template** and set up a server, with the following differences:
 - __a. Use a different **<ShippingServerFolder>** folder for example [C:\ProdA](#) and extract to C:\Prod_A\JazzTeamServer.
 - __b. Choose a different administrator e.g. **myadmin_a**.
 - __c. For the Site A Production server, if the default ports are not yet used, skip **2.2 Configure the Application Server**, otherwise follow that section and change the ports, but choose ports different from the ones used by the shipping server.
 - __d. Use a different a **public URI** such as **https://prod-a:9443/**. Please note, if you had to change the ports, you need to pick the correct port in the public URI.

Host Name and Public URI

To use a different host name, you either need to run on a different machine with a valid host name, or you have to set up an alias in the hosts file.



On Windows the hosts file is

C:\Windows\System32\drivers\etc\hosts

You can simply add a alias by copying the localhost line and changing localhost to the desired name.

This [Wikipedia](#) entry explains the characters allowed in host names.

- __e. Create an example project containing code you can ship.
- __i. Open <https://prod-a:9443/admin/web/sample> (change the public URI part to match your public URI).
- __ii. Create the sample project Money That Matters, also referred as JKA Banking.

3.1.2 Enable Distributed SCM on the Production Server Site A

This step needs to be only done once.

- __49. Start your Production Server, if it is not already started.
 - __a. Log in using an administrative account.
 - __b. Enable Distributed SCM on this server as described in **2.3.2 Configure the Server for Distributed SCM**.

3.1.3 Create a Shipping User and Configure the Project for Shipping

Shipping User on the Production Server



The shipping user on the production server can have any ID and password.

You can reuse an existing user and configure this user for distributed SCM.

- __50. Pick a user on the production server or create a new shipping user account on your production server.

In the example below we will use **shippinguser_a** as a new, dedicated shipping user. If you use LADP, create the user in LDAP, and import the user to the production server.

Follow the instructions in **2.3.1 Create User Account** and replace **shippinguser** with **shippinguser_a**. Make sure to add a Rational Team Concert Developer License to this user as described

- __51. For the projects you want to ship changes to or from you need to define a shipping user, if this user is not yet a member of the project.
- __a. Open the project area administration as described in **2.3.4 Configure the Project Area for Distributed SCM**.
 - __b. Create the **Shipping** role in all of these projects as described in **2.3.4 Configure the Project Area for Distributed SCM**.
 - __c. Add the shipping user **shippinguser_a**. to the Members of the project area and add the required **Shipping** role and other roles required to access SCM as described in **2.3.4 Configure the Project Area for Distributed SCM**.

Handle access restrictions to streams



If access to the streams and visibility of streams is restricted to teams, you have to add the shipping user to the teams required to access the streams and provide the roles, including the **Shipping** role, required to access, read and write the stream you want to ship.

- __52. You have now successfully configured your production server for distributed SCM.

3.2 Create a Dedicated Shipping Server Instance

To ship changes from the site A production server Prod-A to the site B production server Prod-B we need a shipping OUT server at site A. The following steps describe how to create a shipping server. We will refer to the folder again as **<ShippingServerFolder>**.

Adjust Steps for the Shipping Server Instance



If using the instructions below to create an arbitrary shipping server replace **<ShippingServerFolder>** with the desired name of the shipping server.

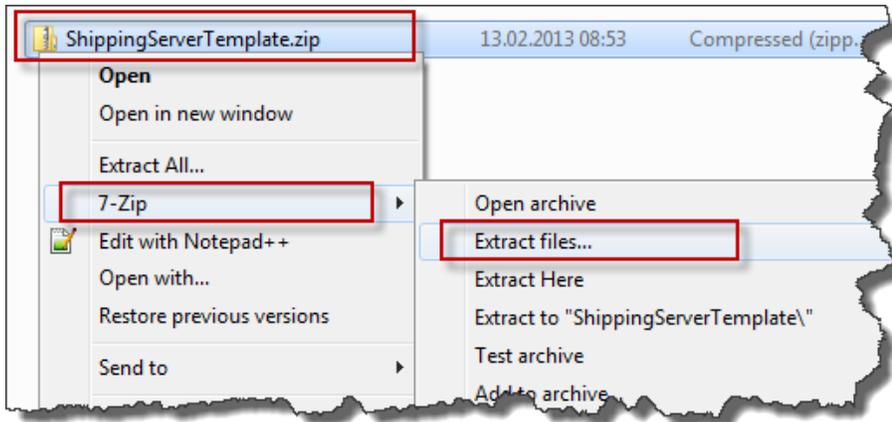
__53. On the machine used to host the shipping server create a new **<ShippingServerFolder>** folder for the shipping server instance. To create the Shipping OUT Server for Prod-A name the folder for example **C:\ShippingServer_A_OUT**.

__54. Unzip the Shipping Server Template archive into the Shipping Server Folder.

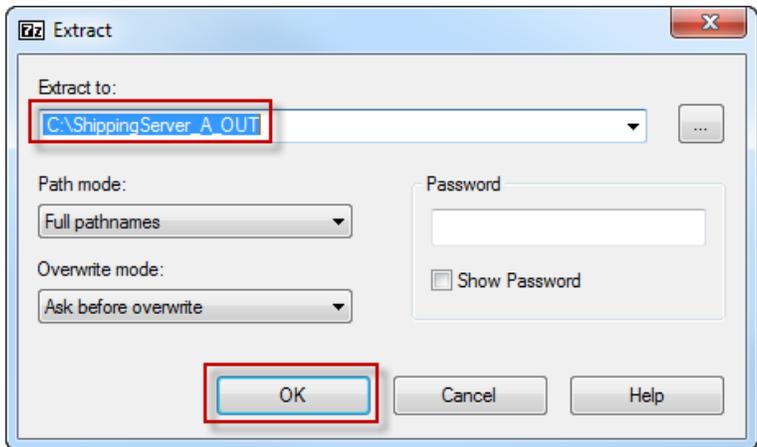
Alternatively Use the Commandline

 It is also possible to use the commandline to do the following steps.

__a. Select the Shipping Server Template Archive, in the context menu select **Extract Files**.



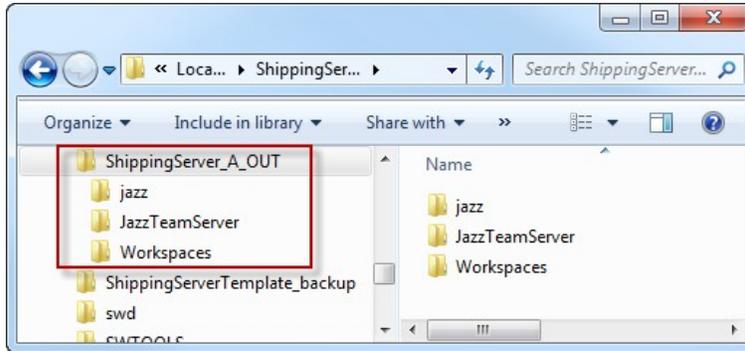
__b. Select or enter the **<ShippingServerFolder>** as extract destination.



__c. Press **OK** to start the extraction.

__d. Wait for the extraction process to finish.

__55. Open the **<ShippingServerFolder>**. The folder should look like below:



__56. You have now a new Shipping Server Instance.

3.3 Transfer the SCM Changes to the Shipping Server

3.3.1 Start Your Shipping Server

__57. If you have another shipping server running, shut down that shipping server.

__58. Use **<ShippingServerFolder>\JazzTeamServer\server\server.startup.bat** or to start the Shipping Server.

Run Only a Single Shipping Server



You can only run one shipping server on one machine at any time. Make sure to shutdown any other shipping server instance on this site.

__a. Monitor the startup for conflicting ports. In case you see conflicts another shipping server is running.

3.3.2 Connect The Eclipse Client to the Production Server

__59. Start the Eclipse Client in **<ShippingServerFolder>\jazz\client\eclipse\eclipse.exe**.

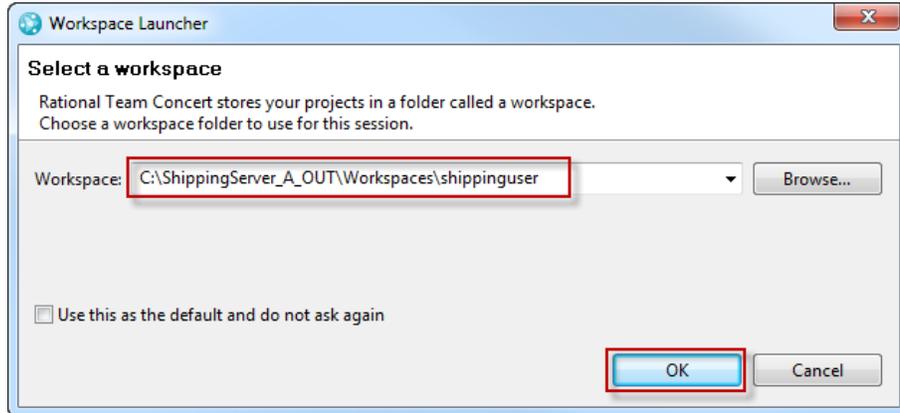
__a. If prompted for a workspace

Pick the Correct Workspace



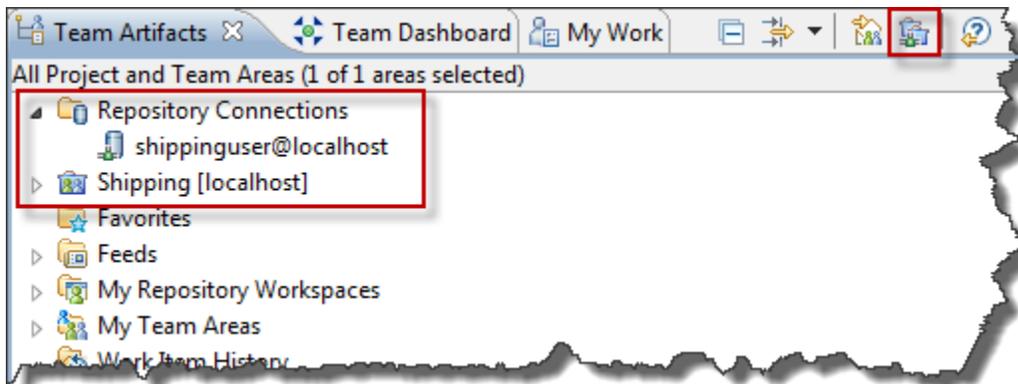
Please make sure to pick the correct Eclipse workspace. You want to use the the workspace for this Shipping Server instance and not for another one. By default the workspace would point to the shipping server template.

- __b. Browse to the workspace, choose **<ShippingServerFolder>\Workspaces\shippinguser** and press OK.



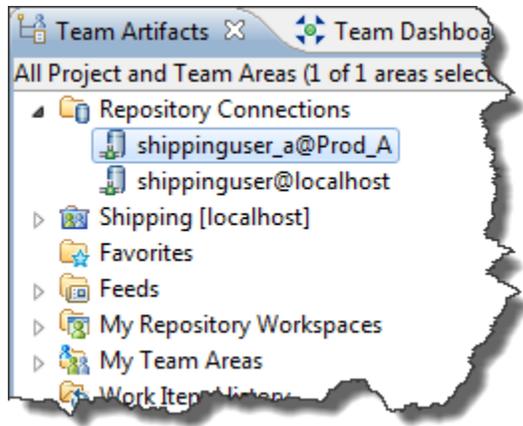
- __c. You should not see a blank entry page, If you do, check if you picked the right workspace.
- __d. The client should connect to the shipping server. If not, check if you picked the right workspace and check if the shipping server is up and running.

- __60. Open the Team Artifacts View, you should see at least the repository connection to the shipping server.



- __61. Create a Repository Connection to your **Production Server**.
 - __a. On the Team Artifacts View
 - __i. Right click on Repository connections and select New>Repository Connection.
 - __ii. For Prod-A use <https://prod-a:9443/ccm> as connection URI.
 - __iii. Enter the *user id* and the *password* for the **shippinguser_a** used for this server.
 - __iv. Click **Apply** and connect to the server.
 - __b. Right click on the new connection and select **Manage Connected Project Areas**

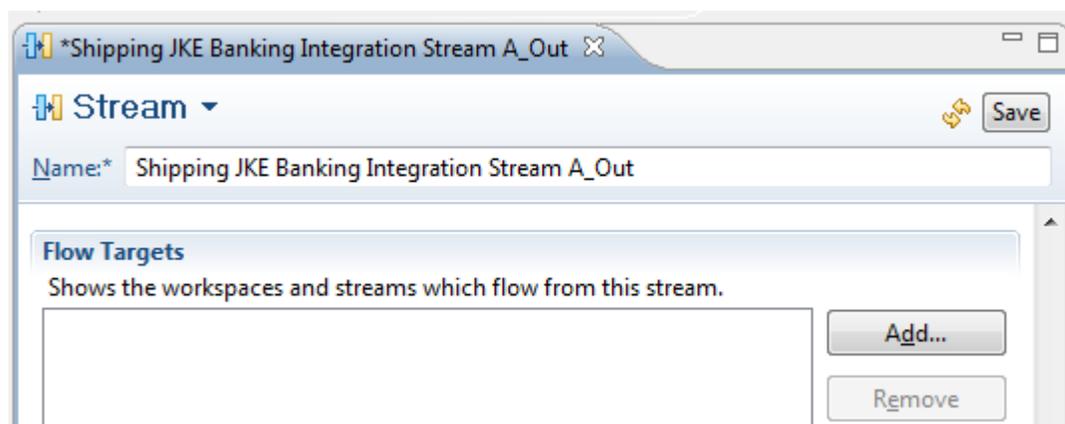
- __i. Connect to the project areas on the production server you need to ship from.
- __62. You are now connected to the production server and to the shipping server. Your Team Artifacts View should look like below. Note the different user ID used for the connections.



3.3.3 Create a Shipping Stream

You can skip this step if you already have created shipping streams

- __63. Select the stream you want to ship to another server.
 - __a. Right click on the stream (if you use the JKE Banking sample pick th JKA Banking Integration Stream) and select **Duplicate**
 - __b. In the Stream Editor rename the Stream to Shipping <Original Stream Name> <Site>_<Direction>, where <Site> is an identifier for the current site and <Direction> ins In or Out. In the example use A_Out as the postfix.



- __c. Save the name change.
- __64. Optional: Manage Flow Targets for documentation
- __a. For documentation purposes you might want to manage the flow targets of the streams.

- __i. For Shipping Out Streams, clear the Flow Targets.
- __ii. Set the Shipping Out Streams as Flow Target in the original stream
- __iii. For Shipping In Streams, clear the Flow Targets.
- __iv. Set the original stream as Flow Target of the Shipping In Streams

3.3.4 Accept SCM Change Sets to the Shipping Server

To replicate the change sets required for shipping from one stream, you need to perform the following activities.

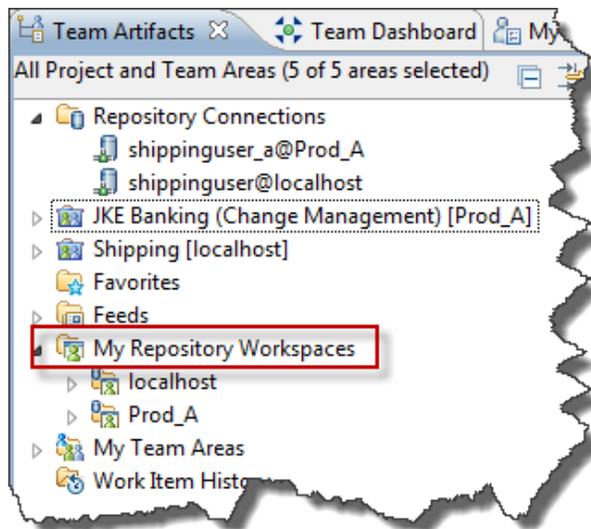
- __65. Create a new Repository Workspace for a stream to ship changes.

Reuse Existing Repository Workspaces



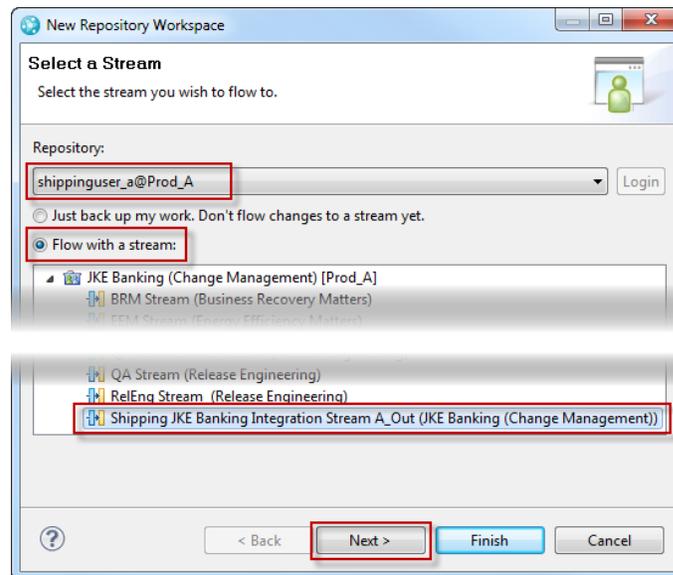
If you have a shipping out repository workspace created in a prior shipping, you should reuse this repository workspace. In this case you only have to look at the repository workspace in the pending changes view and accept changes from the shipping out stream.

- __a. In the Team Artifacts View select the My Repository Workspace section.
- __b. Right click the node and select **New>Repository Workspace....**

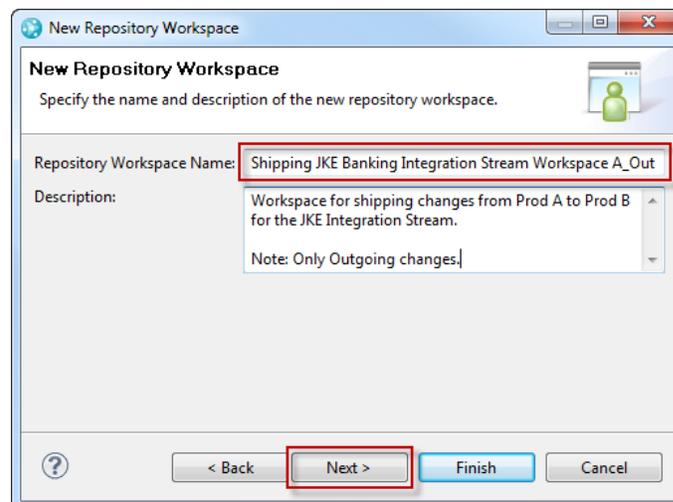


- __c. In the upcoming Repository Workspace Creation dialog

- __i. In the **Repository** drop down box, select the Production Server you are connected to.
- __ii. Select Flow with a Stream to indicate you want to flow with a stream in the other repository.
- __iii. Select the Project Area and Stream from which you want to ship SCM changes sets Out.



- __iv. Press **Next**.
- __v. Provide a useful name and description for the repository workspace. For example add a prefix Shipping and Postfix similar to the stream name A_Out.

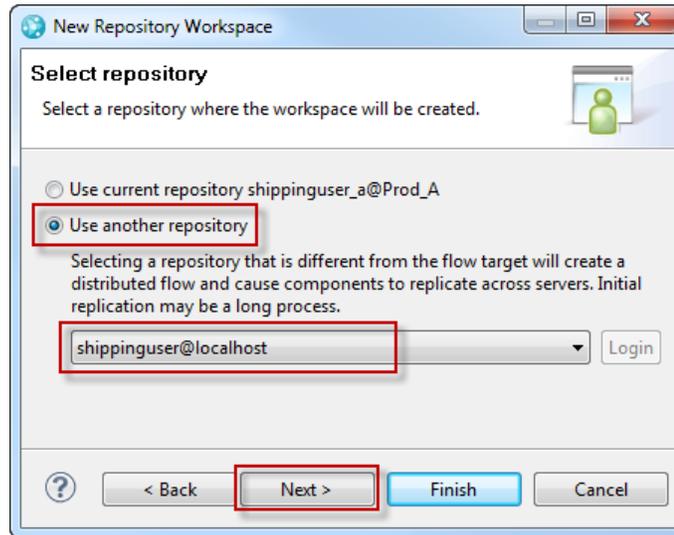


- __vi. Press **Next**.

__vii. The following dialog determines where the repository workspace is stored. By default is is stored in the repository of the stream.

__a. Select the radio button **Use another repository**.

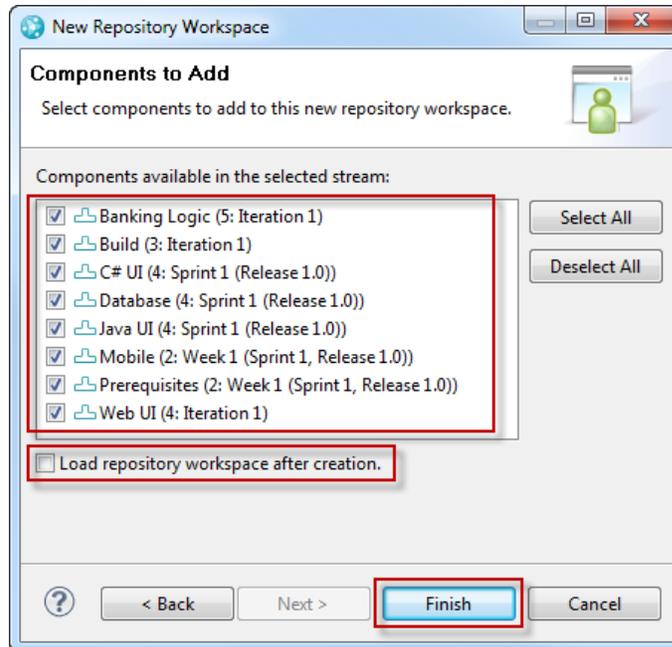
__b. Select the **shipping server** repository as storage location for the repository workspace.



__c. Press **Next** again.

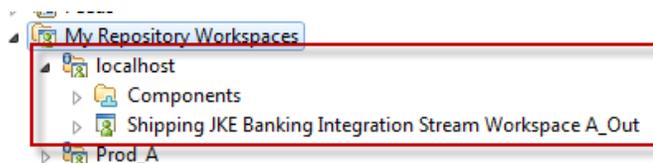
__viii. On the **Read Access Permissions** dialog select **Public** and press **Next** again.

- __ix. On the **Components to Add** dialog make sure all components you want to ship are added and de-select the **Load repository workspace after creation** check box as it is not necessary to load the workspace. You can load the workspace to be sure the changes are transferred. In this case unload the workspace later.

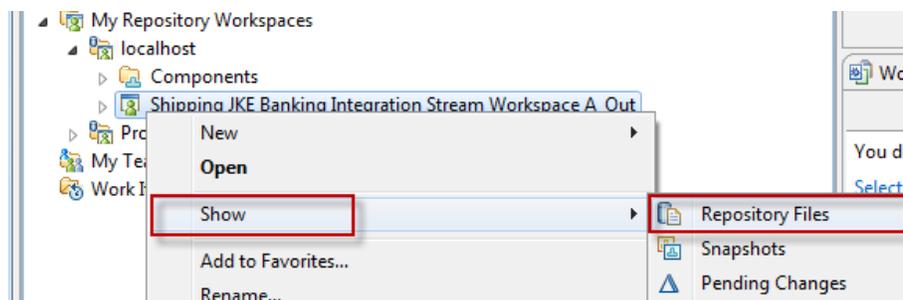


- __x. Press **Finish** and wait for the operation to finish.

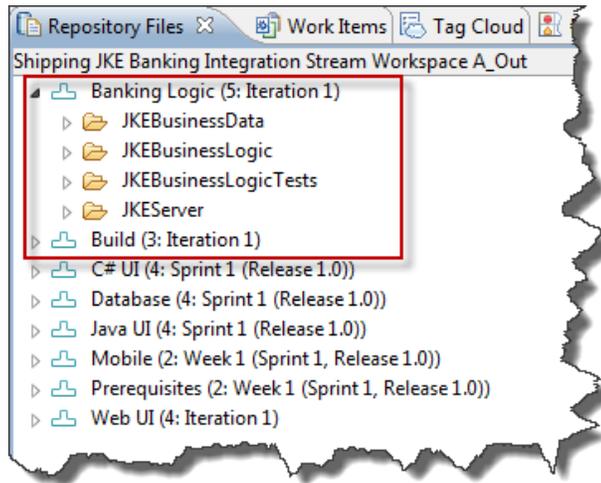
- __d. The changes are transferred to the shipping server repository during the creation of the repository workspace.
- __e. Check that the repository workspace appears in the right repository under your shipping server repository **localhost**.



- __f. Check the content is available by looking at the repository files



- __g. You should be able to see components and files like shown below. If not, try to load the workspace and check your configuration.



- __66. You have now successfully replicated the change sets to the shipping server. You can repeat the steps above for other streams that need to be shipped.
- __67. In subsequent shipping operations, you can reuse the repository workspace you created in this step.

3.4 Backup the Shipping Server for Shipping

The shipping server repository now contains all the changes for the stream to ship. You can now ship these changes to another site, basically by creating a backup of the repository and the configuration.

3.4.1 Create a Partial Backup of the Shipping Server

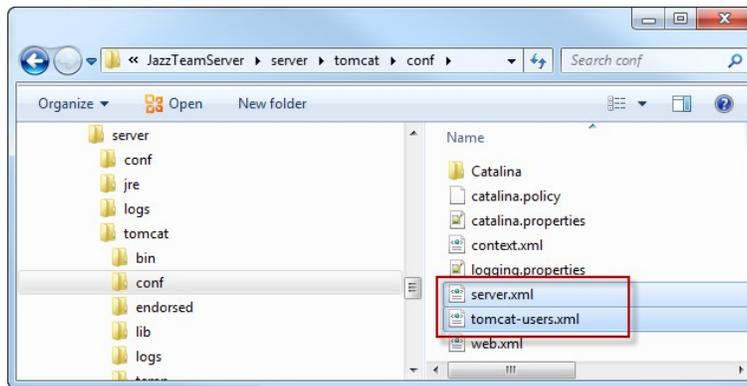
The article [Backup the Rational solution for Collaborative Lifecycle Management](#) talks about what needs to be backed up to restore a server. With only JTS and CCM installed, the scenario is slightly simpler.

It is easy enough to automate the process described below using a script.

To backup the shipping server the server needs to be shut down.

- __68. Shut down the shipping server by using the `<ShippingServerFolder>\JazzTeamServer\server\server.shutdown.bat` file.
- __69. Create a main backup folder. Delete the folder if it already exists.
- __a. Create a folder `C:\BackupShippingServer\`
 - __b. Create a folder `C:\BackupShippingServer\JazzTeamServer`
 - __c. Create a folder `C:\BackupShippingServer\JazzTeamServer\server`
- __70. Backup Tomcat information

- __a. Create a tomcat backup folder
 - __i. Create a Folder C:\BackupShippingServer\JazzTeamServer\server\tomcat\
 - __ii. Create a Folder C:\BackupShippingServer\JazzTeamServer\server\tomcat\conf
- __b. Copy the file C:\ShippingServer_A_OUT\JazzTeamServer\server\tomcat\conf\server.xml to C:\BackupShippingServer\JazzTeamServer\server\tomcat\conf
- __c. Copy the file C:\ShippingServer_A_OUT\JazzTeamServer\server\tomcat\conf\tomcat-users.xml to C:\BackupShippingServer\JazzTeamServer\server\tomcat\conf



__71. Backup the Jazz Information

- __a. Create a Configuration backup folder
C:\BackupShippingServer\JazzTeamServer\server\conf
 - __i. Create a JTS backup C:\BackupShippingServer\JazzTeamServer\server\conf\jts
 - __ii. Create an ADMIN backup
C:\BackupShippingServer\JazzTeamServer\server\conf\admin
 - __iii. Create a CCM backup
C:\BackupShippingServer\JazzTeamServer\server\conf\ccm
- __b. Backup the JTS data
 - __i. Copy the folder
C:\ShippingServer_A_OUT\JazzTeamServer\server\conf\jts\derby and all its contents and subfolders and their contents into
C:\BackupShippingServer\JazzTeamServer\server\conf\jts\
 - __ii. Copy the folder
C:\ShippingServer_A_OUT\JazzTeamServer\server\conf\jts\indices and all its contents and subfolders and their contents into
C:\BackupShippingServer\JazzTeamServer\server\conf\jts\

- __iii. Copy the file
C:\ShippingServer_A_OUT\JazzTeamServer\server\conf\jts\teamserver.properties into C:\BackupShippingServer\JazzTeamServer\server\conf\jts\
es
- __c. Backup the ADMIN data
 - __i. Copy the file
C:\ShippingServer_A_OUT\JazzTeamServer\server\conf\admin\admin.properties into C:\BackupShippingServer\JazzTeamServer\server\conf\admin\
es
 - __ii. Copy the file
C:\ShippingServer_A_OUT\JazzTeamServer\server\conf\admin\friends.rdf into
C:\BackupShippingServer\JazzTeamServer\server\conf\admin\
es
- __d. Backup the CCM data
 - __i. Copy the folder
C:\ShippingServer_A_OUT\JazzTeamServer\server\conf\ccm\derby and all its
contents and subfolders and their contents into
C:\BackupShippingServer\JazzTeamServer\server\conf\ccm\
es
 - __ii. Copy the folder
C:\ShippingServer_A_OUT\JazzTeamServer\server\conf\ccm\indices and all its
contents and subfolders and their contents into
C:\BackupShippingServer\JazzTeamServer\server\conf\ccm\
es
 - __iii. Copy the file
C:\ShippingServer_A_OUT\JazzTeamServer\server\conf\ccm\teamserver.properties into C:\BackupShippingServer\JazzTeamServer\server\conf\ccm\
es
- __72. Select folder C:\BackupShippingServer\JazzTeamServer and create a zip file from it similar to 2.5 Back Up your Shipping Server Template.
- __73. Encrypt the zipped data.
- __74. Send the encrypted zip file to Site B

Lab 4 Distributed SCM and Shipping Changes To a Server

This lab explains how you can restore a shipping server on an other site called Site B, that does not have a network connection with the site where the SCM data was replicated to the shipping server.

4.1 Setup the Shipping Server form a Partial Backup

If you want to install and restore a backup on any site, follow the section below that matches how the backup was performed. The instructions are fore Site B, however, if changes are shipped back to Site A the process is the same.

If you have a shipping server already installed on your site, the shipping server needs to be shut sown.

- __75. Shut down other shipping servers by using the **<ShippingServerFolder>\JazzTeamServer\server\server.shutdown.bat** file.
- __76. Install the Shipping Server and the Eclipse Client on a machine at site B. The name of the shipping server corresponding to the Shipping Server A_OUT used here is **Shipping Server B_A_In**. The **<ShippingServerFolder>** is named **C:/ShippingServer B_A_In** in this example.

Reuse a Previously Installed Shipping Server



If you have already installed the shipping server skip this step.

- __a. Install a Shipping IN Server instance named ShippingServer B_A_In using a Shipping Server Template as described in 2.6 Create a Dedicated Shipping Server Instance. Alternatively you can download and install the Jazz Team Server and the Eclipse Client following 2.1 Create A Shipping Server Template.
- __77. Copy the encrypted zip file on your site into **<ShippingServerFolder>**.
- __78. Decrypt the zip file.
- __79. Unzip the Zip file to **<ShippingServerFolder>** (unzip to here). This should expand the files in a way that the files are restored on top of the existing files. Confirm to overwrite and replace all files.
- __80. You just set up the shipping server and restored the backup.

4.2 Configure the Site B Production Server For Distributed SCM

You need to configure your production server for distributed SCM like you configured the shipping server. This section describes the steps for production server Prod-B. The same steps will be performed for the production server Prod-B. You need to perform these steps only once on any site..

4.2.1 Create an Example Production Server for Site B

The workshop assumes you have a production server set up already. If you don't have a production server you can follow the instructions below to create one similar to **3.1.1 Create an Example Production Server For Site A**.

Shutdown Production Server A if working on on hardware



The workshop assumes this is a new hardware, if you try to run the whole workshop on one hardware, make sure to shutdown Production Server A, before performing the next steps.

- __81. Follow **2.1 Create A Shipping Server Template** and set up a server, with the following differences:
- __a. Use a different **<ShippingServerFolder>** folder for example [C:\Prod_B](#) and extract to `C:\Prod_B\JazzTeamServer`.
 - __b. Choose a different administrator e.g. **myadmin_b**.
 - __c. For the Site A Production server, if the default ports are not yet used, skip **2.2 Configure the Application Server**, otherwise follow that section and choose different ports.
 - __d. Use a different a public URI such as **https://prod-b:9443/**. Please note, if you had to change the ports, you need to pick the correct port in the public URI.

Host Name and Public URI

To use a different host name, you either need to run on a different machine with a valid host name, or you have to set up an alias in the hosts file.



On Windows the hosts file is

`C:\Windows\System32\drivers\etc\hosts`

You can simply add a alias by copying the localhost line and changing localhost to the desired name.

This [Wikipedia](#) entry explains the characters allowed in host names.

- __e. Create an example project you can ship to.

- __i. Open <https://prod-b:9443/admin/web/sample> (change the public URI part to match your public URI).
- __ii. Create the sample project Money That Matters, also referred as JKA Banking.

4.2.2 Enable Distributed SCM on the Production Server Site B

This needs to be only done once.

- __82. Start your Production Server, if it is not already started.
- __83. Enable Distributed SCM on this server as described in **2.3.2 Configure the Server for Distributed SCM**.

4.2.3 Create a Shipping User and Configure the Project for Shipping

- __84. If you have no user account that is supposed to be the shipping user, create this account for your production server as described in **2.3.1 Create User Account**.

In the example below we will use **shippinguser_b** as a dedicated new shipping user. If you use LADP, create the user in LDAP, and import the user. Make sure to add a Rational Team Concert Developer License to this user.

- __85. For the Projects you want to ship changes you need to add the shipping user, if this user is not yet member of the project.
 - __a. Open the project area administration and add the Shipping user **shippinguser_b** to the project area Open the project area administration as described in **2.3.4 Configure the Project Area for Distributed SCM**.
 - __b. Create the **Shipping** role in all of these projects as described in **2.3.4 Configure the Project Area for Distributed SCM**.
 - __c. Add the shipping user **shippinguser_b**. to the Members of the project area and add the required **Shipping** role and other roles required to access SCM as described in **2.3.4 Configure the Project Area for Distributed SCM**.
- __86. You have now successfully configured your production server Prod-B for distributed SCM.

4.3 Deliver Your Shipped Changes into the Production Server

To replicate the changes from your shipping server into the production repository, follow these steps. The description is for Prod-B on site B, however it applies to all sites you ship changes back to.

4.3.1 Connect The Eclipse Client to the Production Server and the Shipping Server

Your shipping server needs to be started to perform the following steps.



Run Only a Single Shipping Server

You can only run one shipping server on one machine at any time. Make sure to shutdown any other shipping server instance on this site.

- __87. Start up the shipping server by using the **<ShippingServerFolder>\JazzTeamServer\server\server.startup.bat** file.
- __88. Make sure the Production server is up.
- __89. Start the Eclipse Client in **<ShippingServerFolder>\jazz\client\eclipse\eclipse.exe**.
 - __a. If prompted for a workspace choose **<ShippingServerFolder>\Workspaces\shippinguser**



Pick the Correct Workspace

Please make sure to pick the correct Eclipse workspace. You want to use the the workspace for this Shipping Server instance and not for another one. By default the workspace would point to the shipping server template.

- __b. You should not see a blank entry page, If you do, check if you picked the right workspace.

The client should connect to the shipping server. If not, check if you picked the right workspace and check if the shipping server is up and running.
- __90. Create a Repository Connection to your production server Prod-B.
 - __a. On the Team Artifacts View
 - __i. Right click on Repository connections and select New>Repository Connection.
 - __ii. For Prod-B use <https://prod-b:9443/ccm> as connection URI.
 - __iii. Enter the *user id* and the *password* for the shipping user **shippinguser_b**.

__b. Click **Apply and connect to the server.**

__91. Connect to the project areas on Prod-B

__a. Right click on the connection and select Manage Connected Project Areas

__b. Connect to the project areas on the production server you need to ship to.

__92. Your Eclipse Client is now set up to ship the changes.

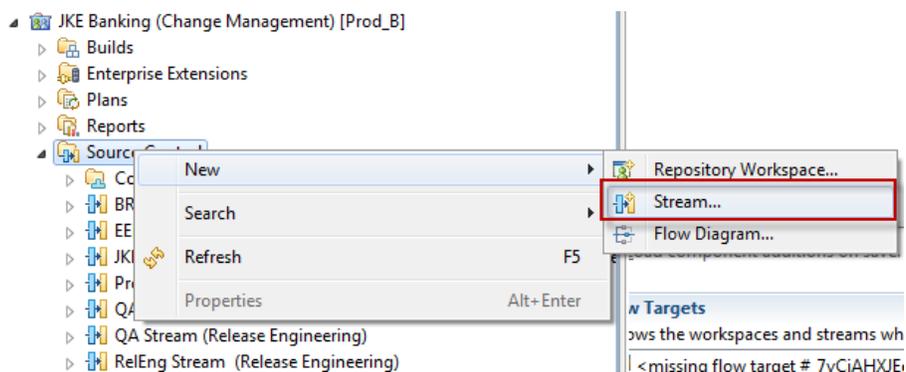
4.3.2 Replicate Your Shipped Changes From The Shipping Repository Workspace into the Production Server Stream

The last step required is to connect your shipping repository workspace to the corresponding stream on the production server on site B and integrate the changes.

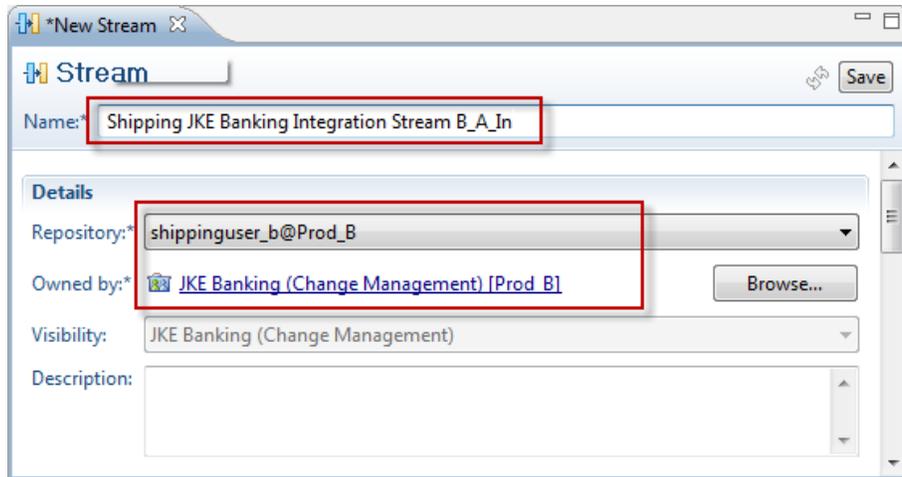
__93. If you have a Shipping In Stream already skip this step and reuse it. Otherwise create a **Shipping IN Stream.**

__a. In the Team Artifacts View select the Source Control node of the project area on the production server.

__b. In the context menu select New>Stream



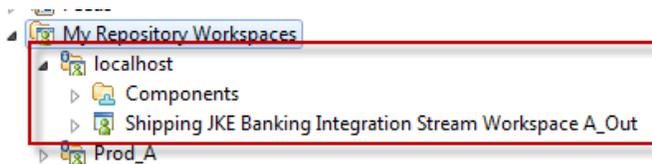
- __c. Name the Shipping In Stream following the naming pattern Shipping <TargetStreamName> <ShipInSite>_<ShipOutSite>_<Direction>, where the two sites describe the local site and the site where the changes originate. As an example name the stream Shipping <StreamName> B_A_In.



- __d. Optional: for documentation set the flow target of the new stream to the target stream where the changes should eventually flow to.
- __e. Save the changes you did.

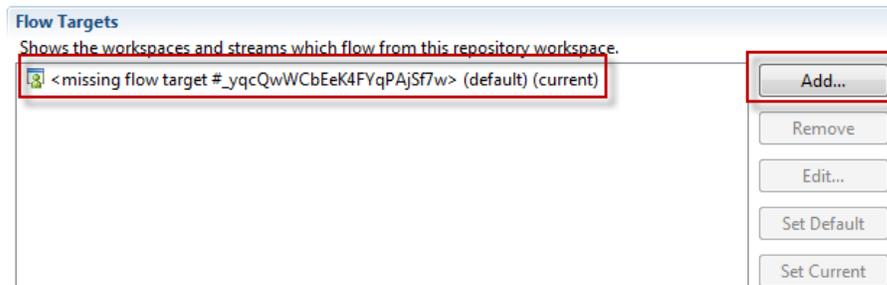
__94. Add a new **Flow Target** to the shipping repository workspace.

- __a. Locate the My Repository Workspaces section in the Team Artifacts view and unfold it.

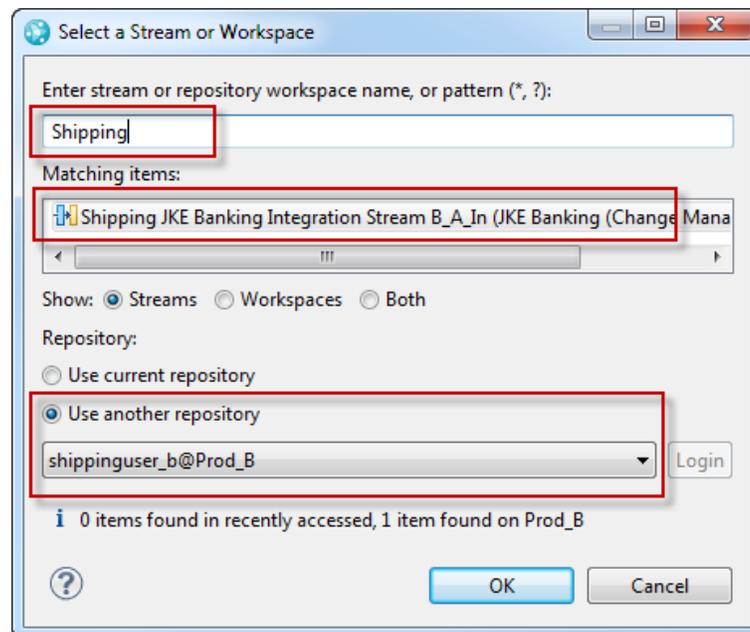


- __b. Locate the repository workspace you created to ship the changes. It should be on this instance of the shipping server and its name starts with **Shipping**.
- __c. Right click the repository workspace and select **Open**.
- __d. In the Repository Workspace editor, scroll down to the Flow Targets section.

- __i. The section should show an unreachable target, since the production server Prod-A is not reachable.

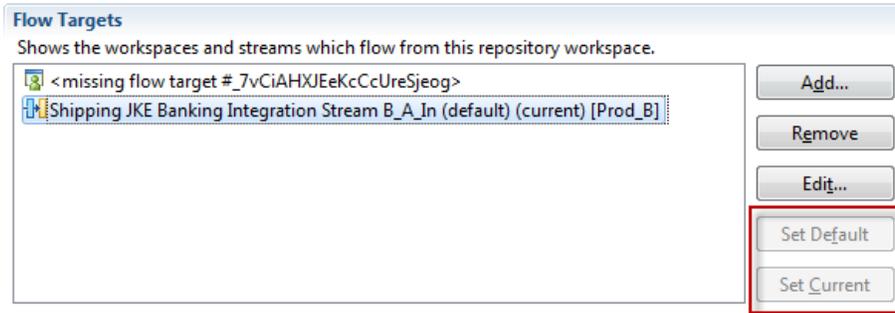


- __ii. Click **Add** to add a new Flow Target
- __iii. Select the corresponding stream in the available Prod-B repository.
 - __a. Select Use Another repository
 - __b. Browse for the production server repository and select it in the drop down box.
 - __c. Search for the corresponding stream you just created and select it in the Matching Items



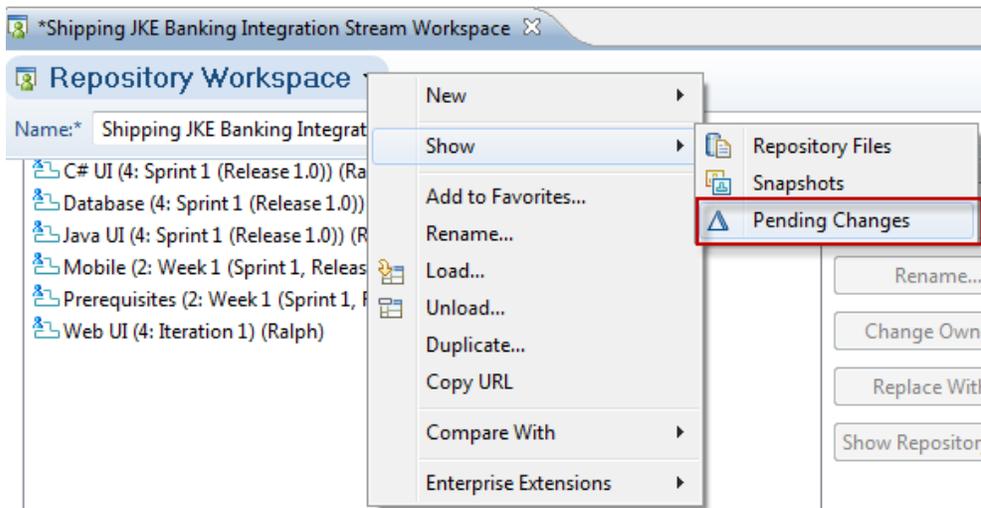
- __d. Click OK to add the new Flow Target.

- __e. On the repository workspace editor select the new flow target as default and current flow target.

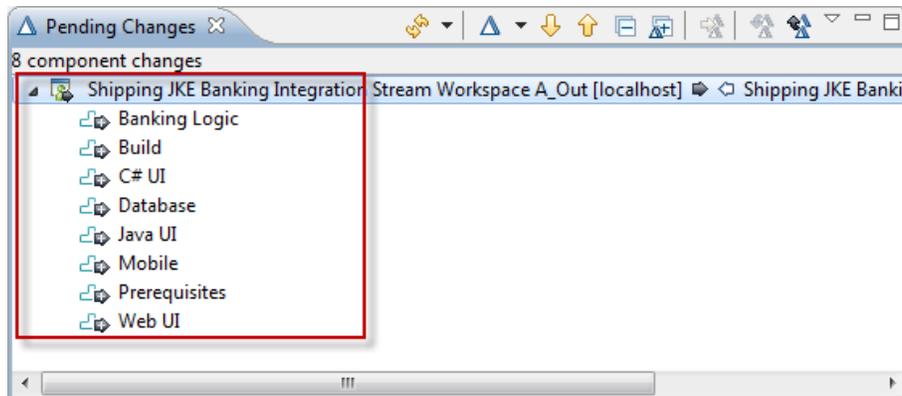


- __f. Save the changes.

- __95. Open the Pending Changes view.

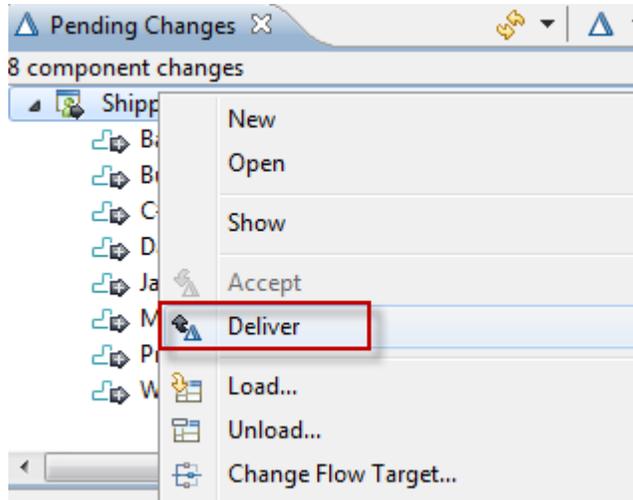


- __a. You can now see outgoing changes and component additions to the production server.

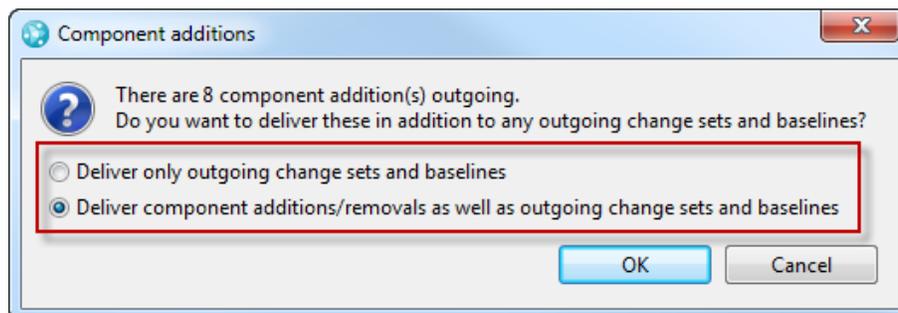


- __96. You have now successfully connected your repository workspace to the production servers stream.

- __97. You can now deliver changes to the stream.
- __a. Right Click on the shipping A_Out repository workspace. Select Deliver to transfer the changes to the Prod-B repository.



- __b. Select delivery of component additions to the stream.



- __c. Click OK to deliver the changes to the stream in Prod-B.

- __98. You have now successfully delivered the change sets to the stream in the repository of the production server Prod-B. They can be integrated into the streams by you or any other user.

4.3.3 Subsequent Shipping

You can use the Shipping Servers you just used to ship subsequent changes from site A to site B.

As described in **1.4.6 Bidirectional Shipping** and the following sections, use a dedicated shipping server to ship changes back. On site A name the folders for the shipping servers for example [C:\ShippingServer_A_OUT](#) and [C:\ShippingServer_B_IN](#). On site B name the folders [C:\ShippingServer_A_IN](#) and [C:\ShippingServer_B_OUT](#).

As described also described there use dedicated streams to ship changes out and in. Use incremental backups to reduce footprint.

Create Automation to automate the accept and backup as well as the restore and deliver operations. Use the RTC SCM CLI for accepting and delivering.

The Shipping_IN Streams can be used to accept, and integrate changes on the different sites. Integrated versions can be delivered to the Shipping OUT Streams to other sites.

4.3.4 Prepare Projects for Shipping

You can use one project for shipping, if you place all Shipping IN and Shipping OUT streams in that project.

If you want to ship from/to other projects you only need to prepare the project for shipping with the shipping user performing the steps below.

- __99. For the Projects you want to ship changes you need to add the shipping user, if this user is not yet member of the project.
- __100. Open the project area administration and add the shipping user to the project area as described in **3.1.3 Create a Shipping User and Configure the Project for Shipping**.
- __101. Create the shipping role in all of these projects as described in the same section.
- __102. Configure the shipping user and add the required Shipping role.

Now you can simply ship the backups back and forth. On the other site you need to follow **3.3 Transfer the SCM Changes to the Shipping Server** to ship the changes.

Appendix A Backup

There are various possible ways to create the backup of the shipping server. Options 2 and 3 are described in the document below.

1. You can create an offline backup following the Deployment Wiki entry [Backing up the Rational solution for Collaborative Lifecycle Management \(CLM\)](#). This backup and restore can be automated using scripts or other programming language.
2. To create backups, use the backup tools provided for your database.
 - See <http://www.ibm.com/developerworks/data/library/techarticle/dm-0502thalamati/> for information for Derby in our context above copying the files works.
3. Backup Types
 - It is possible to create a full back up that stores all data and allows to recreate it independently.
 - For some databases, it is possible to create incremental back ups. This backup type only stores the difference between a state of the database (from a prior backup). Incremental back ups can reduce the amount of data that needs to be stored and shipped.
4. You can essentially ship the whole shipping server, including the RTC Eclipse Client, for example using an image such as VMWare.
 - If using a Tomcat Derby based shipping server you can ship the whole shipping server RTC install for the first time. This will install and restore the complete Target Shipping Server and the required Client.

If you have used the ZIP install as suggested in Lab 2, and run on Tomcat and Derby you can just do the following.

__103. Zip the whole Folder [C:\ShippingServer](#) into **ShippingServer.zip**.

__104. Crypt the zip file.

__105. Send the encrypted zip file to Site B.

This also avoids having to install the shipping server and the client on the other site. You can decrypt and unzip the shipped server. You can rename the install folder as desired.

Appendix B Troubleshooting

In case of any issues:

- Make sure the permissions for the shipping user allow the required operations.

Make sure that you have entered the correct public URI's in the repository connections of the client. a mismatch of the public URI of the shipping server to the one used in the repository connection of the client can prevent you from shipping changes.

Appendix C Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix D Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM

IBM Logo

Rational

Jazz

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Apache, Apache Tomcat and Tomcat are trademarks of The Apache Software Foundation.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See [Java Guidelines](#)

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.



© Copyright IBM Corporation 2010, 2011

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Other company, product and service names may be trademarks or service marks of others.



Please Recycle