

**OpenText StreamServe Installation and  
System Administration v5.6.2**

**Customer Communications Management**

**OPENTEXT<sup>™</sup>**  
Learning Services

The software described in this Workbook is furnished under a license agreement or non-disclosure agreement (the "Agreement") with Open Text Corporation or one of its affiliates ("Open Text"). The software may be used or copied only in accordance with the terms of the Agreement. The information contained in this Workbook is the valuable property of Open Text and represents trade secrets of Open Text. No part of this Workbook may be copied, reproduced, translated or transmitted in any form or by any means without the prior written consent of Open Text.

The information contained in this Workbook is subject to change without notice. If this Workbook is provided in both printed and electronic form, the electronic form will govern in the event of any inconsistency. Open Text makes no representations or warranties concerning such information or the software described in this Workbook except as expressly set forth in the Agreement.

Copyright © 2015 by Open Text Corporation

OpenText is a trademark or registered trademark of Open Text SA and/or Open Text ULC. The list of trademarks is not exhaustive of other trademarks, registered trademarks, product names, company names, brands and service names mentioned herein are property of Open Text SA or other respective owners. All rights reserved.

OpenText StreamServe Installation and System Administration v5.6.2 - First Edition: June 2015

Comments or suggestions about this manual can be sent to LearningContentDev@opentext.com.

Based on OpenText StreamServe 5.6.2

Developed by OpenText Learning Content Development and Learning Services

Course Name: OpenText StreamServe Installation and System Administration

Course Number: 3-3730

Part Number: 3-3730-562-00

# Welcome

Connecting a business application to StreamServe and then connecting StreamServe to output devices will result in many points of possible failures – including network issues, printer device problems, etc. As a system administrator, you monitor the status of the document processing using log files, surveillance, and notifications.

At the end of this course you will be able to:

- Identify the requirements to install StreamServe
- Install StreamServe and review the disk structure
- Review the options to monitor StreamServe including logs, Java notifications, Status Messenger and Reporter
- Configure the Task scheduler
- Use the DB Admin Tool
- Identify the technical aspects that affect availability of StreamServe services
- Describe the trade-off between performance and fault tolerance
- Implement Profiling
- Implement procedures to improve high availability, fault tolerance and performance

Thank you for participating in this course. Should you require any further information, please contact us at OpenText Learning Services.

Good luck, and enjoy your learning experience.

*OpenText Learning Services*

*Open Text Internal Use Only  
Do Not Distribute*

# Table of Contents

<b>1. About StreamServe</b>	
Objectives .....	1-1
Overview .....	1-1
How We Do It.....	1-1
Who We Do It For .....	1-2
Industries .....	1-2
Customers .....	1-3
StreamServe Products .....	1-3
Production Workflow .....	1-3
How It Works .....	1-3
Main Components .....	1-4
StreamServer .....	1-4
Design Center .....	1-5
Control Center .....	1-6
StreamStudio .....	1-6
Web Applications .....	1-6
Access From One Page .....	1-7
More Components .....	1-8
Correspondence Management .....	1-8
StoryTeller .....	1-9
Composition Center .....	1-10
Ad Hoc Correspondence .....	1-11
Correspondence Reviewer .....	1-12
Post-Processing .....	1-13
E-Invoice Center .....	1-13
Output Center .....	1-14
RFID .....	1-14
SAP Document Presentment .....	1-15
StreamServe/Infor .....	1-16
Sustainability Program .....	1-17
Integrations .....	1-17
<b>2. StreamServe Components and Concepts</b>	
Components to Run StreamServer Applications.....	2-1
Control Center .....	2-2
Management Gateway.....	2-3
StreamServe Enterprise Repository (StrsSER).....	2-4
Application Domain.....	2-5
Recovery Actions .....	2-6
StreamServer Applications.....	2-6
Components to Run StreamStudio Web Applications .....	2-8
Service Gateway .....	2-8
StreamStudio Web Portal .....	2-9

User Directories.....	2-10
Components to Run StreamStudio Collector .....	2-11
Components to Run StreamStudio Composition Center .....	2-12
Definition of a Project.....	2-13
Design Center - Main Design Tool.....	2-14
StreamServe Project Components.....	2-15
Platform.....	2-16
Platform Layers.....	2-17
Generic Settings .....	2-17
Physical Layer Settings .....	2-18
Message.....	2-18
Events and Processes .....	2-19
Runtime Configuration .....	2-20
Resource Set.....	2-21
Export, Deploy, Run.....	2-22
Exporting Projects .....	2-22
Deploying Projects .....	2-23
Running StreamServer Applications .....	2-23

### 3. Installation Requirements and Planning

Components for StreamServer Applications .....	3-1
StreamServer .....	3-1
Framework and Management Gateway .....	3-2
StreamServe Enterprise Repository .....	3-2
Application Domain .....	3-3
Runtime Repository .....	3-3
Components for StreamStudio Web Applications.....	3-3
StreamStudio .....	3-4
Service Gateway .....	3-4
User Directories .....	3-4
Collector Archive .....	3-4
Web Content Repository .....	3-4
Planning the StreamServe Installation .....	3-5
Hardware Planning .....	3-5
Hardware Guidelines .....	3-5
Hardware Recommendations for Development Environment .....	3-6
Minimum Hardware Requirements for Test Environments .....	3-7
Hardware Requirements for Initial Production .....	3-8
Hardware Recommendations – Expert Sizing .....	3-8
Initial Sizing .....	3-8
Configuration and Project Development .....	3-8
Testing and Analysis .....	3-9
Re-Optimization .....	3-9
Enterprise Configurations Requirements .....	3-9
Typical Volumes .....	3-9
Example Configurations .....	3-10
Sizing Considerations .....	3-10
Throughput Based Sizing Considerations .....	3-10
Database Management System Sizing .....	3-10

Other Considerations .....	3-11
Separate Physical Servers .....	3-11
Choosing a Platform .....	3-11
Database Requirements for StreamServe Repositories.....	3-11
Database Configuration for Microsoft SQL Server .....	3-12
Database Configuration for Oracle Database .....	3-12
Managing License Files .....	3-13
<b>4. Installing and Upgrading StreamServe</b>	
Installing on a Computer Used for Development and Testing .....	4-1
Required Installation Steps .....	4-2
Installing in a Production Environment .....	4-2
Using a Central Enterprise Repository .....	4-2
Removing StreamServe Components .....	4-3
Upgrading StreamServe Components .....	4-4
Installing and Applying StreamServe Hotfixes .....	4-4
About StreamServe Hotfixes .....	4-4
Exercise - Installing StreamServe .....	4-5
<b>5. After Installation Configuration</b>	
Configuring StreamServe after Installation.....	5-1
About Control Center.....	5-2
StreamServer Components Recap .....	5-3
Planning Your Application Domains .....	5-4
Application Domains – Typical Scenario.....	5-5
Getting Started in Control Center.....	5-6
On First Run.....	5-7
Connecting to the Enterprise Repository.....	5-8
Creating the Enterprise Repository.....	5-10
Renaming the Site .....	5-11
Preparing to Run StreamServer Applications.....	5-12
Application Domain for StreamServer Applications.....	5-13
Creating and Configuring an Application Domain .....	5-14
Application Domain Editor .....	5-15
Administrator Tab .....	5-16
Runtime Repository Tab .....	5-17
Directory Tab .....	5-18
Recovery Tab .....	5-19
Web Services Tab .....	5-20
Web Content Repository Tab .....	5-21
General Tab .....	5-22
Creating a Runtime Repository .....	5-23
Adding a StreamServer Application.....	5-25
New Application Dialog Box .....	5-26
Deploying a Design Center Project.....	5-27
Deploy wizard – Select the Applications .....	5-28
Deploy wizard – Select File .....	5-29
Deploy wizard – Select Physical Layer .....	5-30

Testing a StreamServer Application .....	5-31
Exercise - Configuration After Initial Installation .....	5-32
<b>6. System Monitoring</b>	
StreamServe Components and Log Files .....	6-1
StreamServer Application Log Files .....	6-2
Management Gateway Log File .....	6-3
Service Gateway Application Log File .....	6-4
Task Scheduler Application Log File .....	6-5
Log Messages .....	6-6
Log Message Parts .....	6-6
Log Messages Included .....	6-7
Log Message Severity Levels .....	6-7
Log Examples .....	6-8
Logs in Control Center .....	6-10
Boot, Platform and Application Logs .....	6-11
Management Gateway Logs .....	6-11
Service Gateway Logs .....	6-12
StreamServer Logs .....	6-12
Task Scheduler Logs .....	6-12
Setting Log Levels.....	6-13
Platform Log Levels .....	6-13
Setting StreamServer Application Log Level .....	6-14
Information Used to Log Support Incidents .....	6-15
Debugging Applications.....	6-16
Logging to the Database .....	6-17
Monitoring with Surveillance .....	6-18
Configuring Surveillance .....	6-19
Configuring Surveillance Actions for the Host .....	6-20
StreamServe Notifications Overview .....	6-23
StreamServer Notification Types .....	6-24
Preparing the StreamServe Application to Use Notifications .....	6-25
Exercise 1 - Change the Log Level at Runtime.....	6-27
Exercise 2 - Configure Surveillance to Restart an Application .....	6-28
Exercise 3 - Configure Surveillance to Start a Program.....	6-30
Exercise 4 - Configure Surveillance to Send an Email.....	6-32
Exercise 5 – Configuring Java Notifications.....	6-34
Status Messenger Overview.....	6-37
Status Reports .....	6-37
Notifications .....	6-38
Why Use Status Messenger? .....	6-38
How It Works .....	6-39
Status Messenger Configuration .....	6-40
Disable Report Function for Other Applications .....	6-41
Configuring the -statusevent Startup Argument in the Project .....	6-42
Enabling Notifications .....	6-43
Generating Notifications for Project Components .....	6-44
Retrieving Notifications .....	6-44
Retrieving Notification for Specific Log Messages and Jobs .....	6-46

Configuring the Message for the Status Report .....	6-47
Configuring the Runtime Configuration .....	6-48
Creating More Than One Status Report for the Same Job .....	6-48
Exercise 6 - Status Messenger Project Preparation.....	6-48
Exercise 7 - Configure Status Messenger .....	6-52
Exercise 8 - Configure a Process, Output Connector and the Runtime Configuration..	6-59
Exercise 9 - Test Your Development.....	6-61
Reporter Overview.....	6-63
StreamStudio Application .....	6-64
Storing Job Metadata .....	6-65
Input and Output Jobs .....	6-65
Input & Output Job Instances .....	6-66
Configuring Queues to Store Job Metadata .....	6-67
Accessing Jobs Using Reporter.....	6-69
Preparing Searches .....	6-69
Searching for Jobs .....	6-70
Viewing Search Results .....	6-70
Input Job Details View .....	6-71
Output Job Details View .....	6-73
Resending Jobs.....	6-74
Configuring the Queues .....	6-75
Rerouting the Output .....	6-76
Exercise 1 – Use Reporter.....	6-76

## 7. Common Administrative Tasks

Task Scheduler Overview .....	7-1
Configuring a Task Scheduler Application .....	7-3
Configuring the Schedule.....	7-4
Deploying Document Types.....	7-5
How Document Type Deployment Works .....	7-6
Database Administration Tool.....	7-7
Database Administration Tool – Runtime Repository .....	7-8
Exercise 1 – Create a Task Scheduler Application .....	7-9
Managing StreamServe Applications .....	7-16
Starting, Stopping, and Restarting Applications .....	7-16
Deleting Applications .....	7-17
Updating and Exporting Application Properties .....	7-18
Updating Runtime Properties .....	7-18
Adding Other StreamServe Applications .....	7-19
Administering Applications on Remote Hosts .....	7-20
Applying a Hotfix to the Database.....	7-21
Management Gateway Users .....	7-22
Creating Users .....	7-23
Exercise 2 – Find StreamServe Components and Hotfix Information .....	7-24
Exercise 3 – Add a New User .....	7-26
Directory Structure for StreamServe Applications.....	7-28
Base Directory for Projects .....	7-28
Working Directory .....	7-29

Working Directories for Physical Platform Layers .....	7-30
Export Directory .....	7-31
<b>8. Scalability and Performance</b>	
Scalability Concepts .....	8-1
Availability .....	8-1
Fault Tolerance .....	8-1
Scalability .....	8-1
Clusters .....	8-2
Performance Overview.....	8-3
Profiling the StreamServe Environment .....	8-5
Enabling the Profiler Service .....	8-6
Configuring the Profiler Service.....	8-7
Profile providers .....	8-8
Filters .....	8-9
Optimizing StreamServe Processing .....	8-10
Optimizing Performance with Job Scaling .....	8-10
Drawbacks of Using Job Scaling .....	8-11
Recommendations .....	8-11
Optimizing XMLIN Events Creating a Thread Pool for XMLIN Events .....	8-11
XPath performance Considerations for XMLIN Events .....	8-12
Tuning the Threads for the Input and Output Queues.....	8-12
Configuring the Threads for the Input and Output Queues .....	8-13
Enabling Asynchronous Input Directory Scanning.....	8-14
Storing Queued Documents on Disk .....	8-15
Disabling Sorting for the Queues.....	8-15
Tuning the Service Queue.....	8-15
Tuning the Service Queue for the StreamServer Application .....	8-16
Tuning the Service Queue for the Service Gateway .....	8-17
Optimizing StreamServe Repositories .....	8-18
Tuning the Database Connection Pools.....	8-18
Database Connection Pools - StreamServer .....	8-18
Database Connection Pool – Management Gateway.....	8-20
Database Connection Pools – Service Gateway .....	8-21
Connection Pool in repositorymanager.xml .....	8-21
Connection Pool in securitymanager.xml.....	8-22
Resetting (Flushing) a Database Connection Pool.....	8-23
Increasing the Size of the Memory Buffer .....	8-24
Optimizing StreamServe Web Applications .....	8-25
Tuning the Thread Pool for the Service Gateway .....	8-26
Tuning the Keep Alive Timeout for the Service Gateway .....	8-26
Tuning the Response Timeout for the Service Gateway .....	8-28
Tuning Web Service Properties.....	8-29
Tuning Socket and Connection Timeouts .....	8-30
Load-Balancing Service Gateways .....	8-30
Tuning Timeouts for Failed Service Gateways .....	8-31
Tuning Connections Pools for Service Gateways .....	8-31
Tuning the Preview Cache for the SSSP Application .....	8-32
Allocating Memory to StreamServe Web Applications .....	8-33

Changing Java Virtual Machine Version .....	8-33
Exercise 1 – Profiler Services .....	8-34
Exercise 2 – Analyze Profile Output .....	8-38

Open Text Internal Use Only  
Do Not Distribute

Open Text Internal Use Only  
Do Not Distribute

# Text Conventions

This workbook uses the following conventions:

Convention	What it is Used For
<i>Italic</i>	Italics are used for Workshops and Exercises.
Monospace	Monospaced text is used to represent sample code.
<b>Bold</b>	In instruction steps, indicates the action to be taken. In text it indicates emphasis
<>	Angle brackets (<>) represent an element of syntax you must substitute with a specific value.
	This icon represents a lesson symbol where the student watches the instructor.
	This icon represents a lesson symbol where the student follows along with the instructor.
	This icon represents a lesson symbol where the students perform the exercise on their own.
	This icon represents an optional or advanced lesson symbol where the students perform the exercise on their own.
	This icon represents a note that supplies additional information.
	This icon represents a tip that supplies additional shortcut information.
	This icon represents a collection of Tricks, Tips, and Traps that is used the end of a chapter.
	This icon represents a caution that supplies warning information.

*Open Text Internal Use Only  
Do Not Distribute*



## OpenText StreamServe Installation and System Administration v5.6.2

### Student Attendance Form

**Training Date:** \_\_\_\_\_

**Instructor:** \_\_\_\_\_

**Location:** \_\_\_\_\_

**Student Name:** \_\_\_\_\_

**Position:** \_\_\_\_\_

Management       Technical       Other   
Implementation       End User       Administrator

**Industry:** \_\_\_\_\_

Federal Government       Legal   
Other Government       Manufacturing   
Education       Financial/Insurance   
Integrator       Other

**Company:** \_\_\_\_\_

**Street Address:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**E-mail:** \_\_\_\_\_

**Phone Number:** \_\_\_\_\_



## 1. About StreamServe

### Objectives

On completion of this chapter, participants should be able to:

- Identify the functionality scope of OpenText StreamServe
- Describe the functionality of the OpenText StreamServe solution components

### Overview

StreamServe unlocks and extends the business value of enterprise documents to deliver a personalized and unified customer experience.

### How We Do It

**General Business** StreamServe is a solution for any business, and can be described with three main concepts, as described below.

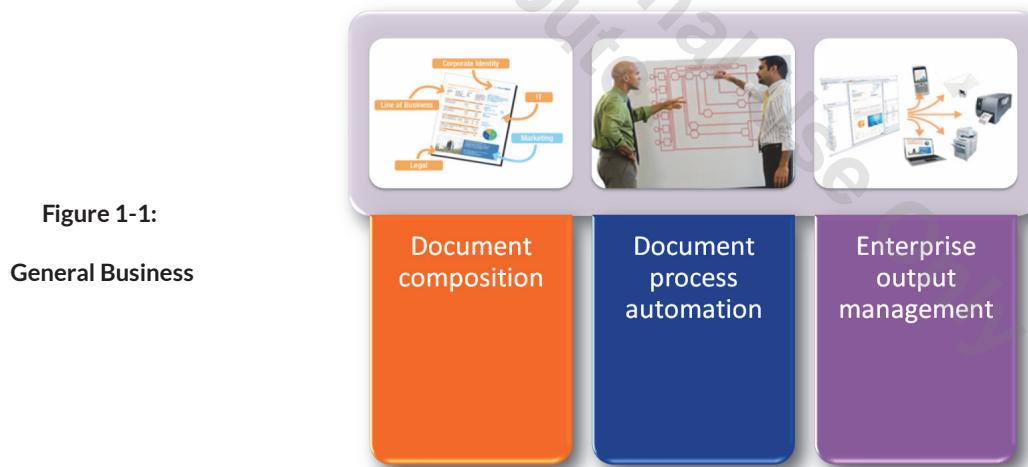


Figure 1-1:

General Business

**Document Composition** Transform ordinary customer communications into powerful one-to-one marketing channels.

**Document Process Automation** Improve customer, partner and supplier relations by streamlining document-intensive business processes.

**Enterprise Output Management** Produce and deliver customized documents when and where you want them.

## Who We Do It For

In addition to handling documents for general businesses, StreamServe focuses on the following industries providing the indicated benefits.

### Industries

- |                     |  |
|---------------------|--|
| <b>Banking</b>      | <ul style="list-style-type: none"><li>● Modernized statements enable more customer-friendly presentation of account status and history, and offer flexible, customer-selectable delivery options that include economical and environmentally friendly electronic formats.</li><li>● Business managers can directly implement rules to automatically capitalize on specific customer profiles; for example, including personalized recommendations for account options, or new or additional products and services.</li></ul> |
| <b>Insurance</b>    | <ul style="list-style-type: none"><li>● Modernized correspondence enables more customer-friendly presentation, and offers flexible, customer-selectable delivery options that include economical and environmentally friendly electronic formats.</li><li>● Business managers can directly implement rules to automatically capitalize on specific customer profiles. For example, bills or claim resolution correspondence can include personalized recommendations for new or additional products and services.</li></ul>  |
| <b>Supply Chain</b> | <ul style="list-style-type: none"><li>● It enables clearer, more customer-friendly presentation of account and transaction information, for example, and accelerates the introduction of targeted marketing messages or compliance information.</li><li>● Based on rules that business managers create and control, the individual customer's metrics trigger the inclusion of additional information on the documents; e.g., customer-specific cross/upsell offers, special sales, or new product offerings.</li></ul>      |
| <b>Utilities</b>    | <ul style="list-style-type: none"><li>● It enables more customer-friendly presentation of usage and patterns, for example, and accelerates the introduction of targeted marketing messages or compliance information.</li><li>● Based on rules that business managers create and control, the individual customer's metrics trigger the inclusion of additional information on the bill; e.g., conservation recommendations, budget programs, or new product offerings.</li></ul>  |

**Customers**

StreamServe technology has been deployed by more than 5,000 customers in 130 countries, representing virtually every major industry sector and a broad cross-section of multinational corporations. From small and medium-sized enterprises to some of the world's largest organizations.

**Figure 1-1:**  
**Customers**

**StreamServe Products**

**Production Workflow** The illustration below show the StreamServe production workflow engine, where various components are included.

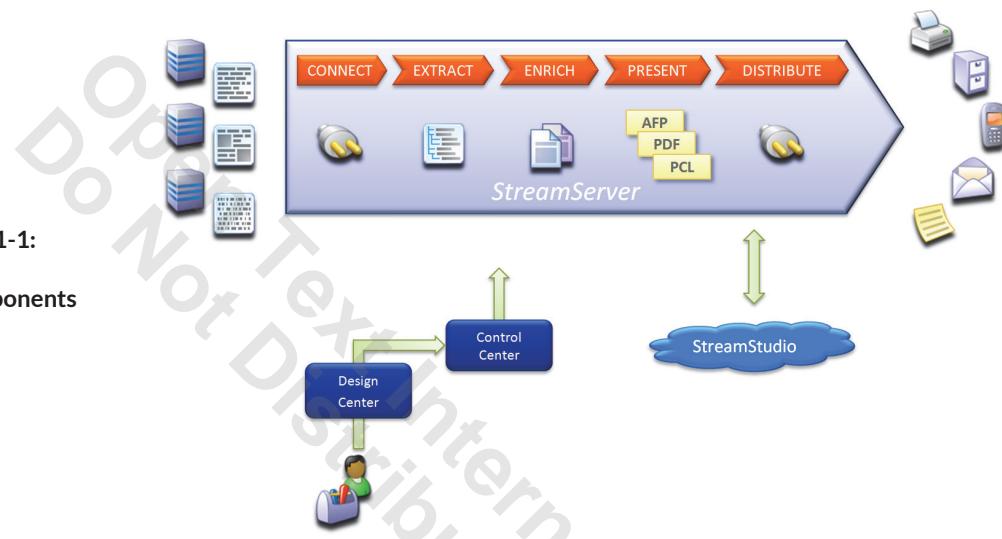
**How It Works** The illustration below shows a general life of a document - from born in the backend system, processed by StreamServe and finally produced at the final destination media.

## Main Components

For a document designer and developer, the following components are the most important:

- Design Center
- Control Center
- StreamServer
- StreamStudio

Figure 1-1:  
Main Components

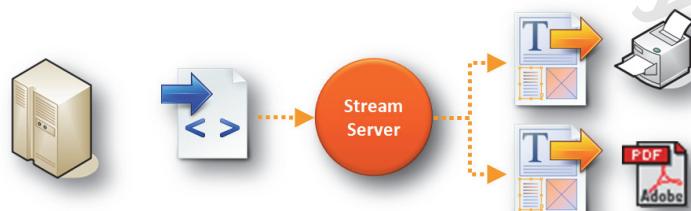


## StreamServer

### Data Collection & Transformation

The StreamServer is the component that handles the collection and transformation of input data, and the delivery of output documents. The StreamServer can run several StreamServer applications, where each StreamServer application is dedicated to a specific StreamServe Project.

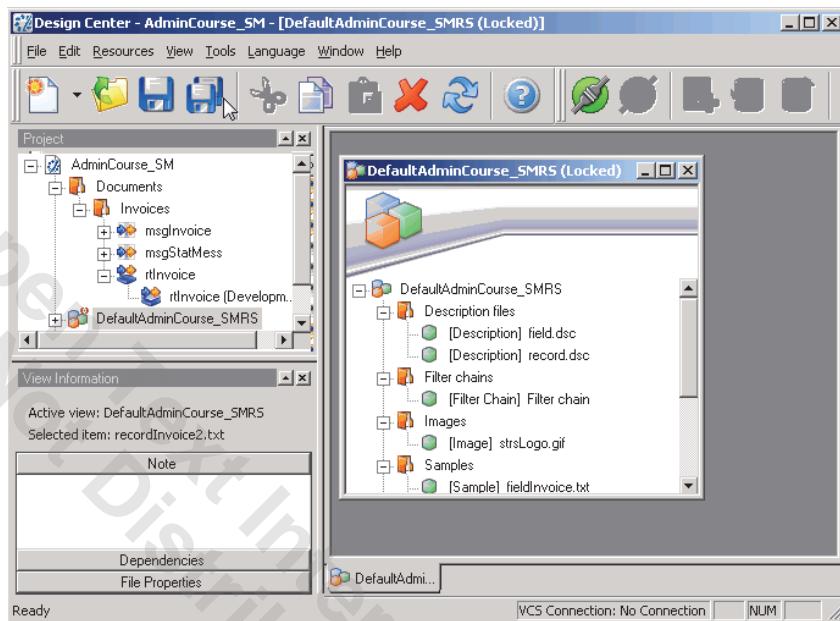
Figure 1-2:  
Data Collection & Transformation



## Design Center

**Main Design Tool** Design Center is the main design tool in StreamServe Persuasion. In Design Center you create your StreamServe Projects.

Figure 1-3:  
Design Center

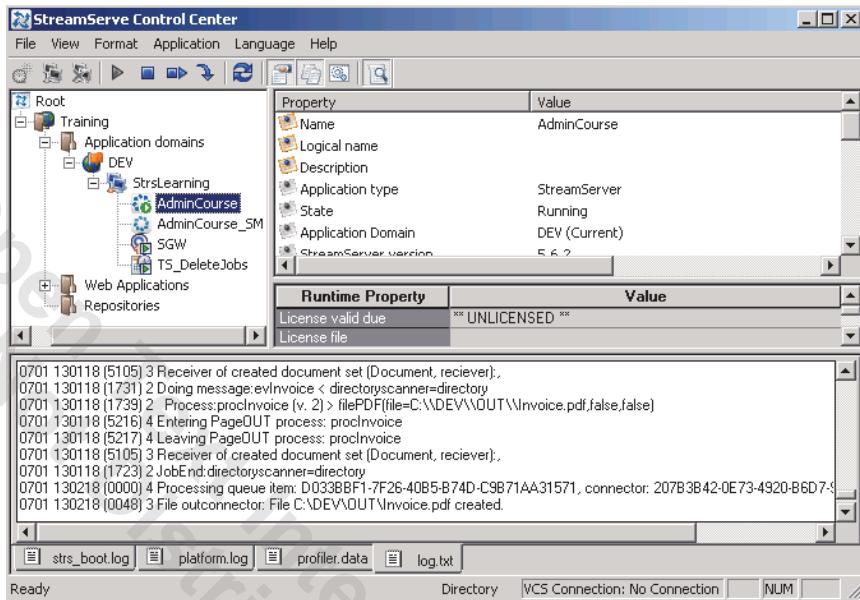


## Control Center

**Administrator Tool** Control Center is the administration tool in StreamServe Persuasion. In Control Center, you run and administer the StreamServer applications.

Figure 1-4:

Control Center



## StreamStudio

### Web Applications

StreamServe Persuasion includes a set of web applications:

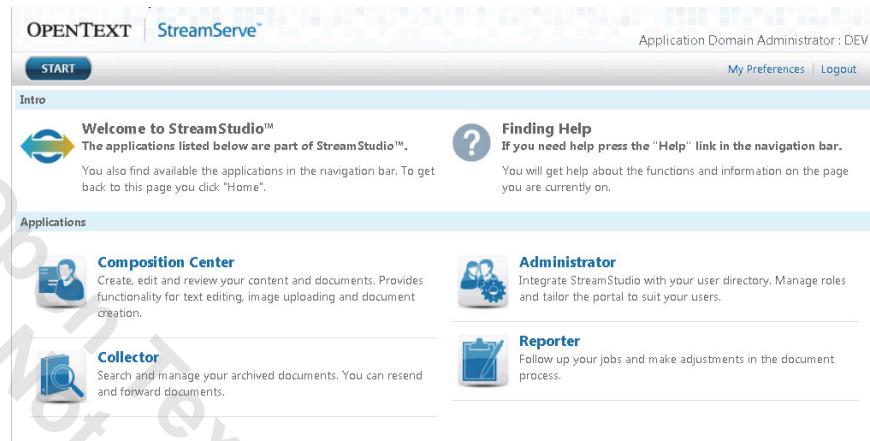
- Reporter
- Collector
- Composer
- Dispatcher
- Administrator
- My Preferences
- Composition Center

## Access From One Page

StreamStudio is the family name of the web applications. All StreamStudio applications are accessed from a single web page. Each web application can be branded and personalized on both corporate and personal level.

**Figure 1-5:**

### StreamStudio



- |                           |   |
|---------------------------|---|
| <b>Reporter</b>           | Reporter monitors all jobs processed by StreamServers. Jobs can be tracked end-to-end, re-processed, viewed, and much more.   |
| <b>Collector</b>          | Collector is a storage solution that enables storage of almost any document type for later retrieval. Collector has extensive search and sorting capabilities.  |
| <b>Composer</b>           | Composer enables any user with editing privileges to add, edit or remove text from already designed documents. Marketing messages, "weekly specials" and last minute changes are easily added in browser-based tools.   |
| <b>Dispatcher</b>         | Dispatcher controls the preferred distribution method for each document recipient. Independent from settings made during Project/ document development, Dispatcher helps the enterprise provide superior customer service.  |
| <b>Administrator</b>      | Administrator is the administrative center for StreamStudio. Users are added, typically by reading them from existing user directories, and assigned to groups and roles. Administrator provides powerful tools for controlling document flow and access.                     |
| <b>My Preferences</b>     | My Preferences ensures that each StreamStudio user can customize the look, feel and behavior of the web applications.   |
| <b>Composition Center</b> | Composition Center enables business users to manage personalized texts and uploaded images to use in business documents. For example, letters and account statements. Business users can also set rules to control when documents are generated and which content to include. |

## More Components

### Correspondence Management

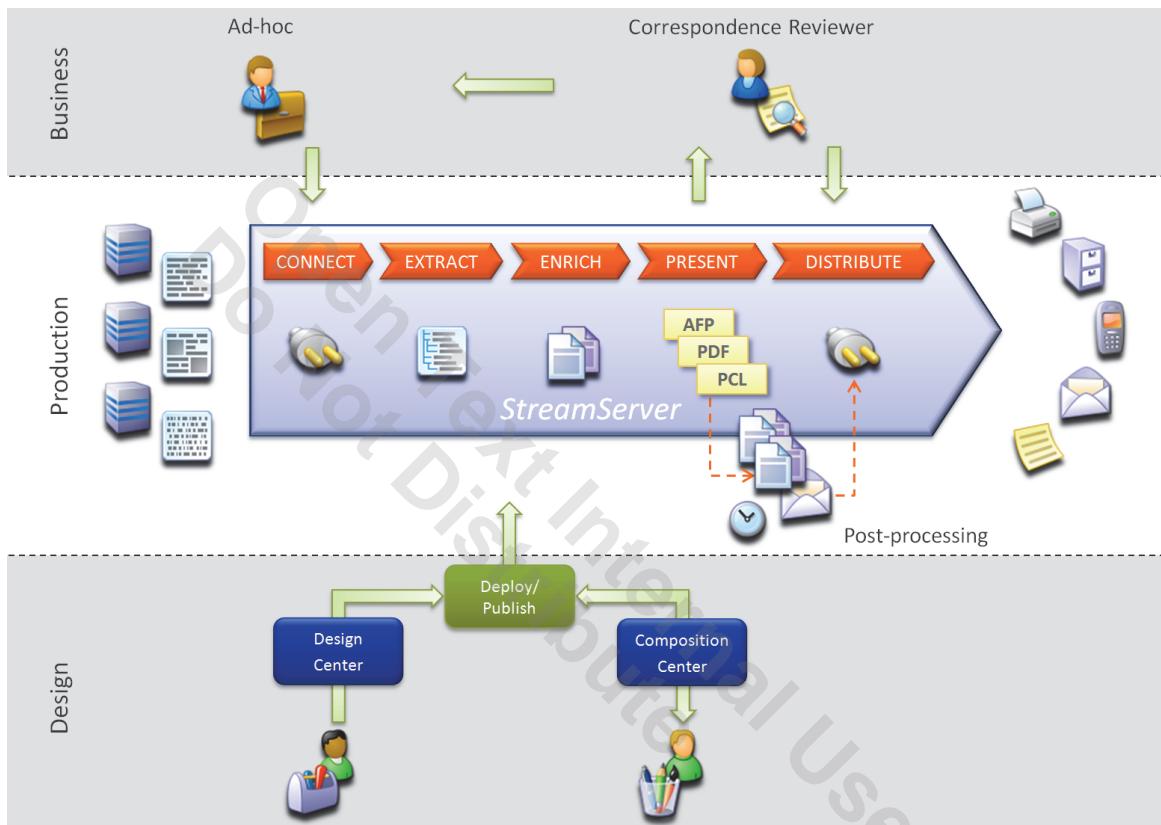


Figure 1-6: Correspondence Management

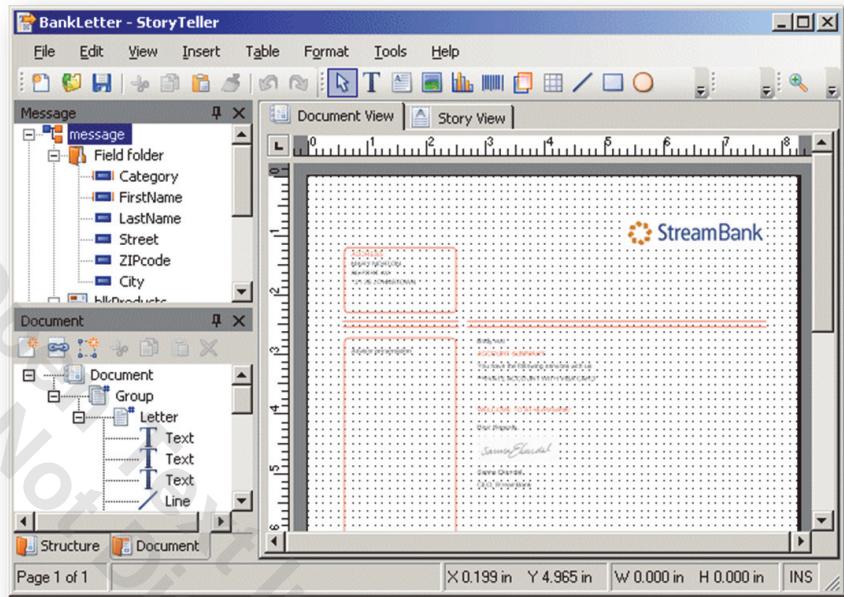
**Manage and Personalize Content** StreamServe Correspondence Management enables business users to manage and personalize content in business documents generated by StreamServer applications.

**Main Parts** Correspondence Management consists of the following main parts:

- The StoryTeller tool
- The StreamStudio Composition Center web application
- The Ad-Hoc Correspondence web extension module
- The Correspondence Reviewer web extension module

## StoryTeller

**Figure 1-7:**  
StoryTeller



**Process and Resource Tool** StoryTeller is a Process and resource tool in Design Center.

**Used by Developers and Designers** The tool is used by document designers and document developers to configure the structure, page design and layout of business documents.

**Data From Message & Composition Center** This is also where fields to retrieve from the Message and placeholders for content retrieved from StreamStudio Composition Center are configured.

**Templates => Composition Center** When a Project is deployed to a StreamServer application, each Composition Center enabled StoryTeller Process is made available as a template in Composition Center.

**Stand-Alone Tool** StoryTeller can also be used stand-alone outside Design Center, for example, by document designers.

## Composition Center

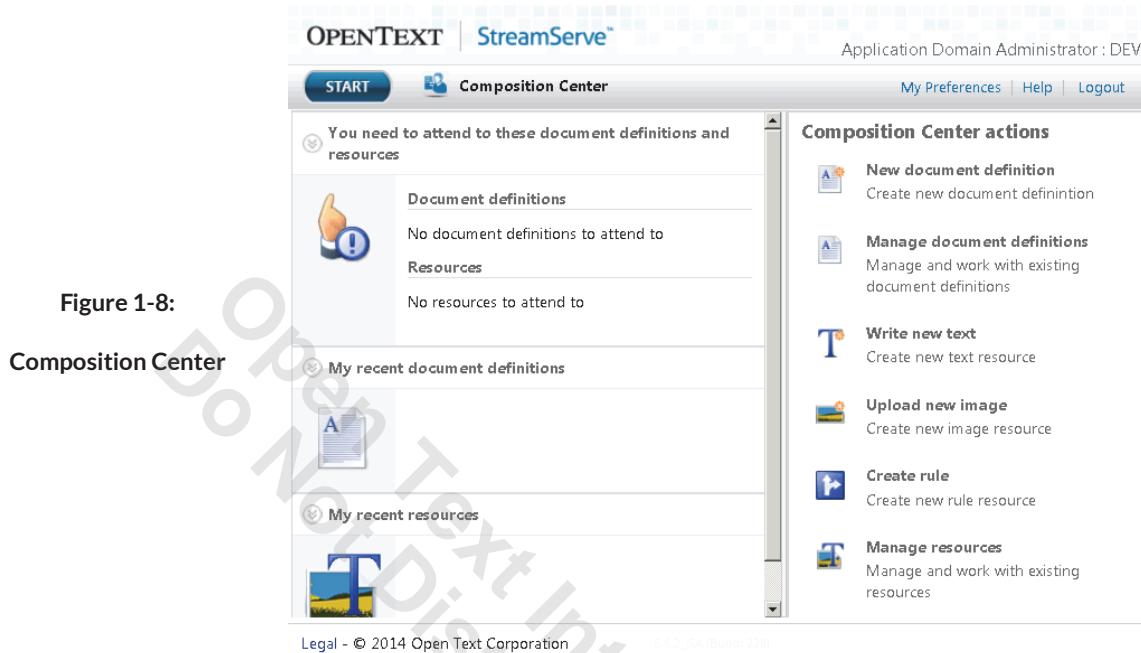


Figure 1-8:

### Composition Center

In previous releases, this application was named StreamStudio Correspondence Manager.

**Business Users Manage Content** StreamStudio Composition Center is a web application used by business users to manage content in documents, such as, personalized texts and uploaded images.

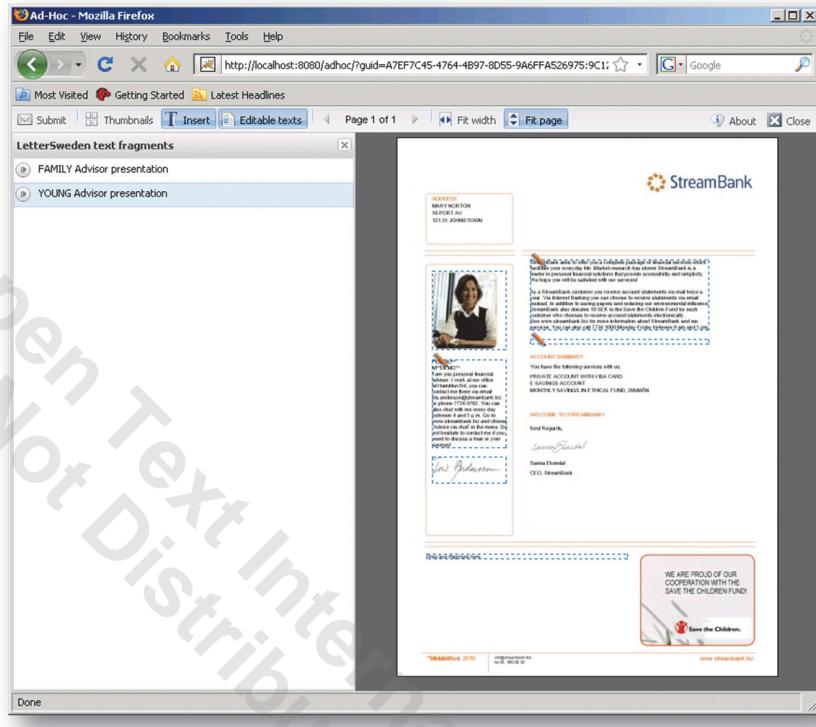
**Various Tasks** In Composition Center, business users can:

- Use the templates generated by StoryTeller Processes to create document definitions
- Add content to document definitions, preview final documents and put document definitions into production
- Set rules that determine when documents are generated and which content from StreamStudio Composition Center is used in production
- Make settings that control how content flows in the documents

## Ad Hoc Correspondence

**Figure 1-9:**

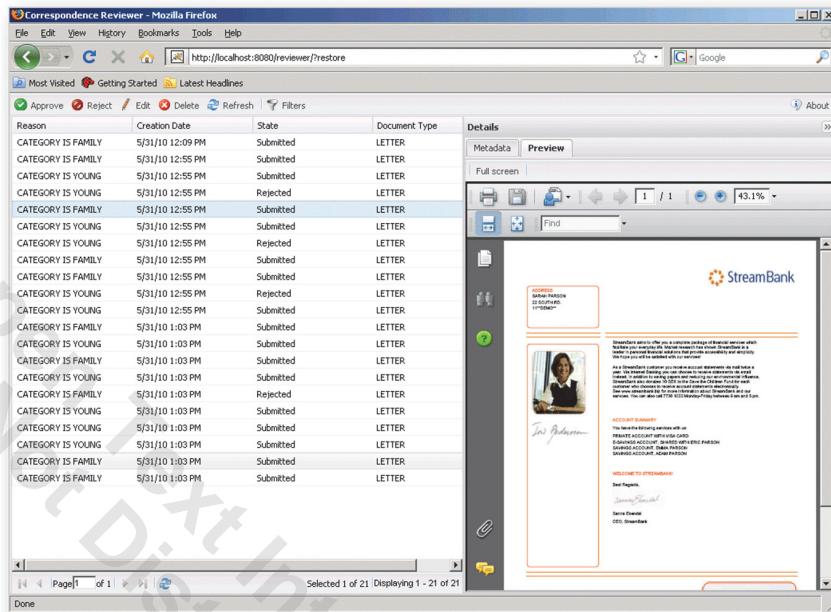
### Ad Hoc Correspondence



- |   |   |
|---|---|
| <b>Personalized Document to Specific Customer</b> | <p>Ad Hoc Correspondence supports an organization when creating ad-hoc customer correspondence. The application is used when writing a new personalized document to a specific customer, or when editing an existing document before it is formatted and distributed to the customer.</p> |
| <b>Document Templates From StreamStudio</b>       | <p>The document templates used in Ad Hoc Correspondence are document definitions, published in StreamStudio Composition Center.</p>   |
| <b>Frequent Customer Interaction</b>              | <p>A typical Ad Hoc Correspondence user is a business user at a department with frequent customer interaction that requires documents to be created on unplanned basis. For example, a business user at a bank, at a call center, or in a claims department.</p>                          |

## Correspondence Reviewer

**Figure 1-10:**  
Correspondence Reviewer



**Quality Assured Documents** Correspondence Reviewer helps an organization to create quality assured documents.

**Review Before Formatted & Distributed** The application is used to review a document before it is formatted and distributed to the customer.

**Approve, Reject, Edit, Delete** The reviewed document can be approved, rejected, edited, or deleted.

**Example—Business Manager Review** An example of a Correspondence Reviewer user is a business manager at a department with frequent customer interaction. The business manager reviews customer correspondence and approves or rejects the documents involved.

## Post-Processing

See the Post Processing product brochure for more information, available in the System Administration E-pack.

The brochure for StreamServe Post Processing features a blue header with the StreamServe logo and 'Post Processing' text. Below the header is a large image of a printing facility with several large industrial printers and workers. To the right of the image is a testimonial: 'We print 10 million pages per year. By cutting the volume of enveloped and mailed printed editions by 40% to 50%, we have realized a return on investment for the StreamServe solution in less than one year!' Below the testimonial are sections titled 'How Post-Processing Works' and 'Key Functionality' with a screenshot of the software interface showing various document layout options.

Figure 1-11:

## Correspondence Reviewer

## E-Invoice Center

See the E-Invoice Center product brochure for more information, available in the System Administration E-pack.

The brochure for StreamServe E-Invoice Center has a blue header with the StreamServe logo and 'E-Invoice Center' text. It features a large image of two people working at a computer. To the right is a testimonial from Peter Jenson, Project Manager, All Group: 'We chose StreamServe as our partner since they offered a solution that could handle requirements for electronic invoicing in a very simple and appealing way.' Below the testimonial is a diagram showing the StreamServe E-Invoice process flow from Supplier to Buyer, involving StreamServe, StreamServe E-Invoicing, and Compliance Gap. The brochure also includes sections for 'Market Scenario', 'Problem', 'Solution', and 'Goal'.

Figure 1-12:

## E-Invoice Center

## Output Center

See the Output Center product brochure for more information, available in the System Administration E-pack.



Figure 1-13:

## Output Center

## RFID

See the RFID product brochure for more information, available in the System Administration E-pack.



Figure 1-14:

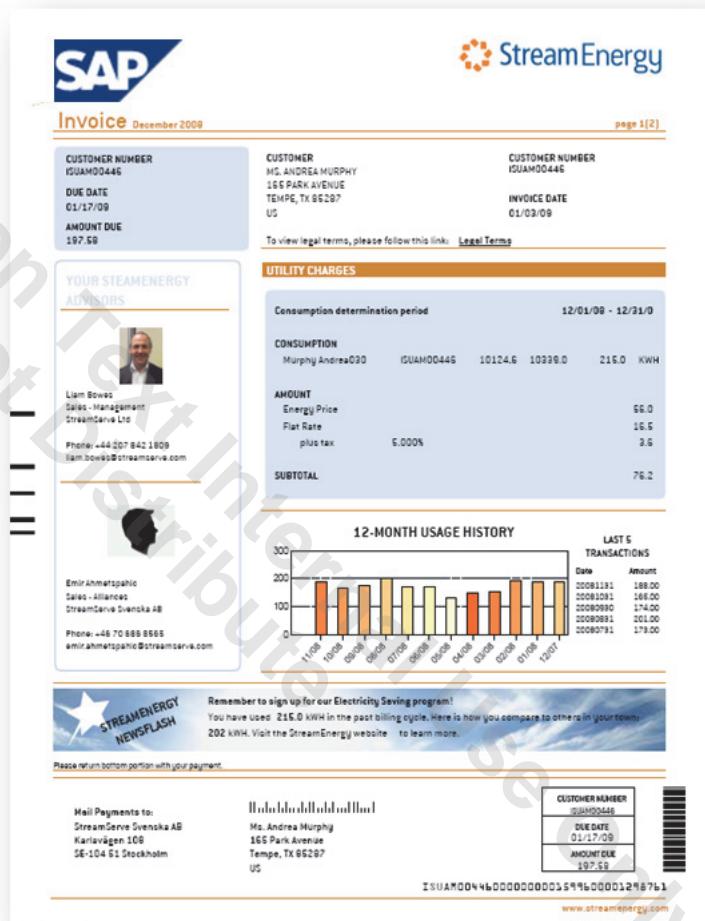
## RFID

## SAP Document Presentment

For more information, go to:

<http://www.sap.com/pc/tech/enterprise-information-management/software/document-presentment/index.html>

**Figure 1-15:**  
SAP Document Presentment



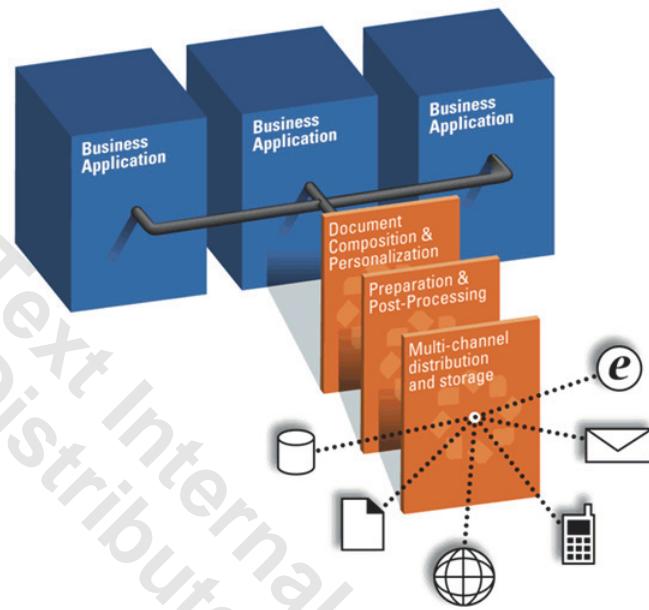
**StreamServe/Infor**

See the Base Package for Infor M3 product brochure available in the System Administration E-pack.

Figure 1-16:

Infor

## Infor - M3



## Sustainability Program

For more information, see:

<http://www.opentext.com/what-we-do/products/customer-experience-management/customer-communications-management>.

**Figure 1-17:  
Sustainability Program**

**Seven Ways to Increase Your Profitability While Helping the Planet**

- Migrate from paper to electronic documents.** StreamServe helps you get rid of paper by getting rid of other costs up to 50%.
- Reduce remaining paper-based costs.** StreamServe helps you reduce energy usage and other costs by up to 50%.
- Streamline document-dependent processes.** StreamServe integrates with your existing systems and replaces them with StreamServe's document management system.
- Capitalize on what you already have.** StreamServe helps you get rid of old equipment and replace it with StreamServe's document management system.
- Optimize IT infrastructure demands.** StreamServe integrates with your existing systems and replaces them with StreamServe's document management system.
- Solidify your standing with your extended community.** StreamServe helps you increase customer loyalty as well as the loyalty of your employees and others in their extended community.

**We Can Help You**

Based upon years of collected results from our customers and through consultation with industry analysts and experts focused on green IT, the StreamServe Sustainability Program offers research, best practices, and tools to help businesses determine the savings they can achieve with StreamServe's business communications solutions.

How much can really be saved?	What can be achieved?
Two million accounts converts 15% of their paper-based bills to electronic bills and replaces its monthly paper bill with an electronic bill. This results in paper based savings of 200,000 pounds annually or the CO <sub>2</sub> equivalent of:	11 homes average annual electricity consumption
StreamServe's utility customer achieves annual savings of up to \$1.6 million USD by delivering bills instantly via electronic means rather than having to print, sort and mail these in paper form.	41 passenger vehicles average emissions
StreamServe's utility customer achieves annual savings of up to \$1.6 million USD by delivering bills instantly via electronic means rather than having to print, sort and mail these in paper form.	10 acres of wetland forest sequestration
StreamServe's utility customer achieves annual savings of up to \$1.6 million USD by delivering bills instantly via electronic means rather than having to print, sort and mail these in paper form.	20 million printed sheets of paper

## Integrations

StreamServe integrates with other OpenText products such as:

- OpenText Enterprise Library Services
- OpenText Directory Services
- OpenText Media Management
- Vista Plus Output Manager 5.5.2
- OpenText Cordys Business Process Management (BPM) 4.
- OpenText Case360 11.3.

Information about StreamServe and the integration with other OpenText products is available in <http://knowledge.opentext.com>.

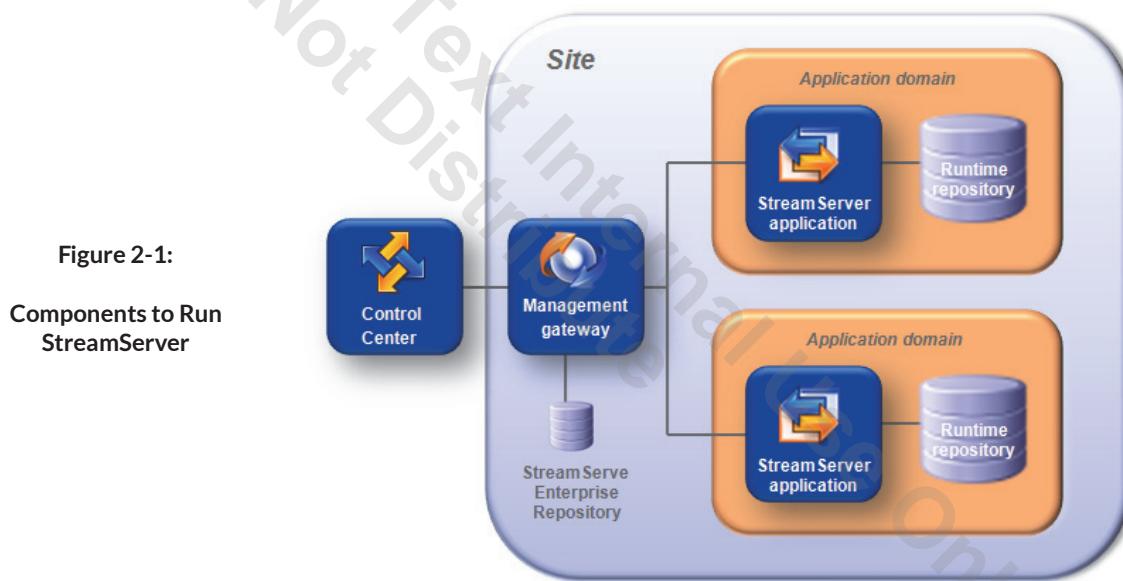
Open Text Internal Use Only  
Do Not Distribute

## 2. StreamServe Components and Concepts

On completion of this chapter, participants should be able to:

- Identify the components to run StreamServer applications
- Identify and describe the different Repositories
- Describe the StreamServer applications
- Identify the components to run the StreamStudio web applications
- Define Project, Platform, Message, Runtime configurations, Resource Set
- Export, deploy and run StreamServer applications

### Components to Run StreamServer Applications



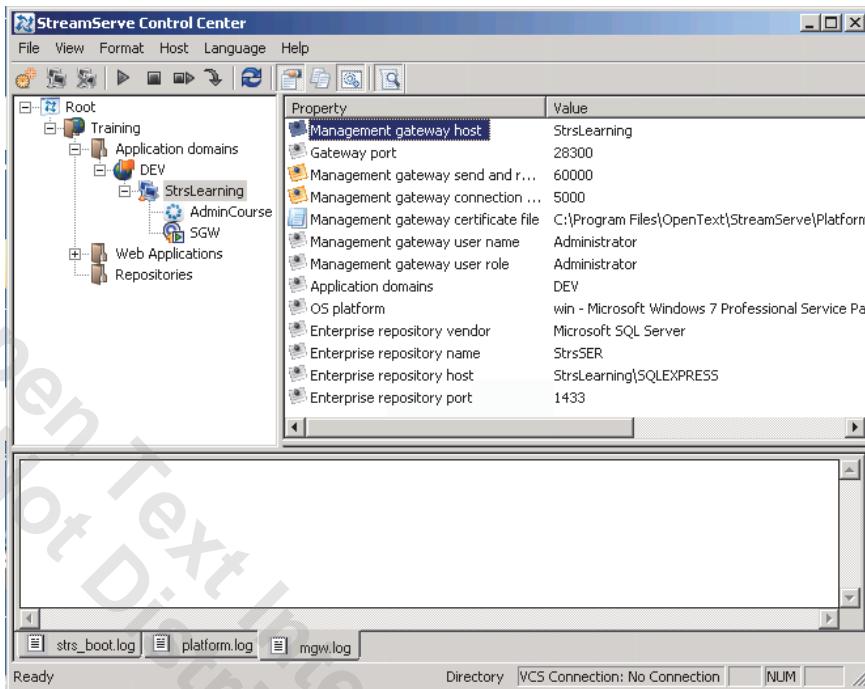
The following components are used to run Design Center Projects:

- Control Center
- Management gateway
- StreamServe Enterprise Repository
- Application domain
- StreamServer application
- Runtime repository

## Control Center

Figure 2-2:

Control Center



Control Center is an administration tool, which you use to deploy, run and administer StreamServe applications.

The applications are organized into:

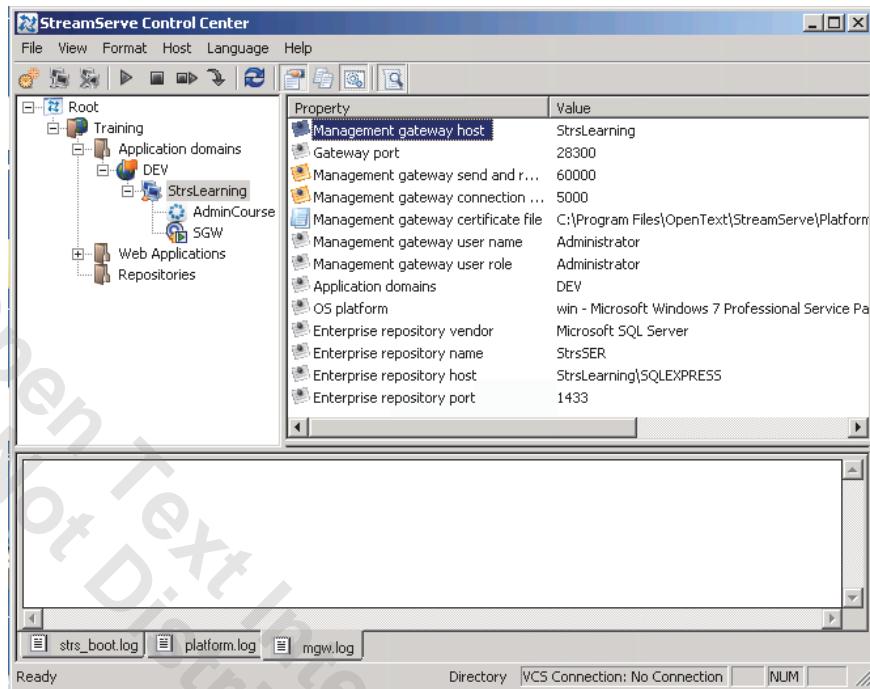
- Sites – for example, by company or organization.
- Application domains – for example, by department or region.

In Control Center, you also configure the StreamServe environment. For example, the databases, ports, etc.

You can use Control Center to run StreamServe applications on both Windows and UNIX hosts.

## Management Gateway

**Figure 2-3:**  
**Management Gateway**



Each computer where you run StreamServe applications has its own management gateway.

On Windows, the management gateway services are called:

- StreamServe Management Gateway 5.6.2
- StreamServe Management Nanny 5.6.2

On UNIX, the management gateway processes are called:

- ManagementGateway
- ManagementNanny

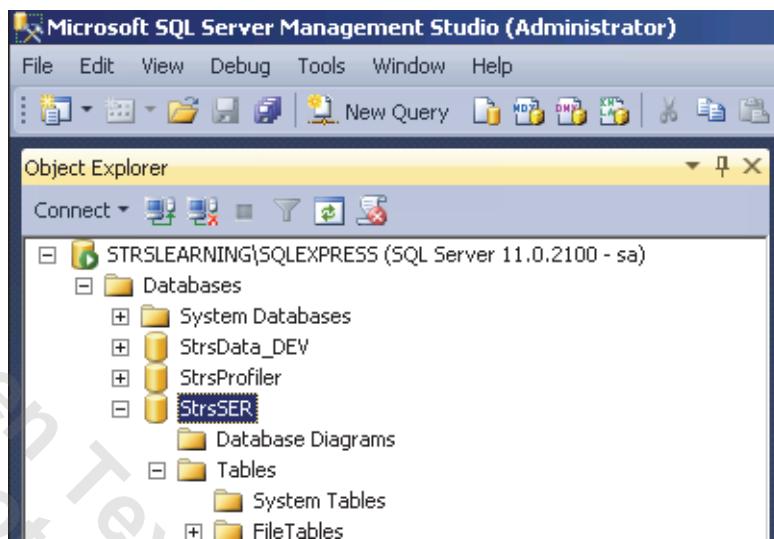
The management gateway is connected to the enterprise repository and Control Center.

The management gateway manages and monitors the applications on the computer.

The management gateway also manages the communication between applications and Control Center.

## StreamServe Enterprise Repository (StrsSER)

Figure 2-4:  
StreamServe Repositories

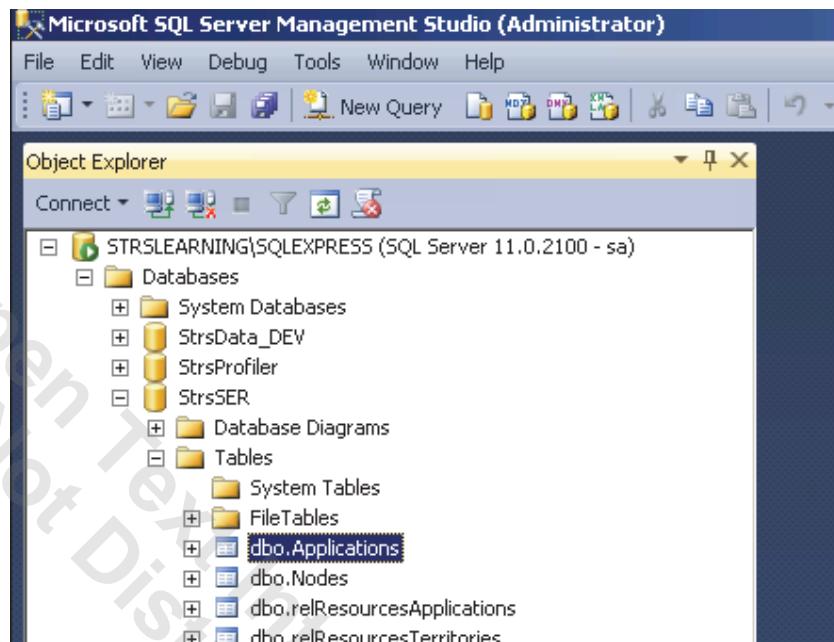


This database stores information about computers, StreamServer applications, and application domains for a company or an organization.

The repository is also the main storage for any document types, for example Invoice, Delivery note, and Order - used by the company or organization.

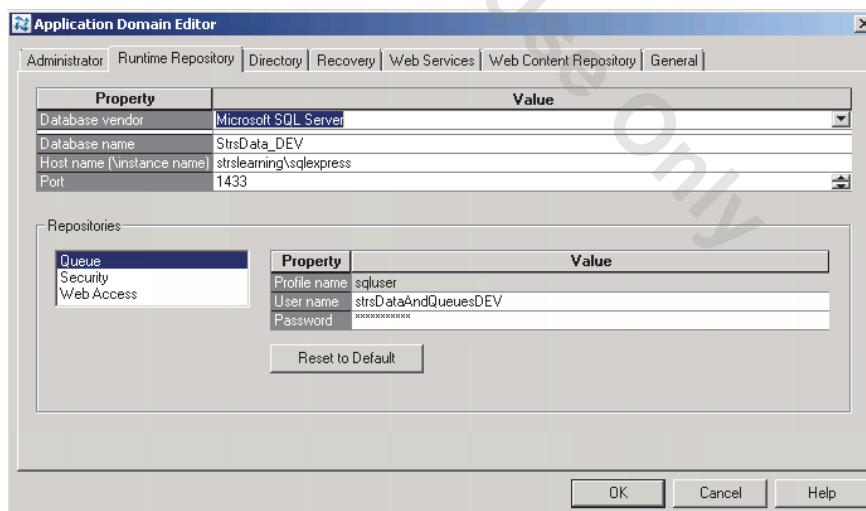
**Example – Enterprise Repository Table**

**Figure 2-5:**  
**Enterprise Repository Table**



**Application Domain**

**Figure 2-6:**  
**Application Domain**



You organize StreamServe applications into application domains.

All applications in one application domain share the same:

- Repositories
- Connection profiles
- Document types

**Applications for Departments or Regions** For example, you can use one application domain for a:

- Company
- Department
- Geographical region

**Application Domains for Test and Production** You can also use the one application domain for the test environment, and another for the production environment.

#### Recovery Actions

You can configure recovery actions for each application domain, for example:

- What actions to take if the connection to the runtime repository is lost.
- Who to notify if one of the applications in the application domain goes down.

## StreamServer Applications

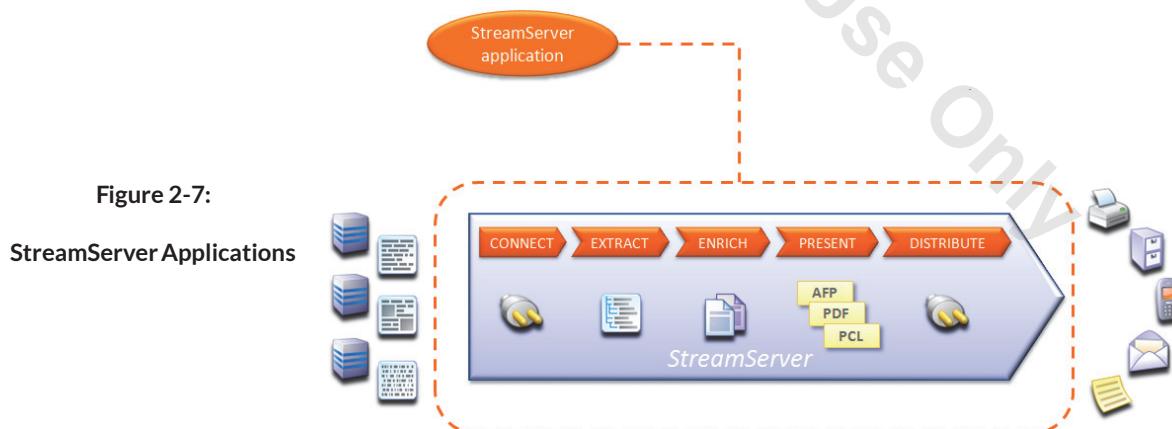


Figure 2-7:  
StreamServer Applications

StreamServer applications receive structured input data from other applications and ERP systems, then transform, format, and transmit documents to multiple communication channels.

Design Center is used to create and modify Project configurations, which define how StreamServer applications should process documents. Project configurations are then exported from Design Center, and deployed to StreamServer applications.

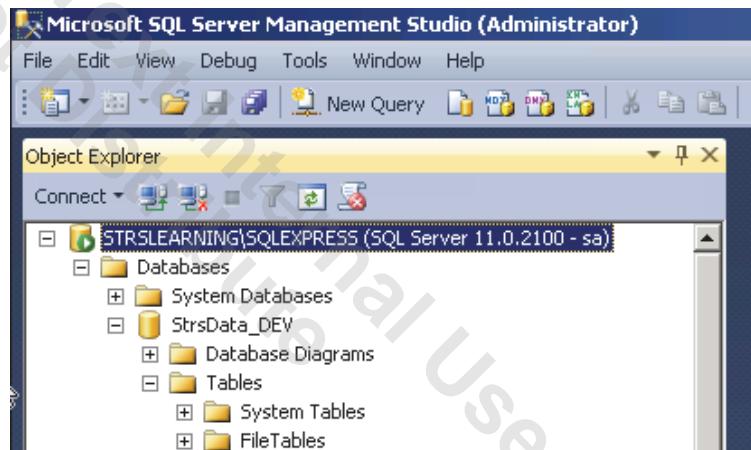
Each Project is deployed to, and run by a separate StreamServer application.

One application domain can contain several StreamServer applications running different Projects.

For failover reasons, you can deploy the same Project to several StreamServer applications.

## Runtime Repository (StrsData)

Figure 2-8:  
Runtime Repository



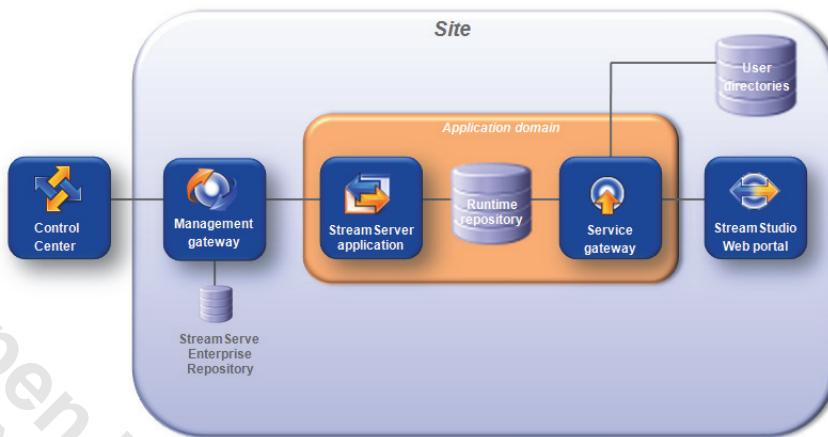
The runtime repository stores jobs and job related information produced by StreamServer applications.

The repository also stores security profiles and web access information for the StreamStudio web applications.

Each application domain requires a separate runtime repository.

## Components to Run StreamStudio Web Applications

**Figure 2-9:**  
**Components to Run StreamStudio Web Applications**

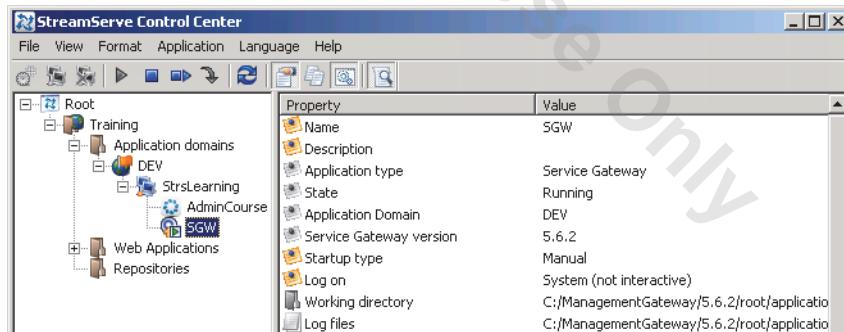


In addition to the components required to run StreamServer applications, you need to follow to run the StreamStudio web applications:

- Service gateway
- StreamStudio web portal
- User directories

## Service Gateway

**Figure 2-10:**  
**Service Gateway**

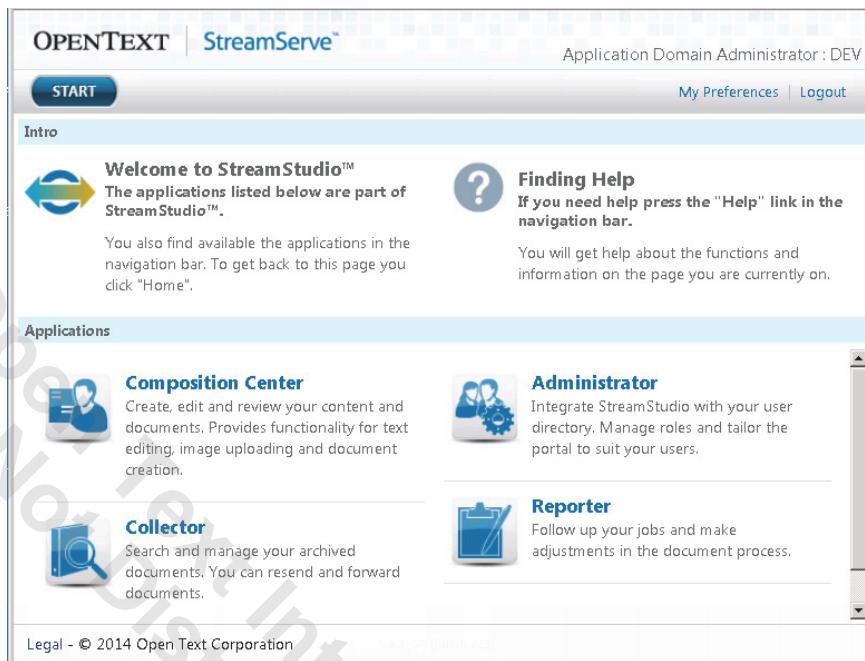


The service gateway is used to access the user directories and the runtime repositories from StreamStudio.

When a user logs in to StreamStudio, the service gateway authenticates the user credentials in the user directory and in the runtime repository.

## StreamStudio Web Portal

Figure 2-11:  
StreamStudio Web Portal



The StreamStudio web portal contains the StreamStudio web applications.

## User Directories

Figure 2-12:  
User Directories

```
dn: dc=streamserve,dc=com
dc: streamserve
description: Default domain for StreamStudio
objectClass: dcObject
objectClass: organization
o: StreamServe

dn: ou=users, dc=streamserve,dc=com
ou: users
objectclass: top
objectclass: organizationalunit

dn: uid=StrsAdmin, ou=users, dc=streamserve,dc=com
userpassword: insecure [REDACTED]
displayname: StreamStudio Administrator
uid: StrsAdmin [REDACTED]
objectclass: top [REDACTED]
objectclass: person [REDACTED]
objectclass: organizationalPerson [REDACTED]
objectclass: inetOrgPerson [REDACTED]
sn: StrsAdministrator [REDACTED]
cn: StrsAdministrator [REDACTED]

dn: uid=StrsReader, ou=users, dc=streamserve,dc=com
userpassword: insecure
displayname: StreamServe Directory Reader
uid: StrsReader
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
sn: StrsReader
cn: StrsReader
```

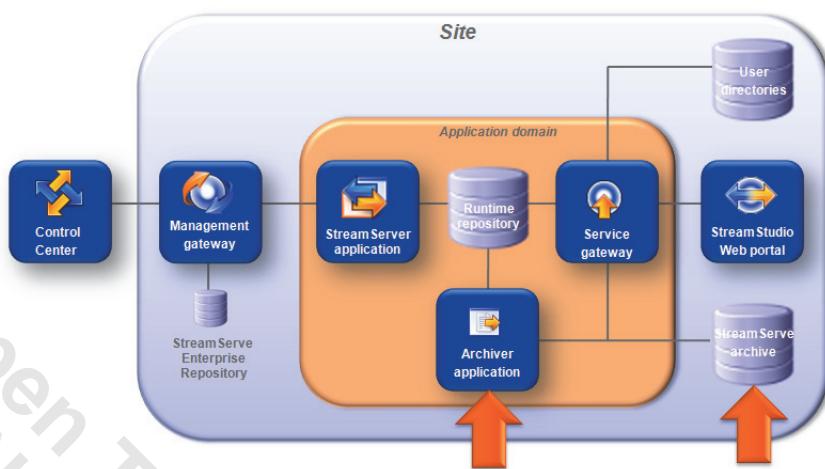
The user directories contain the user profiles to access the StreamStudio web applications.

The user profiles can also be used validate users assigned to StreamServe jobs.

The user directories can be connected to an existing directory server, which must support the LDAP protocol.

## Components to Run StreamStudio Collector

Figure 2-13:  
Components to Run  
StreamStudio Collector



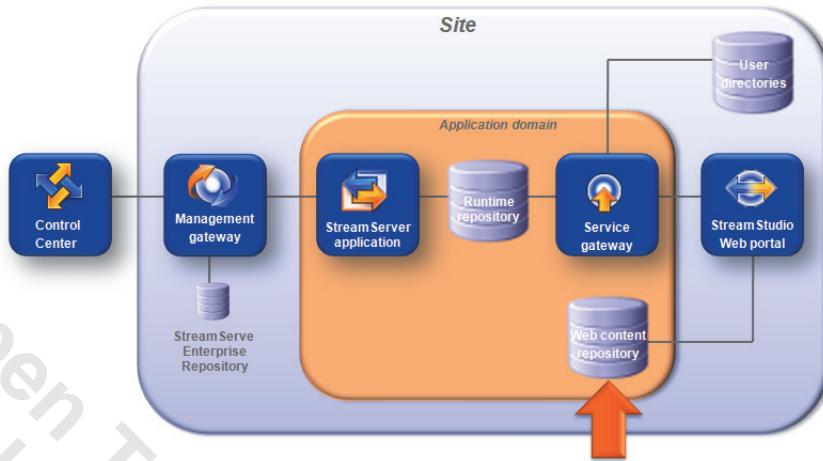
In addition to the basic StreamStudio components, an Archiver application and an Archive are needed to run StreamStudio Collector.

This database stores output documents and related metadata that are accessed from StreamStudio Collector.

This application transfers output documents and metadata from the runtime repository to the StreamServe archive.

## Components to Run StreamStudio Composition Center

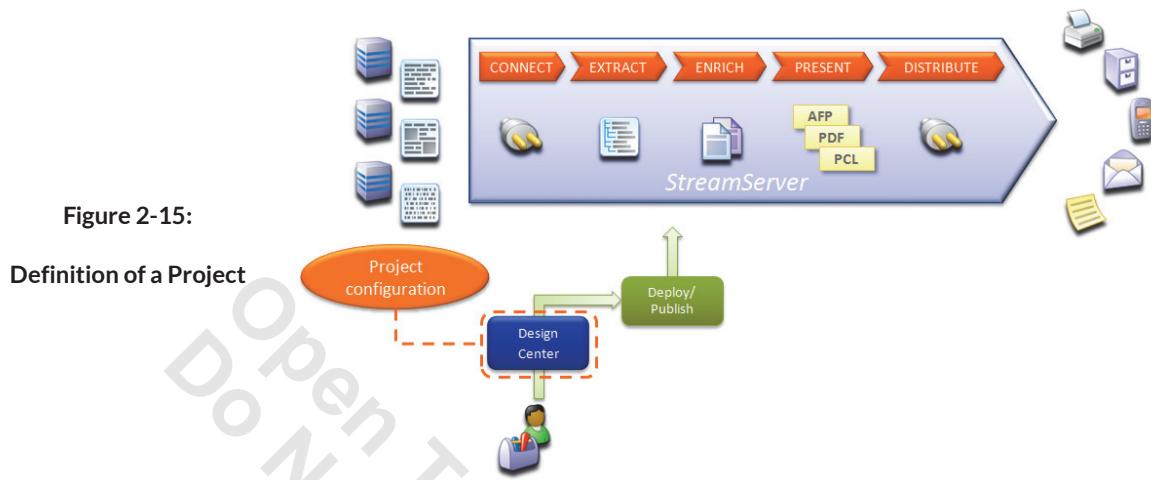
Figure 2-14:  
Components to Run  
StreamStudioComposition  
Center



In addition to the basic StreamStudio components, a web content repository is also needed to run StreamStudio Composition Center.

This database is used by the Composition Center web application for storing document definitions, resources, and rules during the document design phase.

## Definition of a Project



A StreamServe Project is the design of how to:

- Connect to a backend system.
- Extract data fields from incoming data/documents.
- Enrich output documents with graphical features.
- Present document in various formats, for example AFP, PDF and PCL.
- Distribute documents to various channels, such as File, Print, STMP.

In the Project, you configure all the settings needed by the StreamServer application.

## Design Center – Main Design Tool

Figure 2-16:  
Design Center – Main  
Design Tool

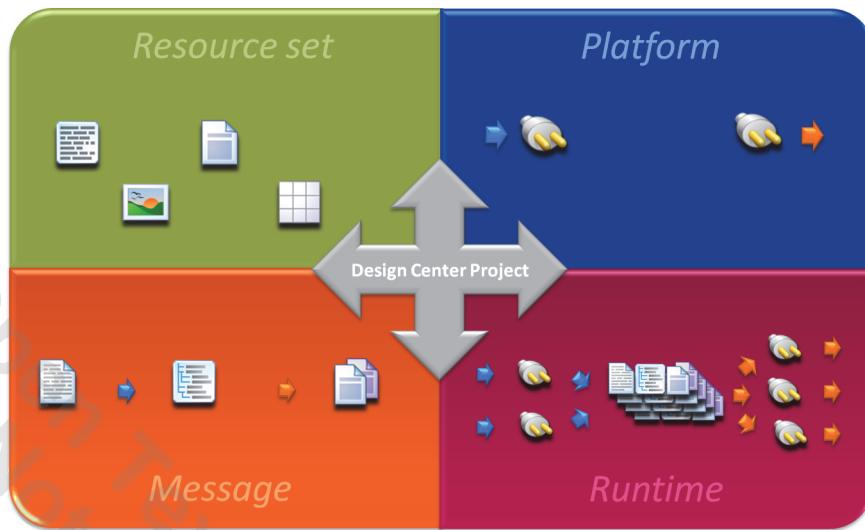


In Design Center, the project developer creates the Project structure and configures all Project components.

From within Design Center you can start all other tools you need when you configure the Project. For example: StreamIN, StoryTeller, Overlay Editor.

## StreamServe Project Components

Figure 2-17:  
StreamServe Project  
Components



A StreamServe Project contains the following main components:

- Platform
- Message
- Runtime configuration
- Resource set

## Platform

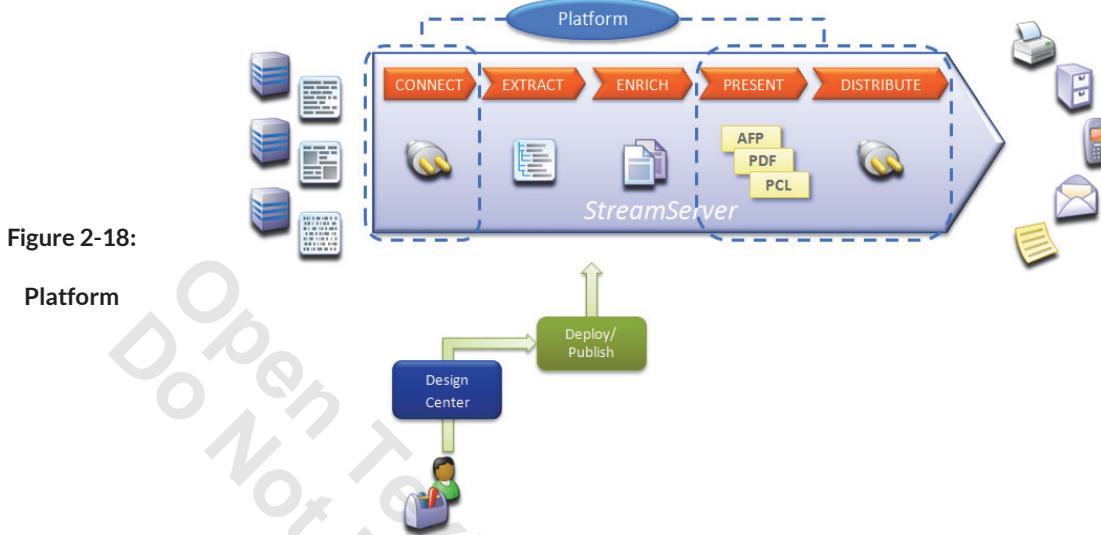


Figure 2-18:

The Platform is where the environment settings are configured.

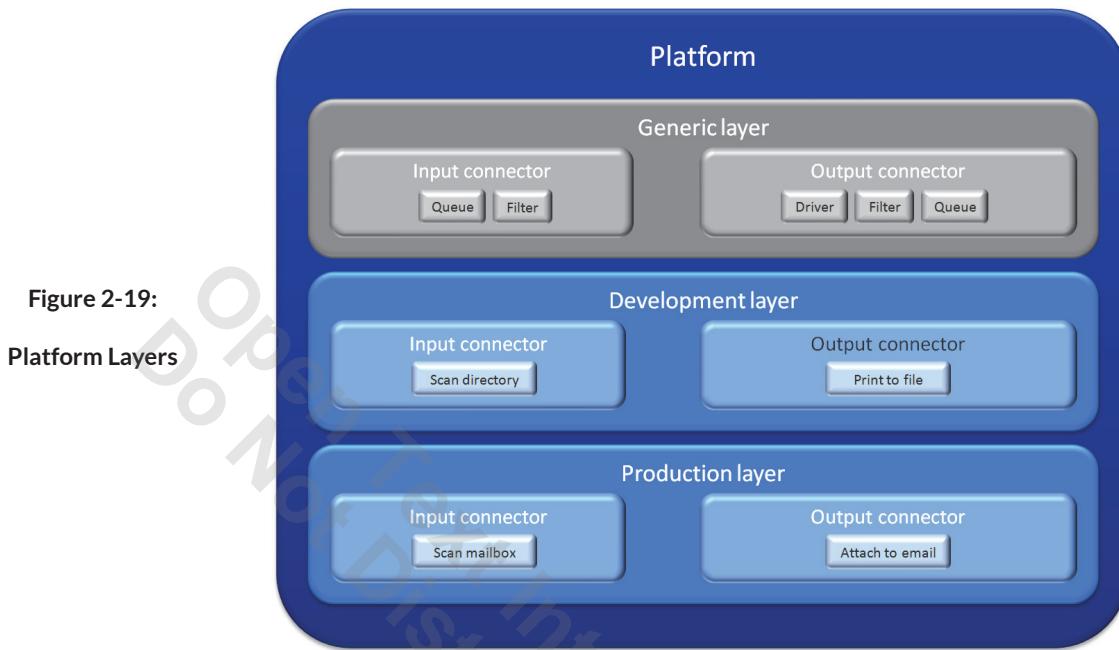
The Platform configuration determines how to:

- Connect to and receive input from source applications.
- Connect to and deliver output to the output devices (printers, faxes, etc).

A Platform contains input and output connectors:

- Input connectors specify how to receive input.
- Output connectors specify how to deliver output.

## Platform Layers



**Figure 2-19:**  
Platform Layers

The Platform is separated into one generic layer and at least one physical layer.

Why use Platform layers?

Using Platform layers makes it possible to use the same Design Center Project in the development, test, and production phase.

One layer can be used for the development environment and another for the production environment. Settings specific to each environment are configured in the corresponding layer.

### Generic Settings

The generic layer contains the settings that must be the same for all environments. For example:

- The number of input and output connectors.
- Which filters to use.
- Which queues to use.
- Which device drivers to use.

## Physical Layer Settings

Each physical layer contains environment specific settings, such as:

- The connector type.
- The settings for the selected connector type, for example, which host to receive the input data from.
- Log level.

## Message

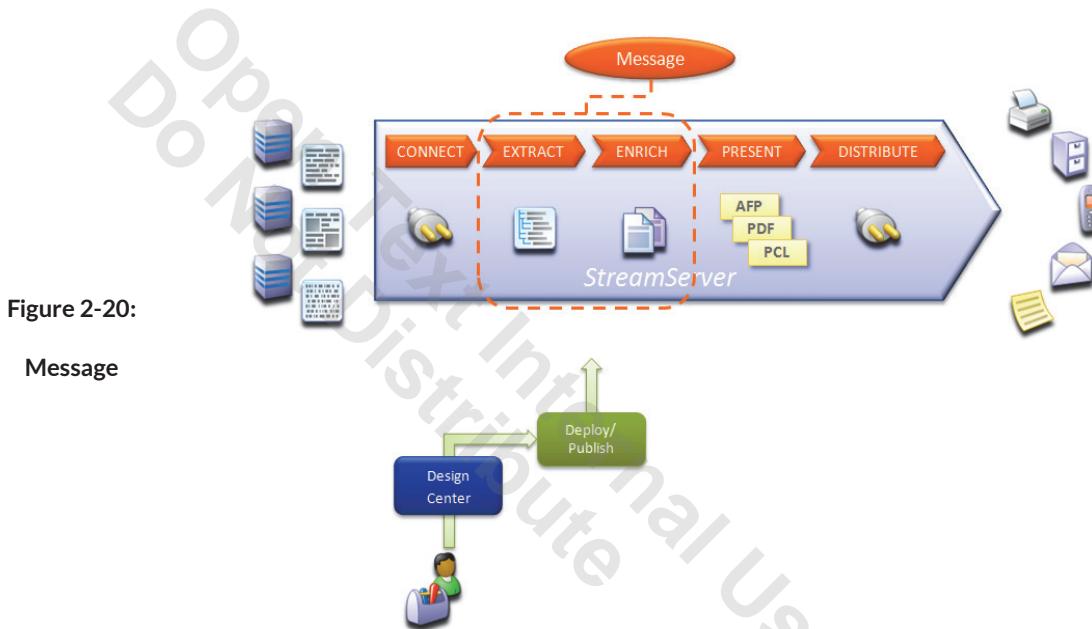


Figure 2-20:

### Message

A Message is where a specific type of document is configured.

The message configuration includes how to:

- Identify and extract fields from the input data.
- Aggregate and compose the output.

Projects can contain several messages. For example, one Message for invoices, another for orders, etc.

## Events and Processes

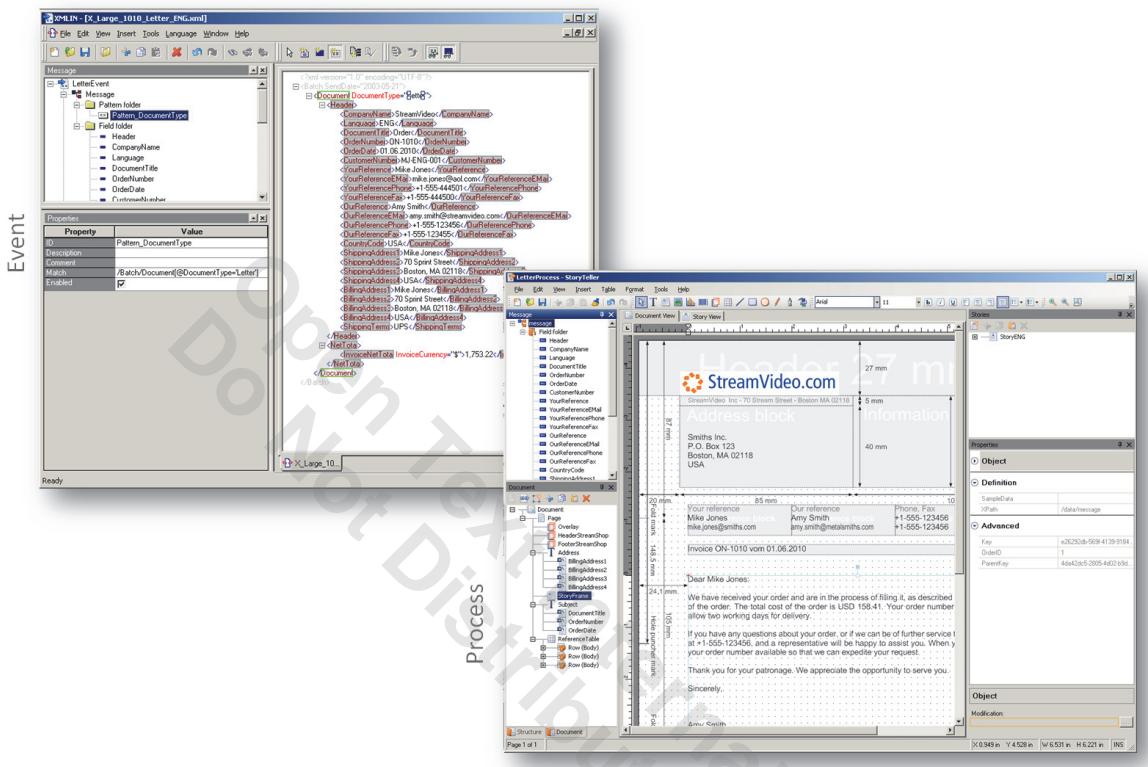


Figure 2-21: Events and Processes

A Message contains Events and Processes:

- An Event controls how to identify and extract fields from the input data.
- A Process controls how to aggregate and compose the output.

## Runtime Configuration

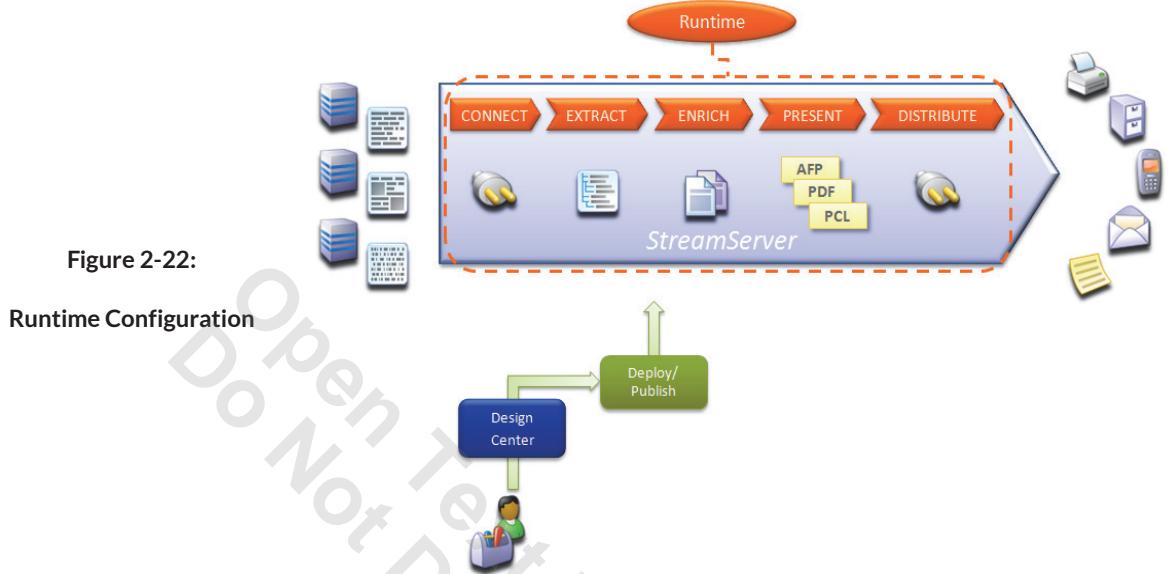


Figure 2-22:

Runtime Configuration

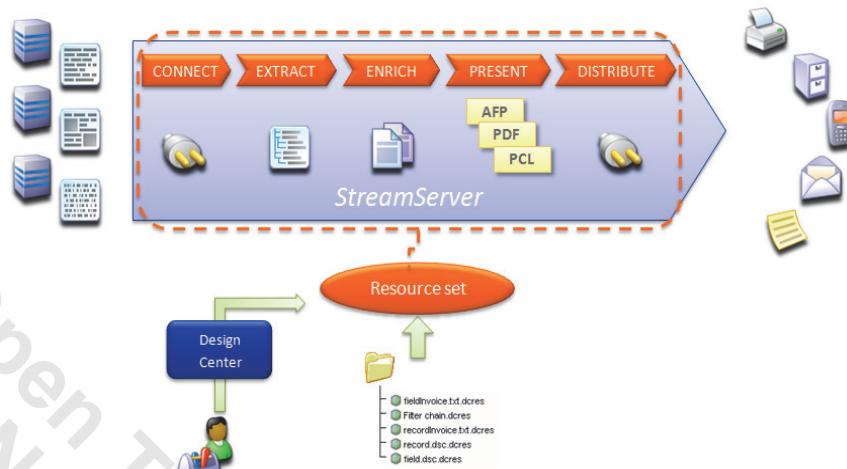
In the Runtime configuration, the project developer specifies how to connect input connectors to Events, and how to connect Processes to output connectors.

A Project can contain several Runtime configurations, for example one Runtime configuration per Message.

## Resource Set

Figure 2-23:

Resource Set



All external files used in a Project are accessed as resources via one or more resource sets.

Examples of files in a resource set:

- Image files
- Overlays
- Tables files
- Function files
- Sample files

## Export, Deploy, Run

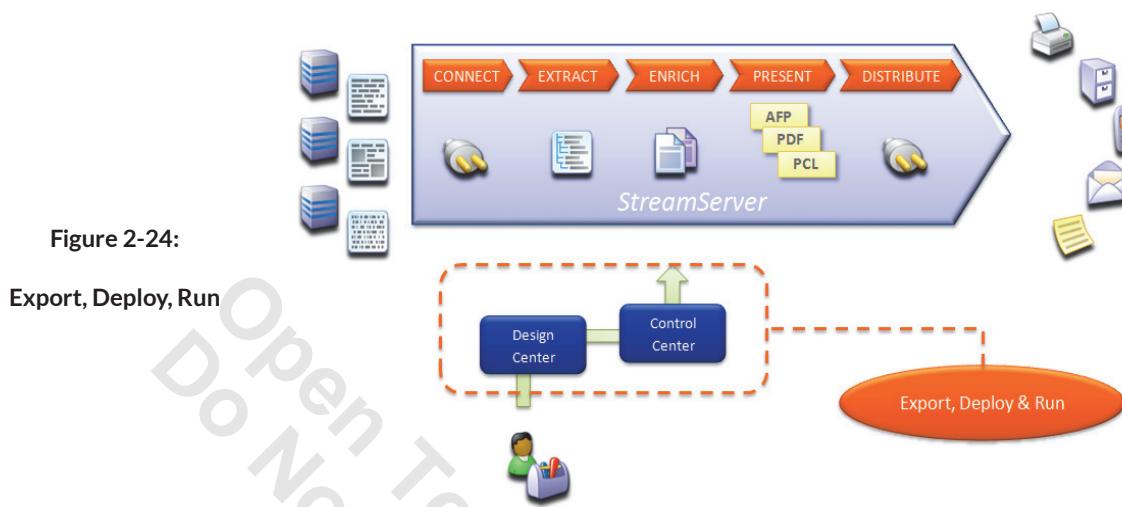


Figure 2-24:

Export, Deploy, Run

Project developers export, deploy and run Projects when they want to:

- Test the functionality of the Project.
- Create a release package when the Project is ready to be deployed in Production.

### Exporting Projects

Projects must be exported from Design Center to generate the files used by the StreamServer application.

**Export Files** Two files are created when you export a Project.

File	Description
Package file	This file contains the configuration files for the StreamServer application.
Checksum file	The checksum SVF file is used to check that the export file is valid during deployment.

**Methods of Exporting** There are two ways to export a Project.

Method	Description
Exporting to disk	A Project can be exported to disk if no Version Control System (VCS) is used for the Project files.  Project developers can also export to disk when testing the Project, even if a VCS is used for the Project files.
Exporting to a VCS	When creating a release, i.e. when the production layer is to be deployed to the StreamServer application, the Project is normally exported and checked in to a VCS.

**Deploying Projects**

The person responsible for running the StreamServer application uses Control Center to deploy the Project to the StreamServer application.

When a Project is deployed, the configuration files are unpacked from the export file to the working directory of the StreamServer application.

**Running StreamServer Applications**

When the Project is deployed, the responsible person can start and supervise the StreamServer application. This can be done using Control Center.

Open Text Internal Use Only  
Do Not Distribute

### 3. Installation Requirements and Planning

On completion of this chapter, participants should be able to:

- Identify the components of the StreamServe software
- Identify the steps that are required to install StreamServe
- Managing License files

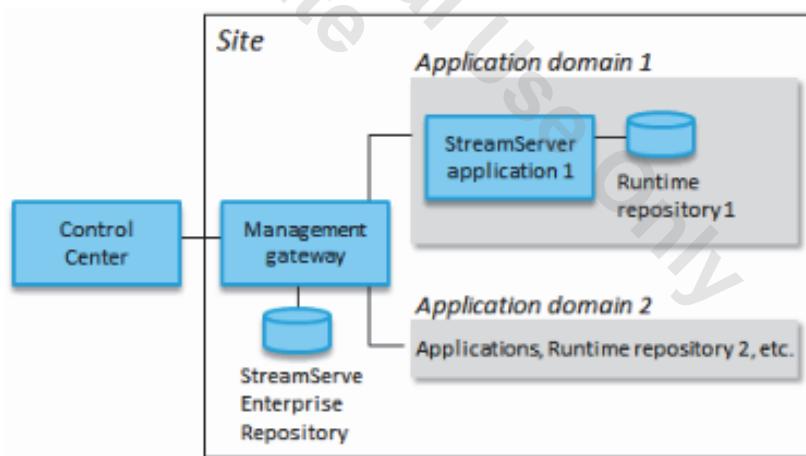
#### Components for StreamServer Applications

A StreamServe installation that is used to run OpenText StreamServe StreamServer applications includes the following main parts:

- StreamServer
- Framework and management gateway
- StreamServe Enterprise Repository
- Application domain
- Runtime repository

This section gives a brief overview of what each component is used for and how you install or configure the component.

Figure 3-1:  
Components for Running  
StreamServer Applications



#### StreamServer

This includes the software to run StreamServer applications.

StreamServer can be installed and run on Windows or UNIX using the Framework and StreamServer Setup.

**Framework and Management Gateway**

The Framework includes a management gateway and the underlying software to run StreamServe applications, such as, service gateways, Archiver applications, and task scheduler applications.

The management gateway performs the following activities:

- Manages and monitors the StreamServe applications on the computer.
- Connects the computer to a StreamServe Enterprise Repository.
- Handles the communication between Control Center and the StreamServe applications.

The management gateway Windows services are called:

- StreamServe Management Gateway <version>.
- StreamServe Management Nanny <version>.

On Windows, the management gateway services use the local administrator account. These services require local administrator access to create and remove other Windows services.

The management gateway processes are called:

- ManagementGateway
- ManagementNanny

The framework and management gateway are part of the Framework and StreamServer Setup.

To manage the StreamServe applications on a computer, each user needs a login to the management gateway. A Management Gateway Administrator with the user name Administrator is included with the Framework and StreamServer Setup, this user is assigned the administrator role.

**StreamServe Enterprise Repository**

This database is used for deployment information, such as:

- StreamServe applications at one company or organization
- Application domains at the company or organization
- Computers used to run these applications
- Master document types used by the company or organization

In distributed environments, you should have one central enterprise repository at your company or organization.

After the installation, you use Control Center to create the enterprise repository or the Command Line Utilities to generate the scripts to create the repository. The default name for enterprise repository is StrsSER.

<b>Application Domain</b>	Depending on what you want to achieve in terms of document storage and access, you organize StreamServe applications in application domains. All applications in one application domain share the same repositories, connection profiles, document types, and channels for reprocess and preview.
	You configure application domains using Control Center or the Command Line Utilities.
<b>Runtime Repository</b>	This database is used to store job and document data for the StreamServe applications in one application domain.  After the installation, you use either Control Center or the Command Line Utilities to generate the scripts for the runtime repository. The default name for the runtime repository is StrsData.

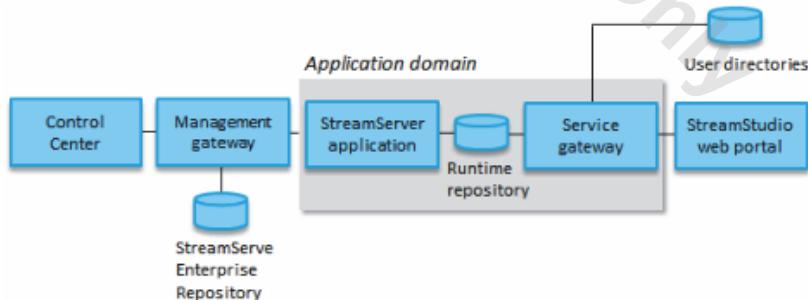
## Components for StreamStudio Web Applications

In addition to the StreamServer components, you use the following components to run OpenText StreamStudio web applications:

- StreamStudio web portal
- Service gateway
- Collector archive (optional)
- Web content repository (optional)
- User directories

This section gives a brief overview of what each component is used for and how you install or configure the component.

Figure 3-1:  
StreamStudio Components



<b>StreamStudio</b>	<p>This is the web portal containing the StreamStudio web applications.</p> <p>The StreamStudio web portal can be installed and run on Windows or UNIX using the StreamStudio Setup.</p> <p>To run StreamStudio you need a Java application server, for example Apache Tomcat. See the Supported platforms and software documentation for a list of the supported Java application servers.</p>
<b>Service Gateway</b>	<p>This is a web service gateway that connects the StreamStudio web applications to the runtime repository and user directories.</p> <p>The service gateway is installed with Framework and StreamServer on both Windows and UNIX. The service gateway Windows service is called StreamServe Service Gateway. The service gateway process is called ServiceGateway.</p>
<b>User Directories</b>	<p>The user directories contain the user profiles for the StreamStudio web applications.</p> <p>The user directories can be connected to an existing directory server that supports the LDAP protocol. See the Supported platforms and software documentation, for a list of the supported user directories.</p> <p>You configure the connections to the user directories using Control or the Command Line Utilities.</p>
<b>Collector Archive</b>	<p>This database stores output documents and related metadata used for the StreamStudio Collector web application. The default name for the Collector archive is StrsCollector.</p> <p>You use Control Center or the Command Line Utilities to generate the scripts used to create the Collector archive.</p>
<b>Web Content Repository</b>	<p>This database stores Composition Center resources and roles during the document design phase. The default name for the web content repository is StrsWeb.</p> <p>You use Control or the Command Line Utilities to create the web content repository.</p>

## Planning the StreamServe Installation

The following steps are required before you begin a StreamServe installation:

- Determine the hardware that is required for your StreamServe solution.
- Ensure that you have the required software for each StreamServe component you want to install.
- Ensure that you have access to a database installation configured for use with StreamServe components.

**Securing StreamServe installations** To comply with your company's security policy you may need to secure your StreamServe installations.

**Supported software and platforms** For the official list of supported software and platforms, see Supported platforms and software documentation.

## Hardware Planning

You can install all StreamServe components on the same computer. You can also use different computers for the following components:

- Control Center
- Design Center
- Framework and StreamServer
- StreamStudio
- StreamServe repositories

The number of computers used to implement your StreamServe solution depends on the individual requirements at your company, such as, company infrastructure, organizational structure, operational practices, etc. You should also consider the complexity of the StreamServe solution and type of applications you will run when planning the hardware.

**Hardware Guidelines** StreamServe components play well with third party products, but it is easier to troubleshoot systems with fewer moving components. Installing multiple services on the same computer can also cause contention for system resources.

**CPU** Tests show that two different StreamServe Projects on the same hardware can differ thousands of times in performance, depending on the complexity of the Project. This makes it very hard to estimate necessary hardware without testing a pilot Project with real data.

Use:

- Multiple CPUs.
- Multi-threading or multiple servers, that is, scale per job.

**Disk** The core StreamServer is extremely write-intensive. File systems associated with StreamServer working directories and temp directories should be write- optimized, and disk caching should be enabled.

If one machine is used for both the StreamServer repositories and the StreamServer applications, it is recommended to use RAID 10 disks, since they are optimized for write-intensive processes. RAID 5 is optimized for read operations and is not recommended for running StreamServer applications.

If the StreamServer applications and the StreamServer repositories are run on separate machines, ordinary disk striping without parity is sufficient for the machine running the StreamServer applications.

Solid State Disks (SSD) can also dramatically increase performance.

Another factor affecting performance is the speed of SCSI/IDE/FC interfaces, or if you have multiple disks but do not configure them in a RAID array.

The disk sizing depends largely upon data retention times.

**Memory** Some examples of factors affecting the amount of memory to use include:

- Third-party memory managers
- Disk cache
- Network
- Wide Area Networks (WANs)

**Hardware Recommendations for Development Environment** These recommendations are intended for development environments where the following StreamServe components run a single machine:

- Design Center
- StreamServer application used for development
- StreamServe repositories

**Design Center** Design Center can be quite resource-intensive, and is not optimized for thin clients.

**Managing StreamServer Applications** StreamServer applications can be managed with Control Center, through third-party tools, or from a command line. StreamServer applications can be run in a headless configuration, and be managed remotely or locally. Remote management using terminal services is also supported.

**Hardware Recommendations** These are the minimum requirements for a machine running StreamServer applications in the scenario described above. These recommendations are also applicable to virtual machines. However, in this case you must have additional hardware available for the applications running on the physical machine.

Platform	MS Windows
Processor	Dual-core
Memory	4GB or more
Hard Disk	40 GB
Network	Fast Ethernet, TCP/IP
Display	1280x1024, or higher

**Minimum Hardware Requirements for Test Environments** These are the minimum hardware requirements for running StreamServer applications in a test environment. It is assumed that only StreamServer applications run on this hardware, and that Design Center and the StreamServe repositories do not. These recommendations are intended for small environments, since they are only capable of processing relatively small quantities of data.

Platform	Memory
IBM AIX	1 GB per CPU, or more
Red Hat Enterprise Linux	1 GB per CPU, or more
SUSE Linux Enterprise Server	1 GB per CPU, or more
Oracle Solaris	1 GB per CPU, or more
Microsoft Windows	1 GB per CPU, or more

**Hardware Requirements for Initial Production**

These recommendations should satisfy the performance requirements for most production environments involving large volumes of data running through a single StreamServer application in small batches, with a Project of low-to- medium complexity. It is assumed that only StreamServer applications run on this hardware, and that Design Center and the StreamServe repositories do not. Architecturally, a single centralized instance is assumed, with no Wide Area Network bandwidth constraints.

Size	
Large	<ul style="list-style-type: none"><li>● 2 servers with 4 CPUs</li><li>● 8 GB RAM available for StreamServer applications per server.1 GB per CPU, or more</li></ul>
Medium	<ul style="list-style-type: none"><li>● 2 CPUs</li><li>● 8 GB RAM available for StreamServer applications</li></ul>
Small	<ul style="list-style-type: none"><li>● 2 CPUs</li><li>● Minimum of 2 GB RAM available for StreamServer applications</li></ul>

## Hardware Recommendations – Expert Sizing

Enterprise software solutions are highly Project dependent. Hardware sizing, integration testing, and performance tuning in more complex environments are iterative processes, rather than fixed requirements. Throughput-based sizing phases are outlined below.

**Initial Sizing**

- This document provides preliminary suggestions for hardware planning.
- It is recommended to engage OpenText StreamServe Professional Services at this phase, in order to create a custom Project plan.

**Configuration and Project Development**

- Install and configure the StreamServer application.
- Scope and develop StreamServe Projects. Functional requirements and Project complexity should be considered.
- Run a pilot test of the specific Project in the production environment.
- Validate Project functionality. Final document layout is less of a consideration.
- Develop test plans and scenarios for integration, performance, and availability.

- |                             |   |
|-----------------------------|---|
| <b>Testing and Analysis</b> | <ul style="list-style-type: none"><li>● Integration, performance, and stress tests under various scenarios.</li><li>● Verification of availability and disaster recovery.</li><li>● Detect and correct most significant issues in test systems prior to go-live.</li></ul>                                |
| <b>Re-Optimization</b>      | <ul style="list-style-type: none"><li>● Periodic system reevaluations should be performed as customer specifications evolve and new applications come online.</li><li>● Re-sizing is often necessary to compensate for changes in network topology, business processes, or increased user load.</li></ul> |

## Enterprise Configurations Requirements

Enterprise configurations usually characterized by a combination of the following requirements:

- High availability
- Failover and redundancy
- Extremely large volumes of data
- Very large batch files
- Many concurrent users or input streams
- Very low real-time latency
- Complex StreamServe Projects, involving extensive:
  - Scripting
  - Post-processing
  - Dynamic database lookups
  - Business logic
- Wide Area Network bottlenecks
- Many output devices
- Distributed output devices
- Assured delivery
- Encryption and high security

### Typical Volumes

Typical StreamServe volumes are as follows:

**Large Installation** About 10 million documents per month.

**Medium Installation** 1-2 million documents per month.

**Example Configurations**

The table below shows example configurations. It is assumed that only StreamServer applications run on this hardware, and that the StreamServe repositories are run on separate hardware.

There should be approximately 2 to 3 GB of free memory available for each StreamServer application.

Size	
Very Large	<ul style="list-style-type: none"><li>• Latency requirements for real-time documents</li><li>• 4 servers with 4 CPUs each</li><li>• 16 GB RAM available for StreamServer applications per server</li><li>• Connected to a high-performance I/O system</li></ul>
Large	<ul style="list-style-type: none"><li>• 2 servers with 4 CPUs</li><li>• 8 GB RAM available for StreamServer applications per server</li></ul>
Medium	<ul style="list-style-type: none"><li>• 2 CPUs</li><li>• 8 GB RAM available for StreamServer applications</li></ul>
Small	<ul style="list-style-type: none"><li>• 2 CPUs</li><li>• Minimum of 2 GB RAM available for StreamServer applications</li></ul>

## Sizing Considerations

The hardware resources required to process documents depend on a number of factors, such as:

- The number of documents produced.
- The average size of each document.
- How much time is available to process the job(s).

**Throughput Based Sizing Considerations**

An optimal StreamServer configuration can provide linear scalability in terms of added processing power (CPU clock frequency or more individual CPUs). However, this assumes that i/o bottlenecks are avoided.

**Database Management System Sizing**

The Database Management System (DBMS) needs individual sizing and should be connected to StreamServer using a fast, reliable network connection. As a general rule, the DBMS can be sized down to half of the StreamServer sizing requirements, but should be optimized for high-write performance.

## Other Considerations

### Separate Physical Servers

In general, StreamServer applications should be run on a separate server, independent of other enterprise applications. Although StreamServer applications can share resources with other applications, isolating them enables you to:

- Simplify troubleshooting
- Reduce resource contention
- Increase scalability
- Increase consistency of throughput

With smaller implementations it is often difficult to justify.

### Choosing a Platform

The choice of platform tends to be more political than technical. In most organizations, there are more Windows administrators and users may find it easier to use. However, UNIX variants can be more robust and support rolling upgrades.

Most customers run StreamServer applications on the same platform as their source application. Although it is not necessary to run on the same platform, resolving system issues tends to be easier when working with only one set of system administrators.

There are some technical considerations that may affect hardware decisions. While StreamServer applications run on many platforms, Design Center only runs on Windows platforms. Different StreamServe components can run on different platforms, but in smaller implementations, where resource consolidation is desirable, this can be a factor.

Always have expert knowledge and experience available within the company for the platform you choose.

## Database Requirements for StreamServe Repositories

A database installation is required for the StreamServe repositories. For a list of supported database vendors and versions, see Databases in the Supported platforms and software documentation.

All the StreamServe repositories must use the same database vendor. For example, if the enterprise repository runs on Oracle, the runtime repository, the Collector archive, and the web content repository must also run on Oracle.

Before creating the StreamServe repositories, you must configure the database installation for use with StreamServe components. This section describes the minimum database requirements that must be fulfilled in a development or testing environment.

**Database Configuration for Microsoft SQL Server** This section describes the minimum database requirements that must be fulfilled before creating StreamServe repositories in a testing environment.

For detailed information, see OpenText StreamServe Microsoft SQL Server Database Guidelines, available at OpenText Knowledge Center.

- Database Configuration**
- Enable TCP/IP.
  - Specify Mixed Mode authentication.
  - For all IP addresses, specify the static TCP port number that you use to access the database.



Do not specify any dynamic TCP ports.

---

- Select a proper collation.

**SQL Server Browser Service for Named Instances** If you use a named instance of SQL Server, the SQL Server Browser service must be started before you create the StreamServe repositories.

**Database Configuration for Oracle Database** This section describes the minimum database requirements that must be fulfilled before creating StreamServe repositories in a testing environment.

For detailed information, see OpenText StreamServe Oracle Database Database Guidelines, available at OpenText Knowledge Center.

**Database Configuration** • Specify the following database server parameter:

Parameter	Value
db_block_size	<ul style="list-style-type: none"> <li>Must be at least 8192.</li> <li>If the database server has more than 4GB memory, you can raise this value to 16384.</li> </ul>

- Create a UTF database with:
  - Character set AL32UTF8.
  - National character set AL16UTF16.
- Make all tablespaces locally managed.
- Where applicable, make tablespaces automatic segment space managed (SEGMENT SPACE MANAGEMENT AUTO, which is default on Oracle 11g).
- For Oracle 12c, create a service name in tnsnames.ora for every PDB (pluggable database) you want use for a StreamServe repository.

**Increasing Unicode Support** If the database character set uses a multi-byte character encoding scheme and the default Unicode support is not enough, you can increase the Unicode support by specifying the following database server parameter:

Parameter	Value
nls_length_semantics	<ul style="list-style-type: none"> <li>CHAR</li> <li>The char value requires more space in the database than the default byte value.</li> </ul>

## Managing License Files

Unlicensed StreamServer applications run in demo mode by default. In demo mode, texts in the output are randomly removed and replaced by the text string Demo. To license StreamServer applications, you must get a license file and connect the license file to the StreamServer applications.

Open Text Internal Use Only  
Do Not Distribute

## 4. Installing and Upgrading StreamServe

On completion of this chapter, participants should be able to:

- Describe a development and production installation topology
- Install StreamServe
- Remove StreamServe components
- Upgrade StreamServe
- Install a StreamServe hotfix

### Installing on a Computer Used for Development and Testing

This section describes the main steps to install StreamServe components on a computer used to develop StreamServe Projects, and run StreamServe applications for development and testing purposes.

In this scenario, you install and configure the following StreamServe components:

- Framework and StreamServer
- Management gateway
- Control Center
- Design Center
- Application domain
- StreamServe Enterprise Repository
- Runtime repository
- Service gateway

You create the repositories in a local database. For example, Microsoft SQL Server Express.

You can also install the StreamStudio components and OpenDS on the same computer.

## Required Installation Steps

In order to install StreamServe you need to:

1. Download the Setup.

You can download the OpenText StreamServe Setup from the StreamServe Product area on OpenText Knowledge Center <https://knowledge.opentext.com>.

2. Preparations

Before you begin, you must:

- Confirm you have the software and hardware that is required for each StreamServe Component you want to install. See supported platforms and software documentation.
  - Install any third-party software that is required. For example, SQL Server Express, Apache Tomcat, etc.
  - Prepare the database server that you will use to host the StreamServe repositories. See “Database Requirements for StreamServe Repositories” on page 3-11.
3. Install Framework and StreamServer using the Setup wizard.
  4. Install Control Center using the Setup wizard.
  5. Install Design Center using the Setup wizard.
  6. Install StreamStudio using the Setup wizard. See [Installing StreamStudio on Windows](#) on page 41.
  7. (Optional) – Install OpenDS.

## Installing in a Production Environment

This section describes the main steps to install StreamServe components in a production environment.

### Using a Central Enterprise Repository

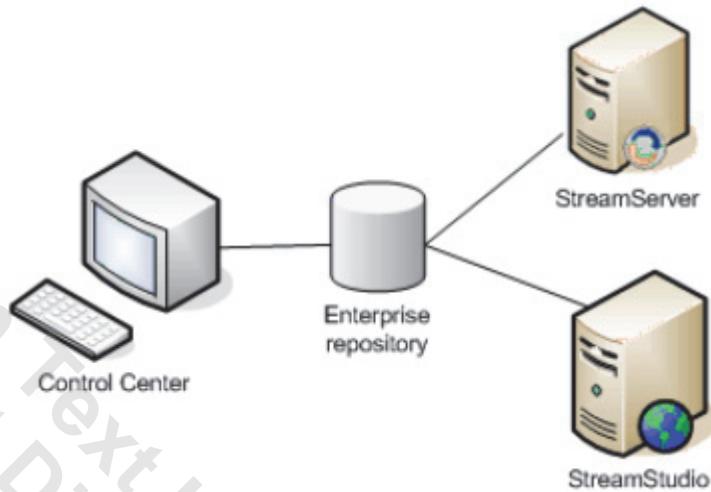
When you run StreamServe software in a production environment, you create one central StreamServe Enterprise Repository. You then connect Framework and StreamServer, and Control Center to the central enterprise repository.

For performance and scalability reasons, you may want to install and run Framework and StreamServer on more than one computer. In this scenario, you must connect all the installations to the central enterprise repository.

If you want use Control Center to deploy the StreamStudio web application files, you must install Framework and StreamServer on the same computer as StreamStudio and then connect the computer to the central enterprise repository.

Figure 4-1:

StreamServer and Control Center Using a Central Enterprise Repository



## Removing StreamServe Components

You can remove the following components from Windows Control Panel:

- Framework and StreamServer
- Control Center
- Design Center
- StreamStudio web applications
- Database Administration Tool



Windows services for StreamServe applications are removed when you remove Framework and StreamServer.

Design Center Projects are not deleted when you remove Design Center.

The StreamServe repositories are not deleted when you remove StreamServe software. You must use an external tool if you want to delete the StreamServe repositories.

## Upgrading StreamServe Components

The procedure to upgrade to OpenText StreamServe 5.6.2 depends on StreamServe version from which you are upgrading.

The OpenText StreamServe 5.6.2 Upgrading Instructions Reference Guide available in <http://knowledge.opentext.com> describes the upgrade procedures from StreamServe 4.1.2 SPx, Persuasion SP3, SP4 and SP5, OpenText StreamServe 5.6 and 5.6.1.

In general, the upgrade process consists of the following steps:

1. Create a 5.6.2 environment
2. Upgrade projects configured to store documents
3. Upgrade projects configured to collect documents
4. Migrate repositories
  - a. Migrate security data
  - b. Migrate web content repositories
  - c. Migrate stored messages
  - d. Migrate document definitions

## Installing and Applying StreamServe Hotfixes

Installing a hotfix means that files in the OpenText StreamServe installation folder and the Management Gateway base directory are replaced. If the hotfix includes database changes, you must also apply the hotfix to the StreamServe repositories. If the hotfix includes changes to the web application archive (WAR) files, you must redeploy the WAR files to the Java application server.

### About StreamServe Hotfixes

Each hotfix incorporates all previously released bug fixes for a specific OpenText StreamServe release. You must install all hotfixes included in the hotfix package, that is you cannot install specific fixes only.

All hotfixes for a specific release will be included in the next service pack for this release.

Each hotfix is made available for all applicable platforms.

Hotfixes are distributed by OpenText Customer Support. The Customer Support team keeps track of all hotfixes distributed, and will provide any additional information you may need to apply the hotfix.

**Hotfix Build Number** Each hotfix is identified by the release to which it is applicable and a build number, for example 5.6.2 Hotfix Build nnn.

The build number is a unique number used internally by OpenText to identify specific hotfixes and releases. Any time you refer to a specific hotfix, you should use the release name and the build number to avoid any misunderstandings.

The latest hotfix for a specific release is always the one with the highest build number. Build numbers are neither related to the number of hotfixes built, nor to the hotfix contents.

## Exercise - Installing StreamServe



### Install the Framework and StreamServe

1. Create a folder called C:\InstallFiles.
2. Unzip C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\5.6.2\_GA\_2014-12-19\_Build\_229.zip to C:\InstallFiles.
3. Launch the Setup application located in C:\InstallFiles\5.6.2\_GA\_2014-12-19\_Build\_229.
4. In the Setup application window select **Framework and StreamServer**.

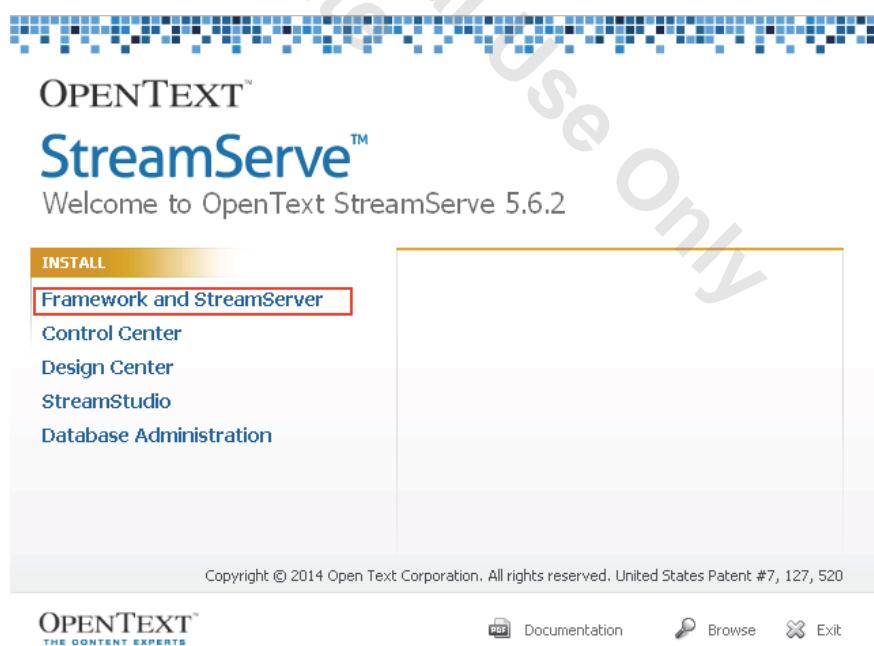


Figure 4-2:  
Setup Application - Welcome Page

5. In the OpenText StreamServe Framework and StreamServer 5.6.2 - InstallShield window select the **Next** button.

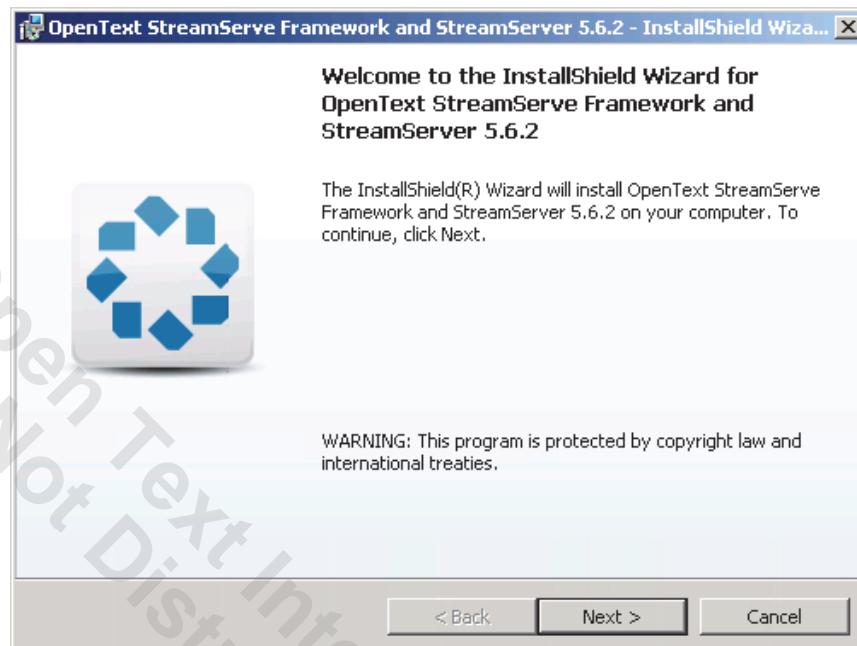


Figure 4-3:

Setup Application -  
OpenText StreamServe  
Framework and  
StreamServer Welcome  
Page

6. In the Installation Information window select the **Next** button.

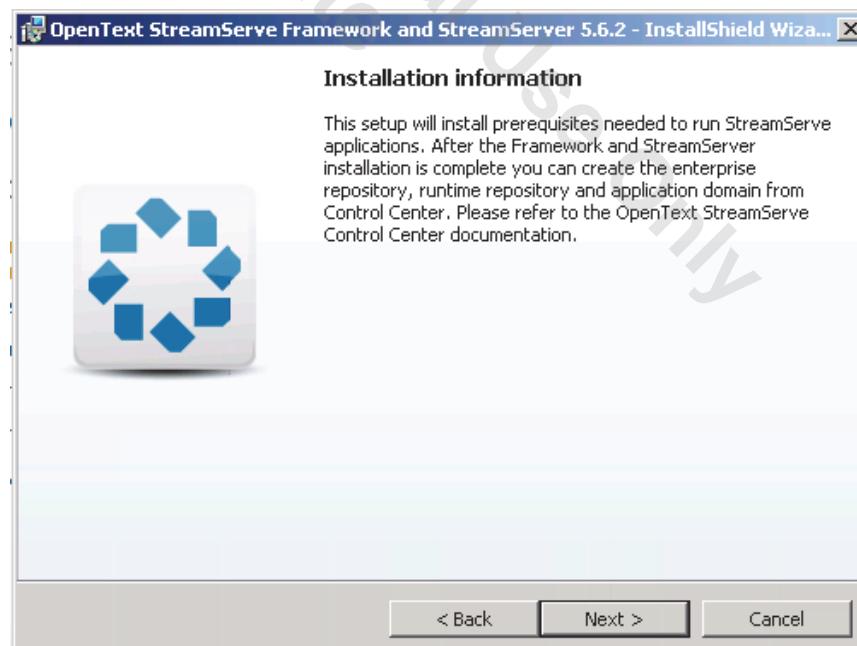
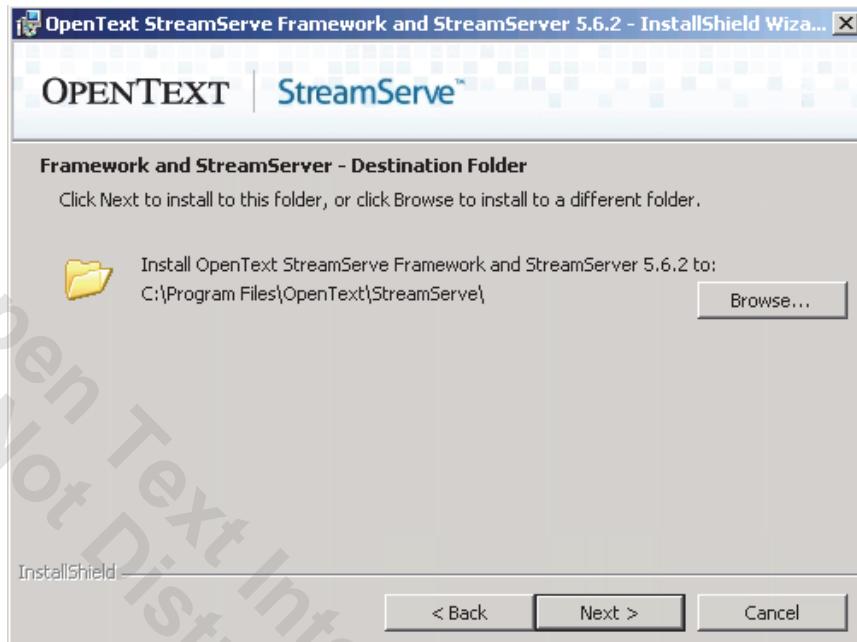


Figure 4-4:

Setup Application -  
Information Window

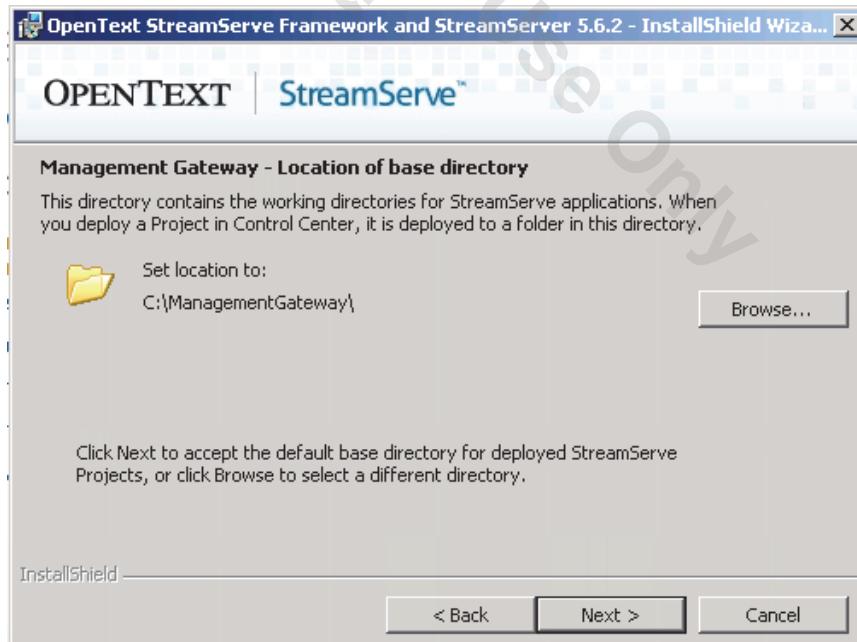
7. In the Framework and StreamServer - Destination Folder leave the default folder and select the Next button.

**Figure 4-5:**  
Setup Application -  
Framework and  
StreamServer Destination  
Folder



8. In the Framework and StreamServer - Destination Folder leave the default folder and select the Next button.

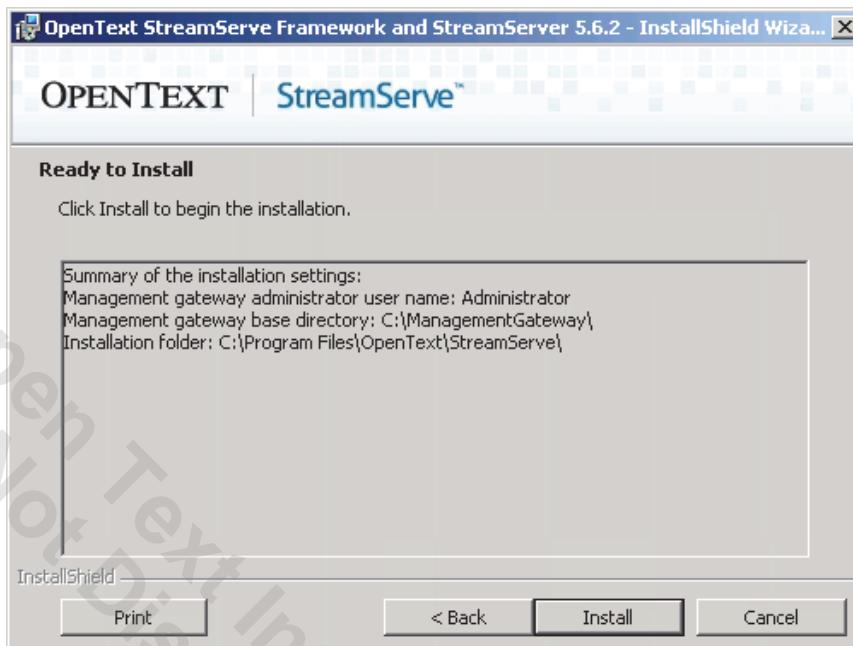
**Figure 4-6:**  
Setup Application -  
Management Gateway  
Destination Folder



9. In the summary window select the **Install** button.

Figure 4-7:

Setup Application -  
Installation Summary

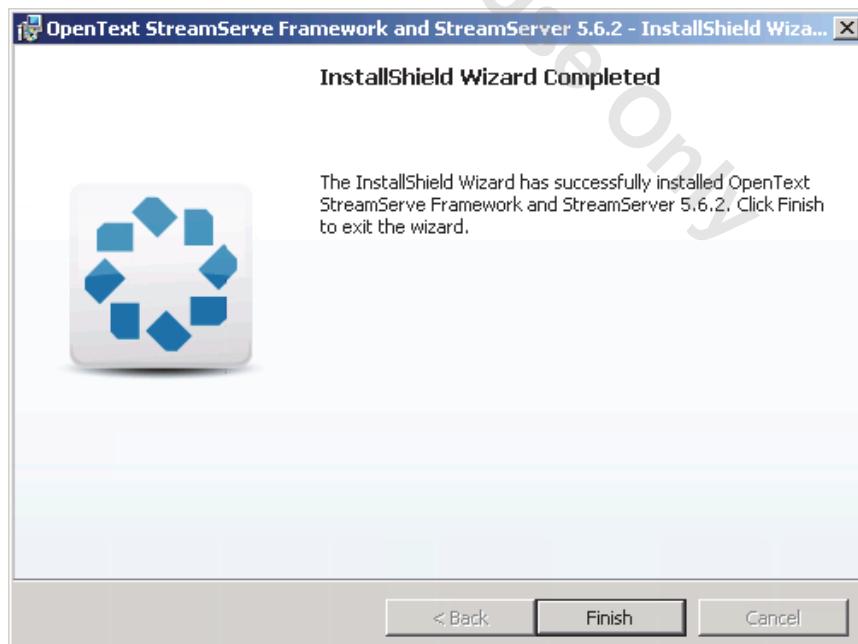


The Framework and StreamServe installation process starts and will take a few minutes.

10. Once the installation is completed select the **Finish** button.

Figure 4-8:

Setup Application -  
Installation Complete



You are taken back to the Setup application window. Proceed with the next installation step: Install Control Center.

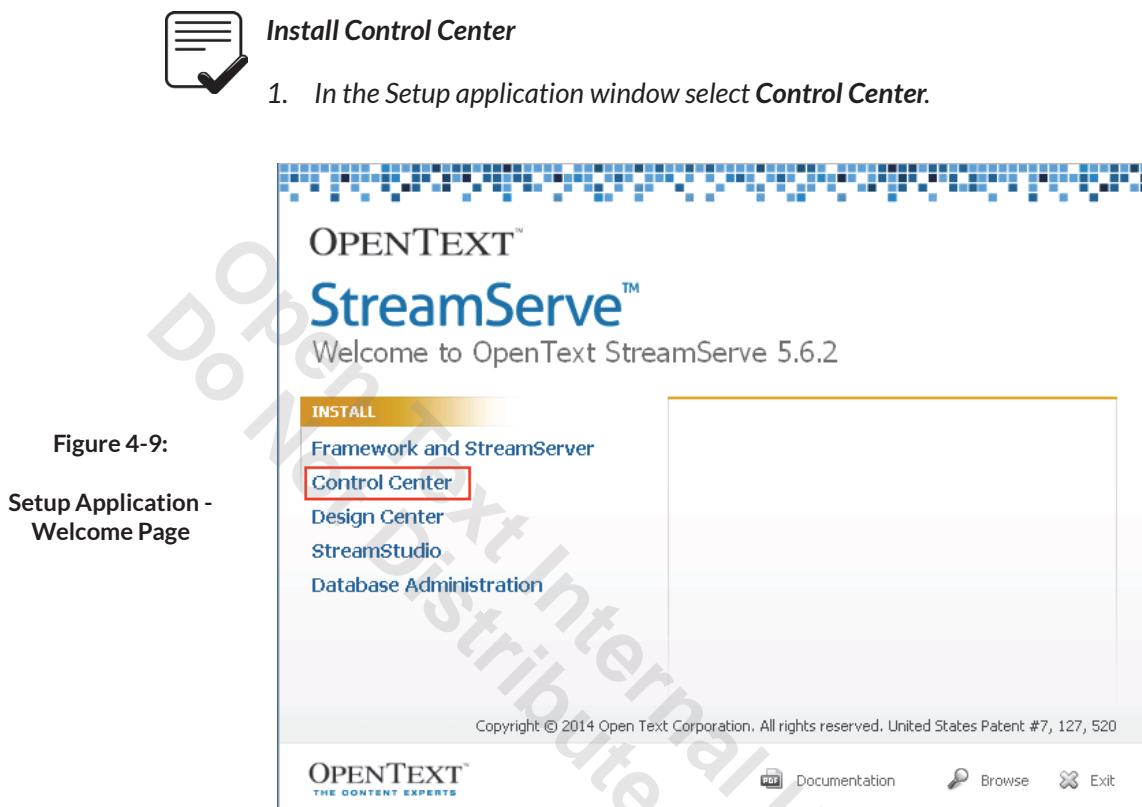


Figure 4-9:

Setup Application -  
Welcome Page

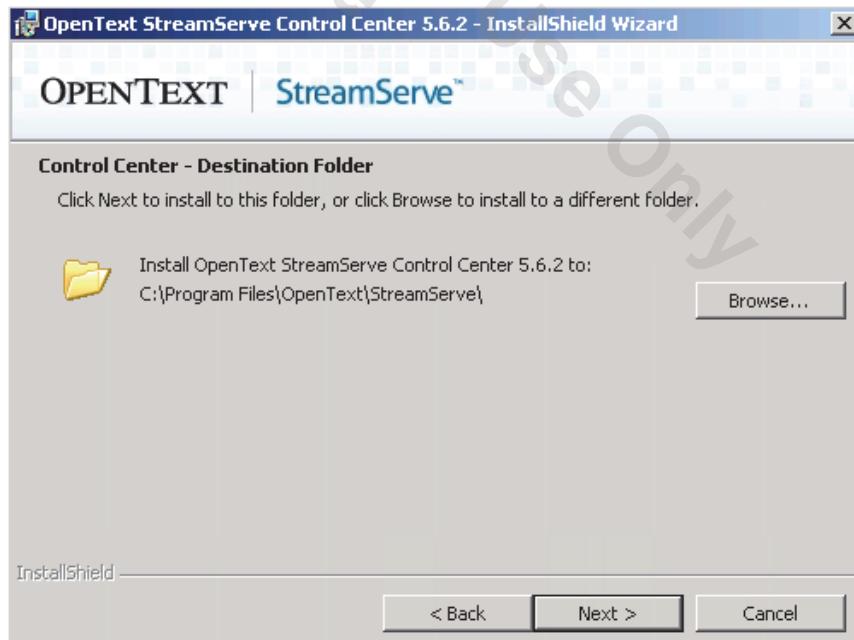
2. In the OpenText StreamServe Control Center Welcome window select the **Next** button.

Figure 4-10:  
Setup Application -  
OpenText StreamServe  
Control Center Welcome

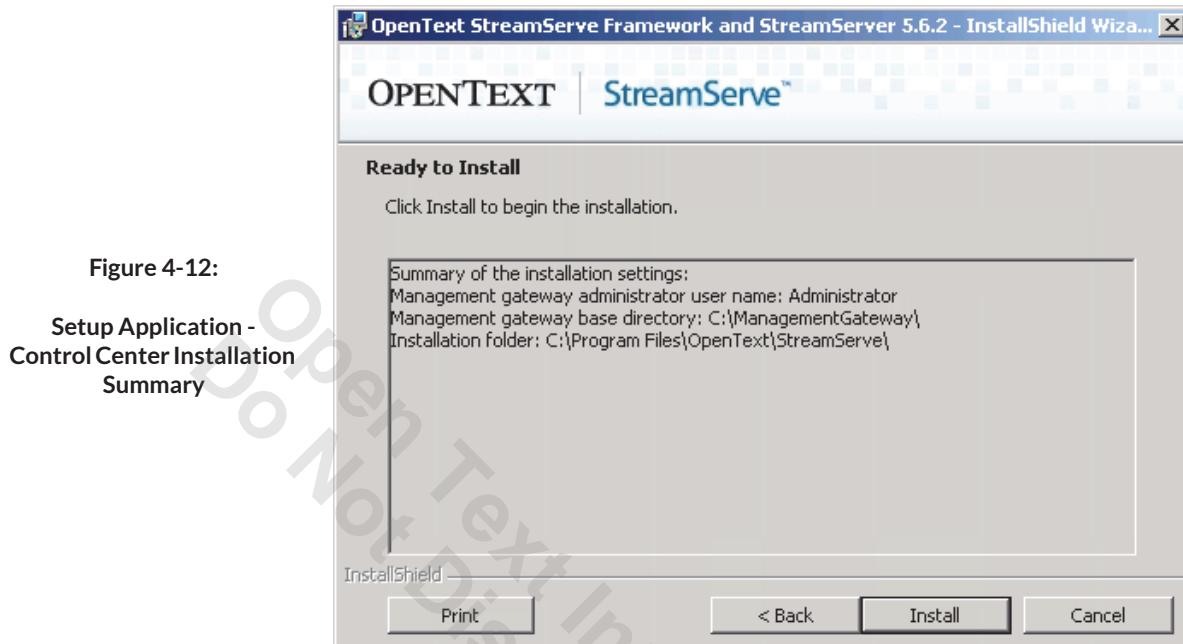


3. In the Control Center - Destination Folder leave the default folder and select the **Next** button.

Figure 4-11:  
Setup Application -  
Control Center  
Destination Folder



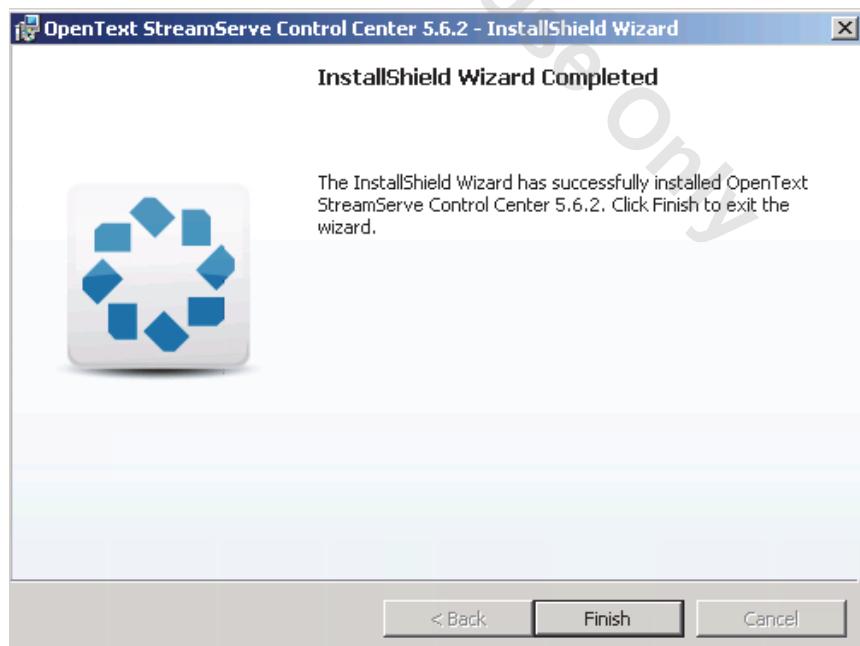
4. In the Control Center summary window select the **Install** button.



**Figure 4-12:**  
Setup Application -  
Control Center Installation  
Summary

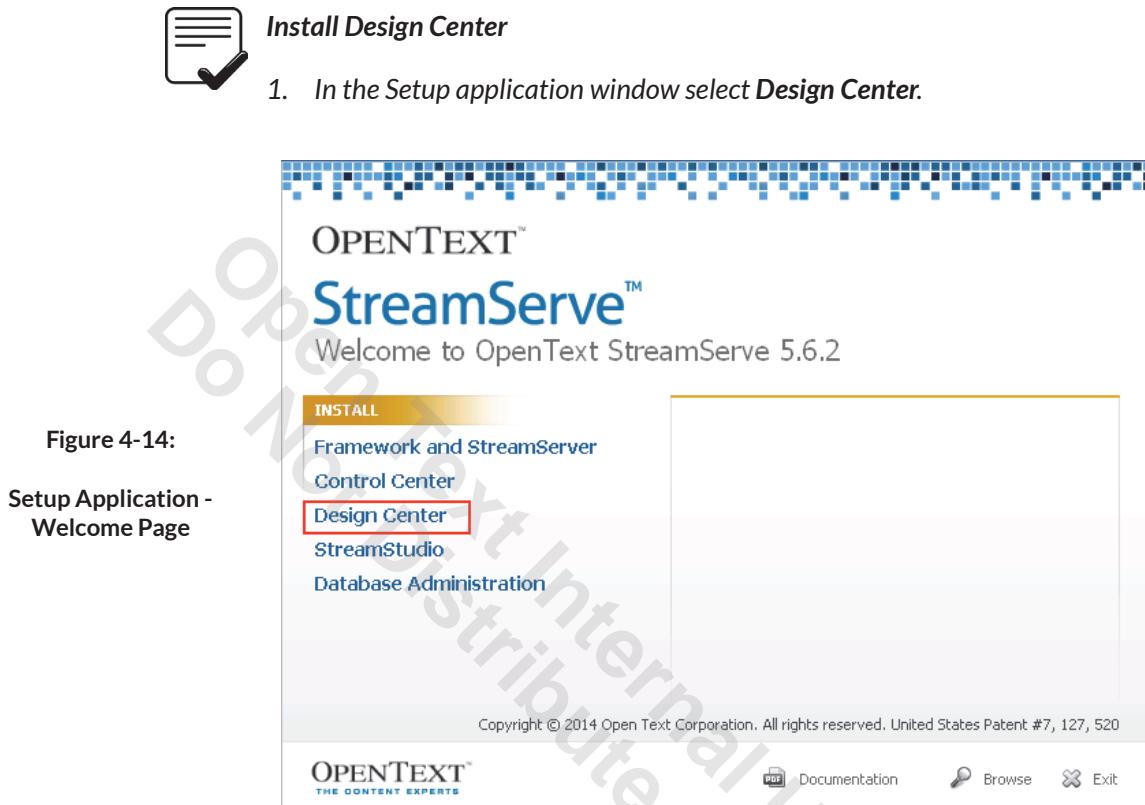
The Control Center installation process starts and will take a few minutes to complete.

5. Once the installation is completed select the **Finish** button.



**Figure 4-13:**  
Setup Application -  
Control Center Installation  
Complete

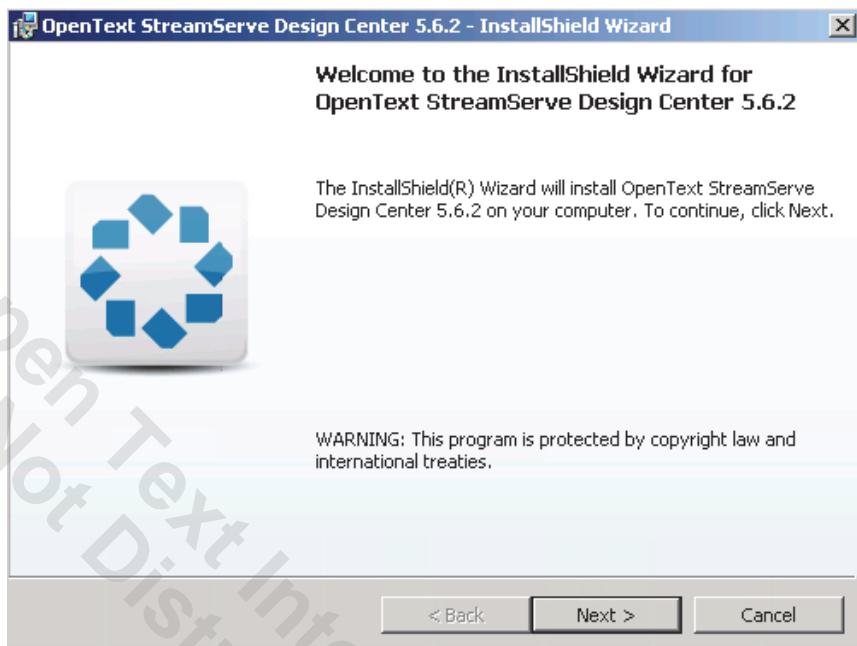
You are taken back to the Setup application window. Proceed with the next installation step: Install Design Center.



**Figure 4-14:**  
Setup Application -  
Welcome Page

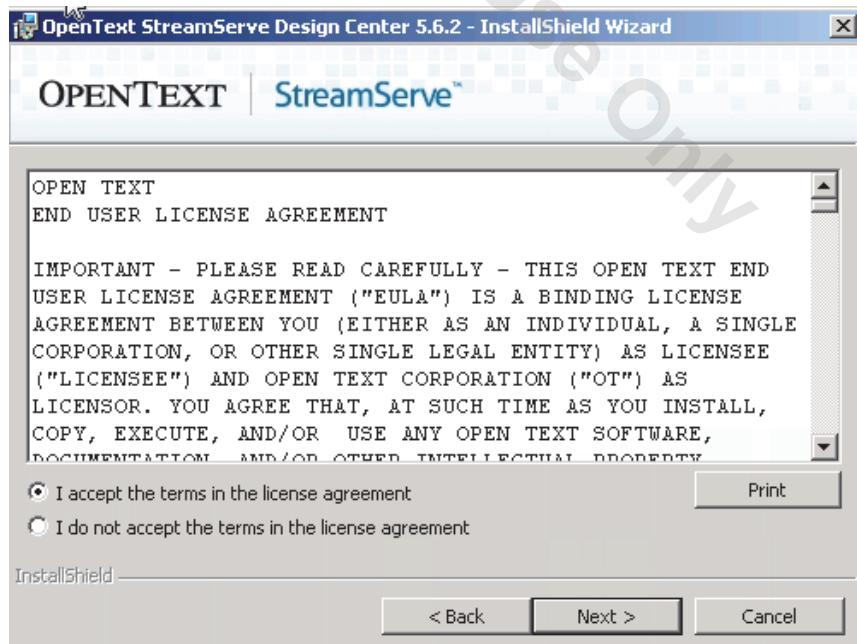
2. In the OpenText StreamServe Design Center Welcome window select the **Next** button.

Figure 4-15:  
Setup Application -  
OpenText StreamServe  
Design Center Welcome



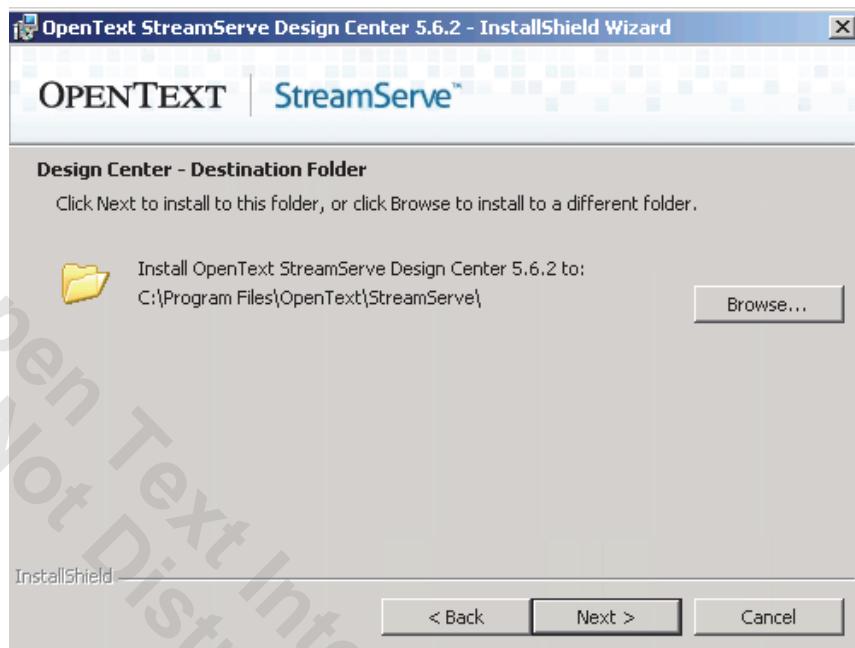
3. In the OpenText StreamServe Design Center End User License Agreement window accept the terms in the license agreement and then select the **Next** button.

Figure 4-16:  
Setup Application -  
OpenText StreamServe  
Design Center License  
Agreement



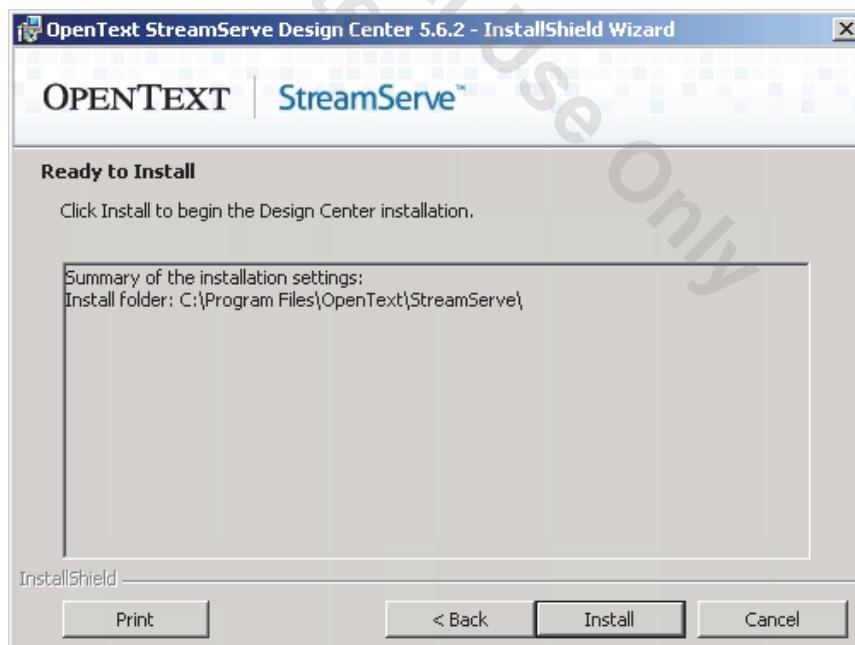
4. In the Design Center - Destination Folder leave the default folder and select the **Next** button.

**Figure 4-17:**  
Setup Application - Design  
Center Destination Folder



5. In the Design Center summary window select the **Install** button.

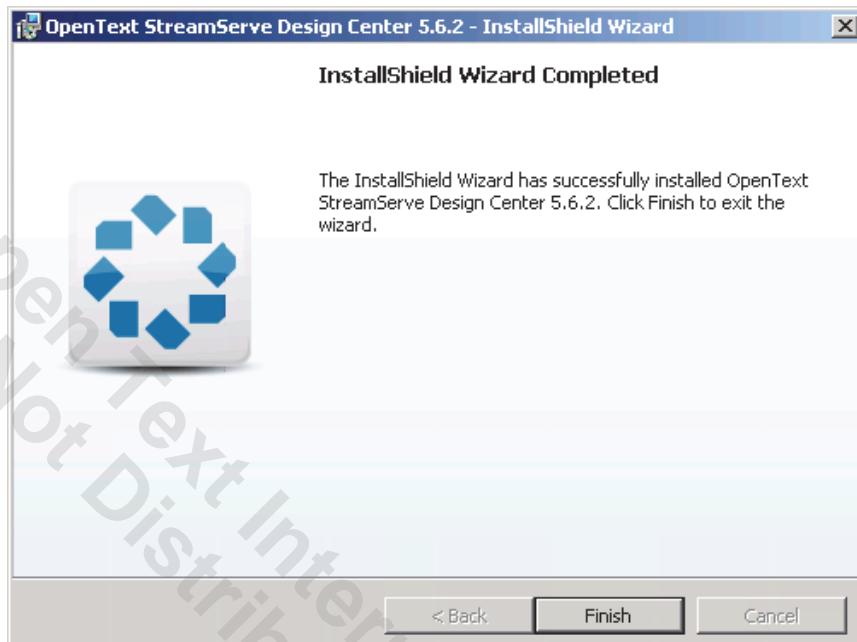
**Figure 4-18:**  
Setup Application - Design  
Center Installation  
Summary



The Design Center installation process starts and will take a few minutes to complete.

6. Once the installation is completed select the **Finish** button.

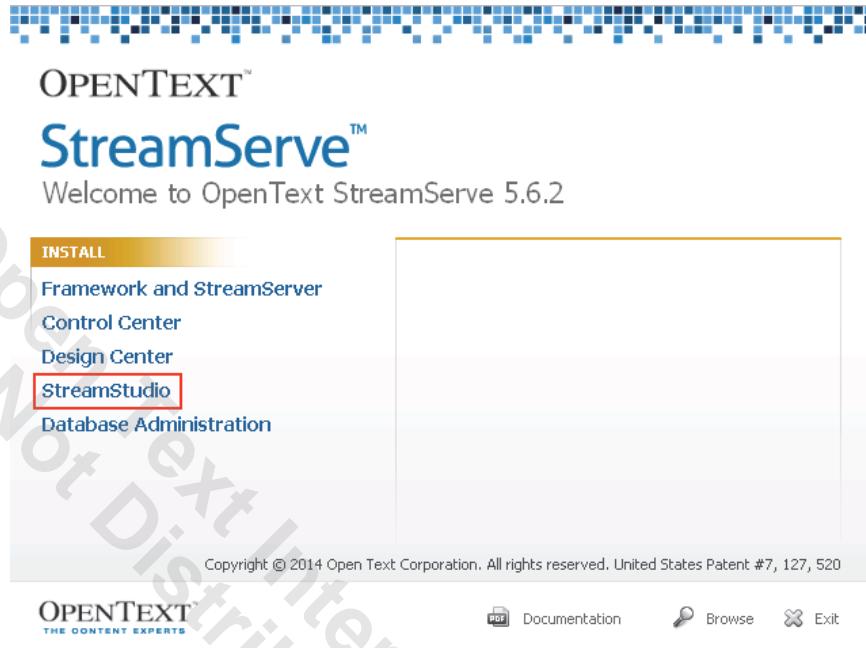
Figure 4-19:  
Setup Application - Design  
Center Installation  
Complete



7. You are taken back to the Setup application window. Proceed with the next installation step: Install StreamStudio.

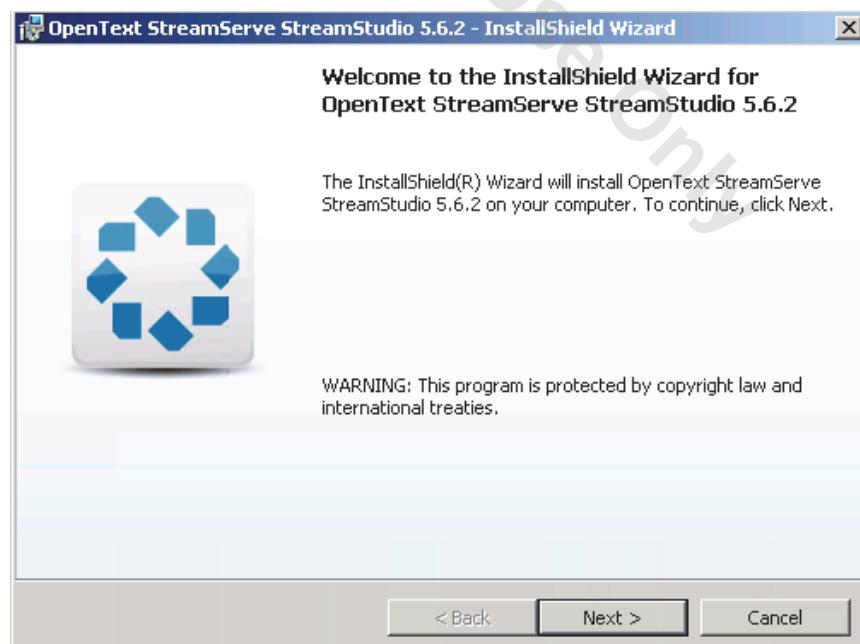
-  **Install StreamStudio**
1. In the Setup application window select **StreamStudio**.

Figure 4-20:  
Setup Application -  
Welcome Page



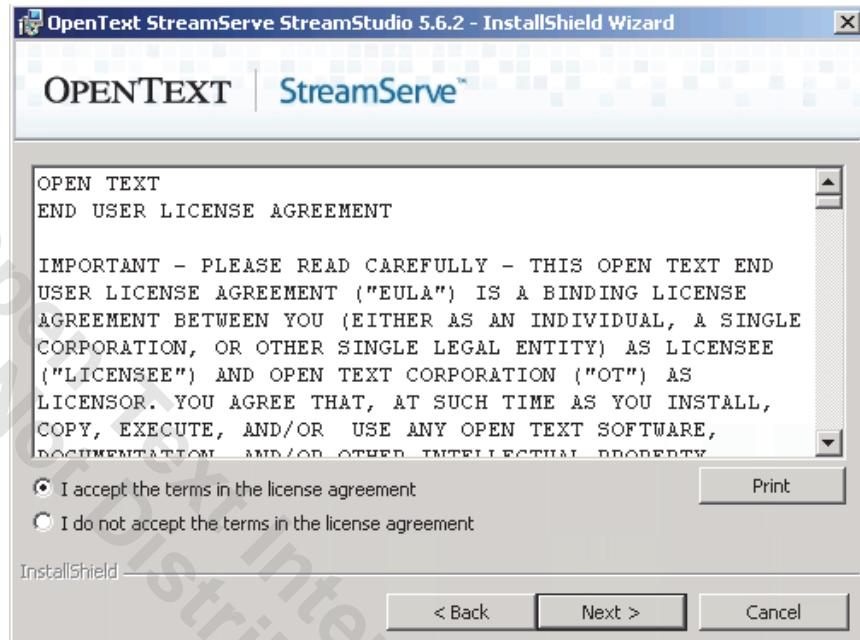
2. In the OpenText StreamServer StreamStudio Welcome window select the **Next** button.

Figure 4-21:  
Setup Application -  
OpenText StreamServer  
StreamStudio Welcome



3. In the OpenText StreamServe StreamStudio End User License Agreement window accept the terms in the license agreement and then select the Next button.

**Figure 4-22:**  
Setup Application -  
OpenText StreamServe  
StreamStudio License  
Agreement



4. In the StreamStudio summary window select the **Install** button.

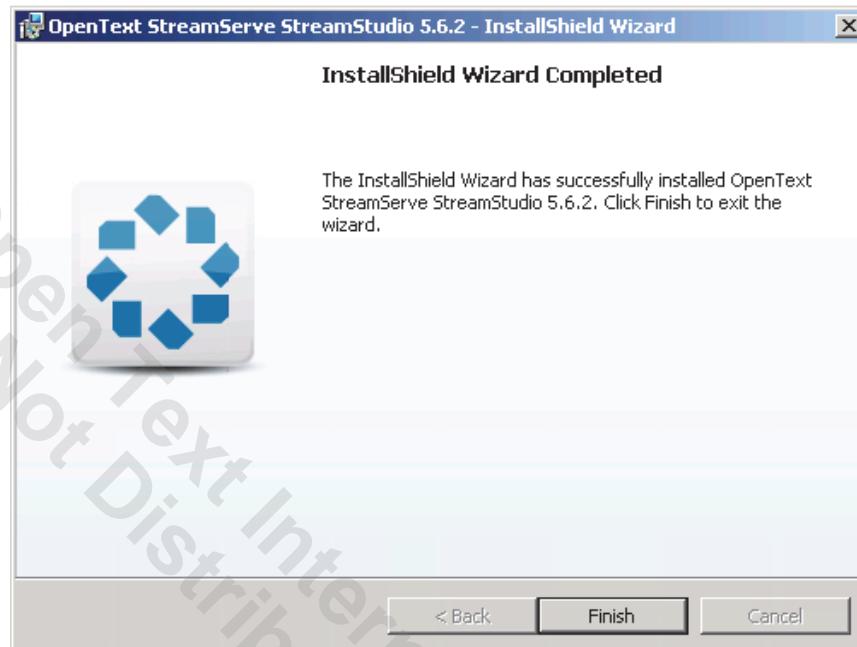
**Figure 4-23:**  
Setup Application -  
StreamStudio Installation  
Summary



The StreamStudio installation process starts and will take a few minutes to complete.

5. Once the installation is completed select the **Finish** button.

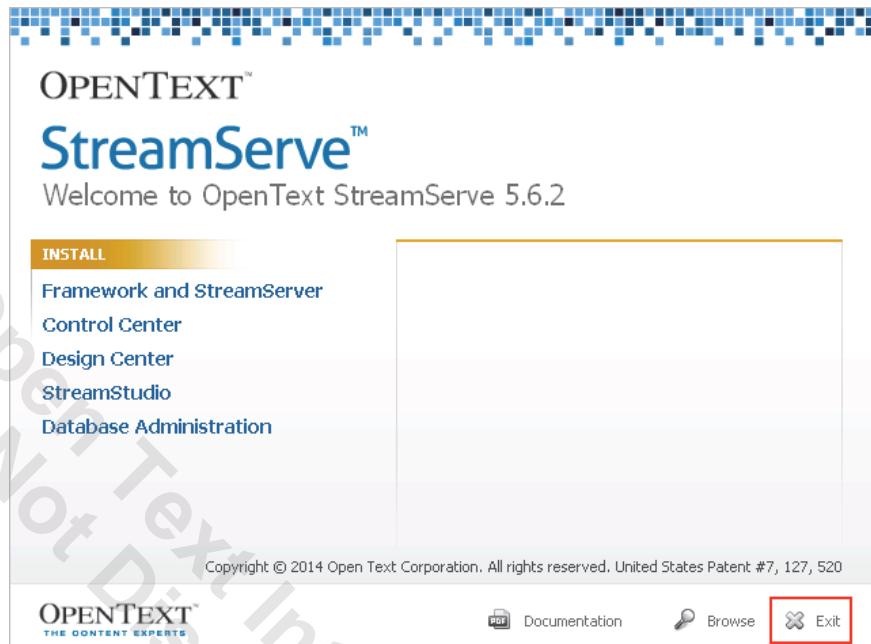
Figure 4-24:  
Setup Application -  
StreamStudio Installation  
Complete



You are taken back to the Setup application window.

6. In the StreamServe welcome window click the **Exit** link.

**Figure 4-25:**  
**Setup Application - Exiting  
the Welcome Page**



Open Text Internal Use Only  
Do Not Distribute

## 5. After Installation Configuration

On completion of this chapter, participants should be able to:

- Describe the configuration steps required after StreamServe installation
- Define and create Application Domains
- Define and create Runtime Repositories
- Add StreamServer Applications

### Configuring StreamServe after Installation

This section describes the steps that are required after the installation to get started in Control Center, create the enterprise repository, and run StreamServer applications.

It also describes the steps that are required to run StreamStudio web applications.

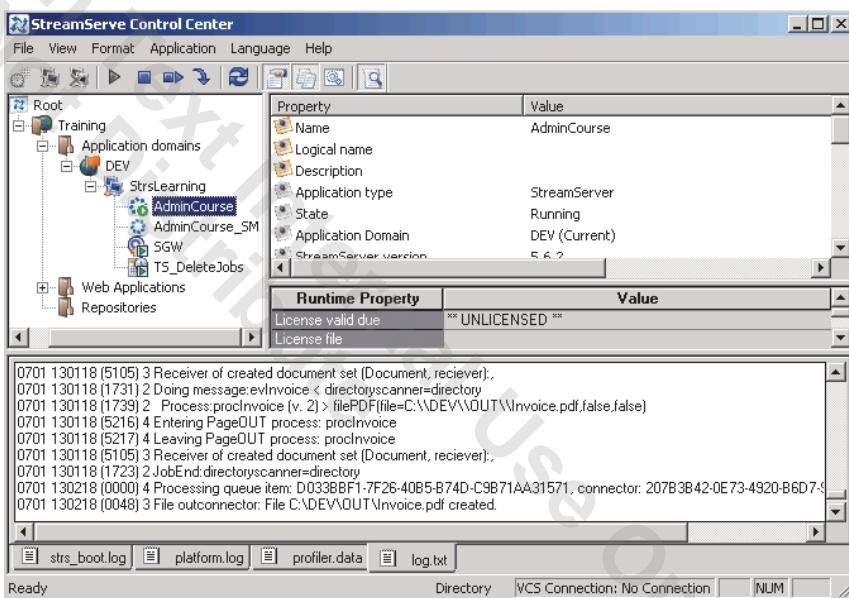
1. Get started in Control Center
  - a. Open Control Center and enter a new password for the Management Gateway Administrator (user name Administrator).
  - b. Control Center, specify the connection settings for the enterprise repository.
  - c. In Control Center, rename the site.
2. Create the StreamServe Enterprise Repository  
In Control Center, create the enterprise repository. See Creating StreamServe Enterprise Repository in the Control Center documentation.
3. Prepare to run StreamServer Applications
  - a. In Control Center, configure an application domain.
  - b. In Control Center, create the runtime repository.
4. Prepare to Run StreamStudio Applications
  - a. Configure the application domain for StreamStudio.
  - b. If required, create the repositories needed for the StreamStudio web applications you will run.
  - c. Add the service gateway application(s) for StreamStudio.
  - d. Deploy the StreamStudio web portal to the Java application server.

5. Prepare to Run Ad Hoc and Reviewer Applications
  - a. Configure the application domain for StreamStudio.
  - b. If required, create the repositories needed for the Ad Hoc and Reviewer web applications you will run.
  - c. Add the service gateway application(s) for Ad Hoc and Reviewer.
  - d. Deploy the Ad Hoc and Reviewer web portal to the Java application server.

Most of these tasks are performed in Control Center.

## About Control Center

**Figure 5-1:**  
Control Center



Control Center is an administration tool used to deploy, run and administer StreamServe applications.

You can administer the following StreamServe applications from Control Center:

- StreamServer (application used to run Design Center Projects)
- Archiver
- Service gateway
- Task Scheduler

You use Control Center to create the StreamServe repositories. You can either create the repositories directly from the Control Center (for Microsoft® SQL Server and Oracle® Database) or you can generate database scripts to be executed later in an external tool.

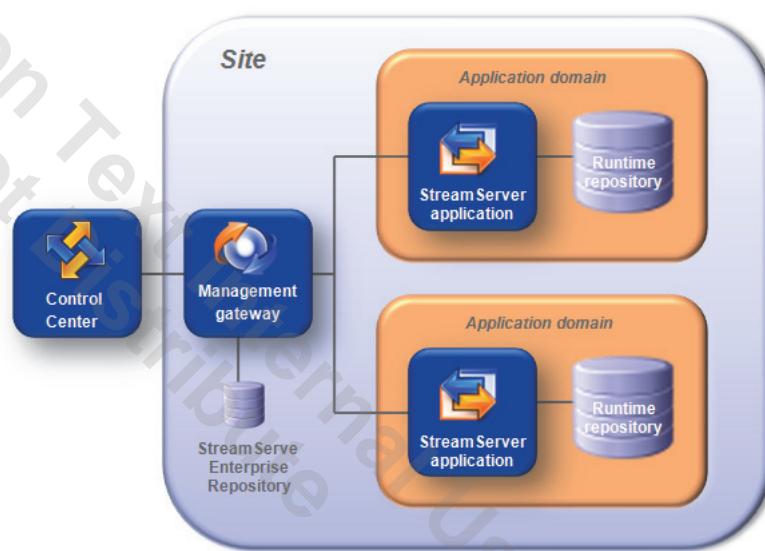
You can apply database hotfixes to the StreamServe repositories directly from Control Center.

You can use Control Center to run applications and create repositories on both UNIX and Windows hosts.

From Control Center, you can administer application on both the local host and remote hosts.

## StreamServer Components Recap

Figure 5-2:  
Control Center



**Site** Each company or organization is represented by a site in Control Center. From the site, you configure computers used to run StreamServe applications, application domains, and StreamServe applications.

**Enterprise Repository** This is a database used to store information about computers, StreamServe applications, and application domains for a company or organization.

**Management Gateway** This manages and monitors the StreamServe applications on a computer, and also connects the computer to the enterprise repository.

**Application Domain** This is a group of one or more StreamServe applications. All applications in one domain share the same runtime repository.

**Runtime Repository** This database stores jobs and job related information produced by the StreamServer applications in one application domain. If you run StreamStudio, the repository is also used to store security profiles and web access information.

**StreamServer Application** This type of application runs a Design Center Project.

## Planning Your Application Domains

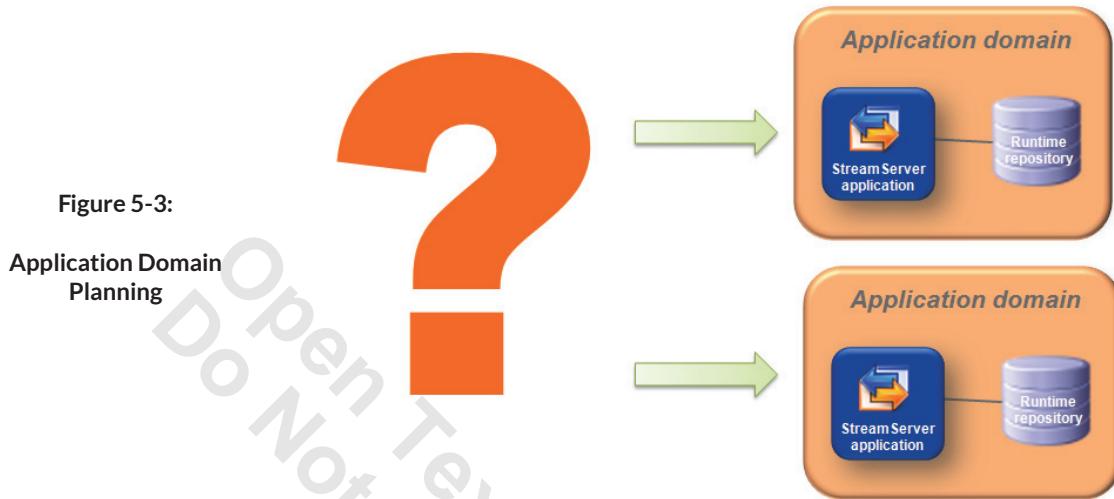


Figure 5-3:

Application Domain Planning

Before getting started in Control Center, you should decide how you want to organize your StreamServe applications into application domains.

When planning how to group your applications into an application domain, you should consider the following:

- All applications in one application domain share the same:
  - Runtime repository (if you run Composition Center applications the applications also share the same web content repository).
  - Document types.
  - Channels for reprocess and preview, etc
  - Composition Center templates.
  - Directory server(s) for the internal and external user directories.
  - Web security settings.
- You can use different computers to run applications in the same domain. Or the same computer to run applications in different domains.
- All the StreamServe repositories must use the same database vendor.

## Application Domains – Typical Scenario

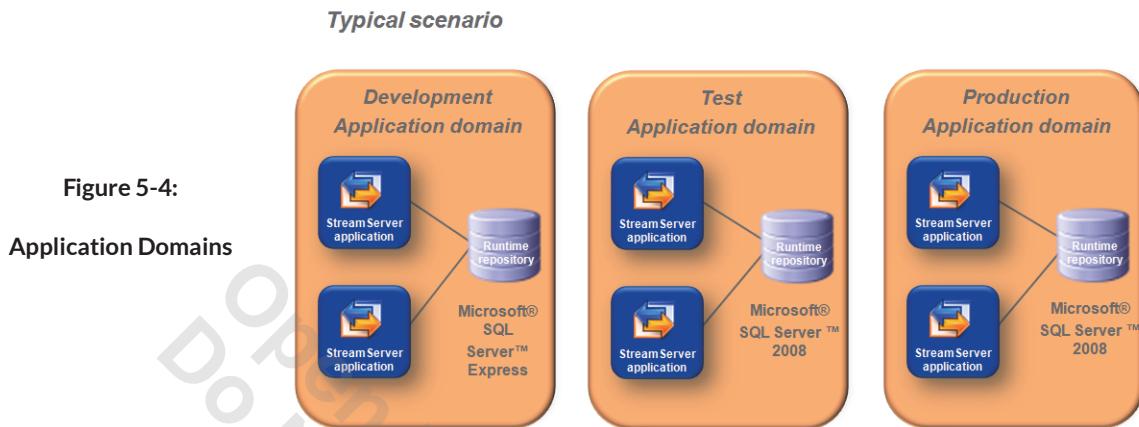


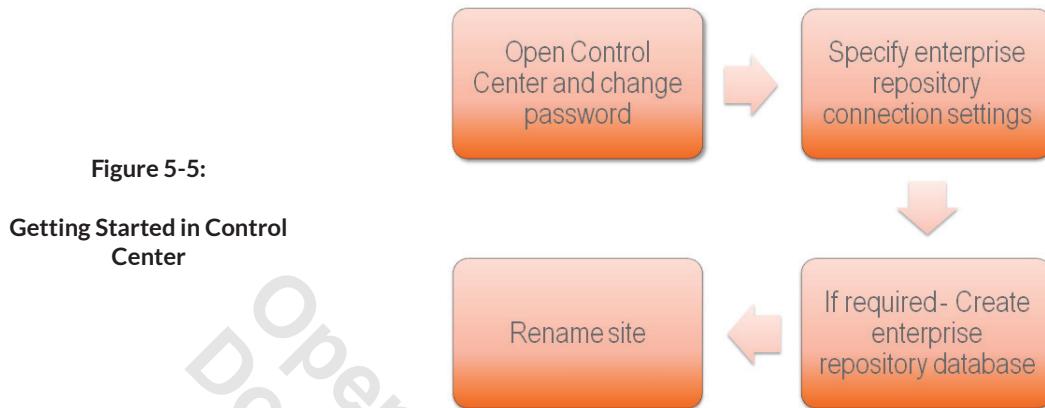
Figure 5-4:  
Application Domains

A typical scenario is to have three application domains, which are used for the following environments:

- Development
- Testing
- Production

<b>Development Environment = Development Application Domain</b>	<p>During the development phase, Projects are deployed to development StreamServer applications in the development application domain.</p> <p>In the development environment, SQL Server Express can be used for the runtime repository. Both the StreamServer applications and the SQL Express database can be run on a developer's laptop.</p>
<b>Test Environment = Test Application Domain</b>	<p>When a Project is ready to enter the formal testing phase, it is deployed to a test StreamServer application in the test application domain.</p> <p>Individual company requirements determine the hardware used for the StreamServer applications and runtime repository in the test environment. Ideally, the test environment should be as similar as possible to Production environment in order to accurately test performance.</p>
<b>Production Environment = Production Application Domain</b>	<p>When a Project is ready to run in production, it is deployed to the production application domain.</p> <p>In the production environment, Microsoft SQL Server is used for the production runtime repository. This repository only contains real documents (no test data).</p>

## Getting Started in Control Center



The following tasks are required to get started in Control Center:

- Change the default password for the Management Gateway Administrator.
- Specify the connection settings for the enterprise repository.
- If you are making a new installation of StreamServe Persuasion, you must create the database for the enterprise repository.
- For best practices, you should also rename the site node.

## On First Run

Figure 5-6:

On First Run



The first time you open Control Center, you are automatically connected management gateway on the local computer as the Management Gateway Administrator.

This user is created with the Control Center and Framework setup, or Framework setup on UNIX. This user is assigned the Administrator role.

The user name for the Management Gateway Administrator is Administrator

You must enter a new password for the Management Gateway Administrator the first time you open Control Center.

The user name and password are case-sensitive.

## Connecting to the Enterprise Repository

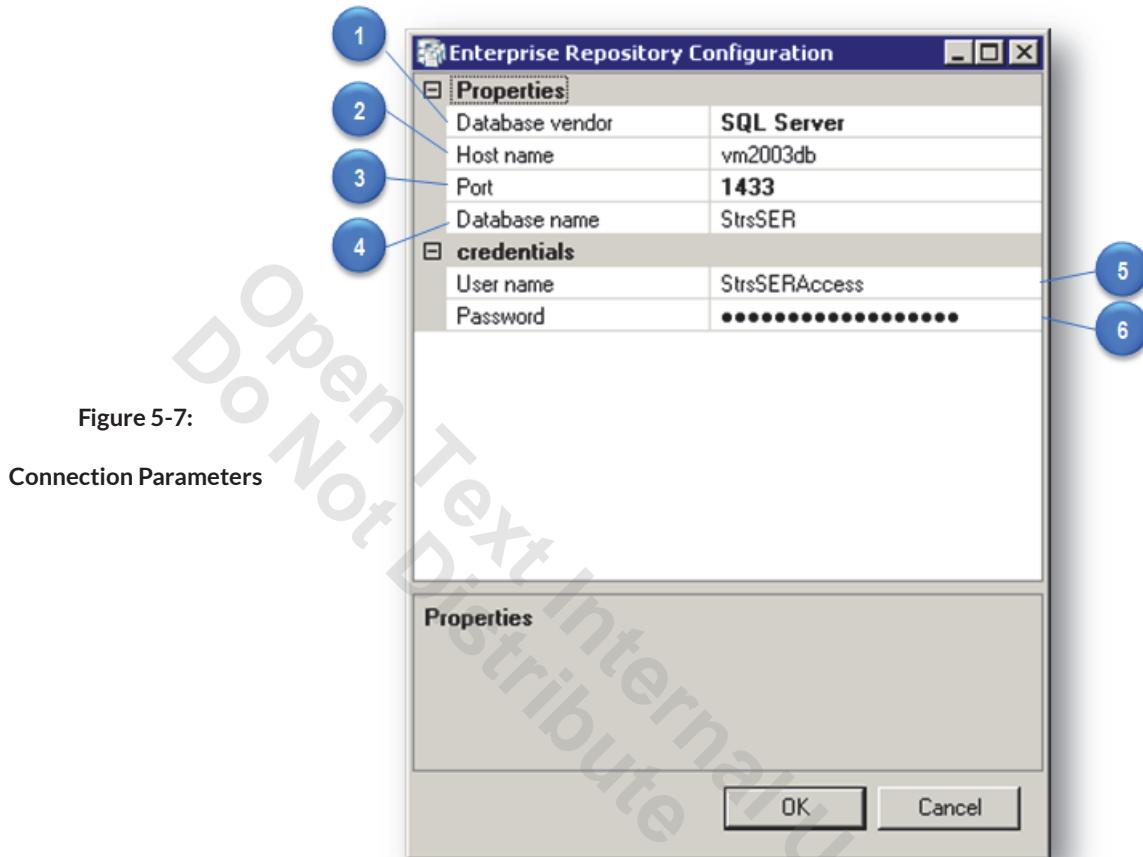


Figure 5-7:  
Connection Parameters

The Enterprise Repository Configuration dialog is displayed automatically. This dialog is used to configure the connection settings for the enterprise repository.

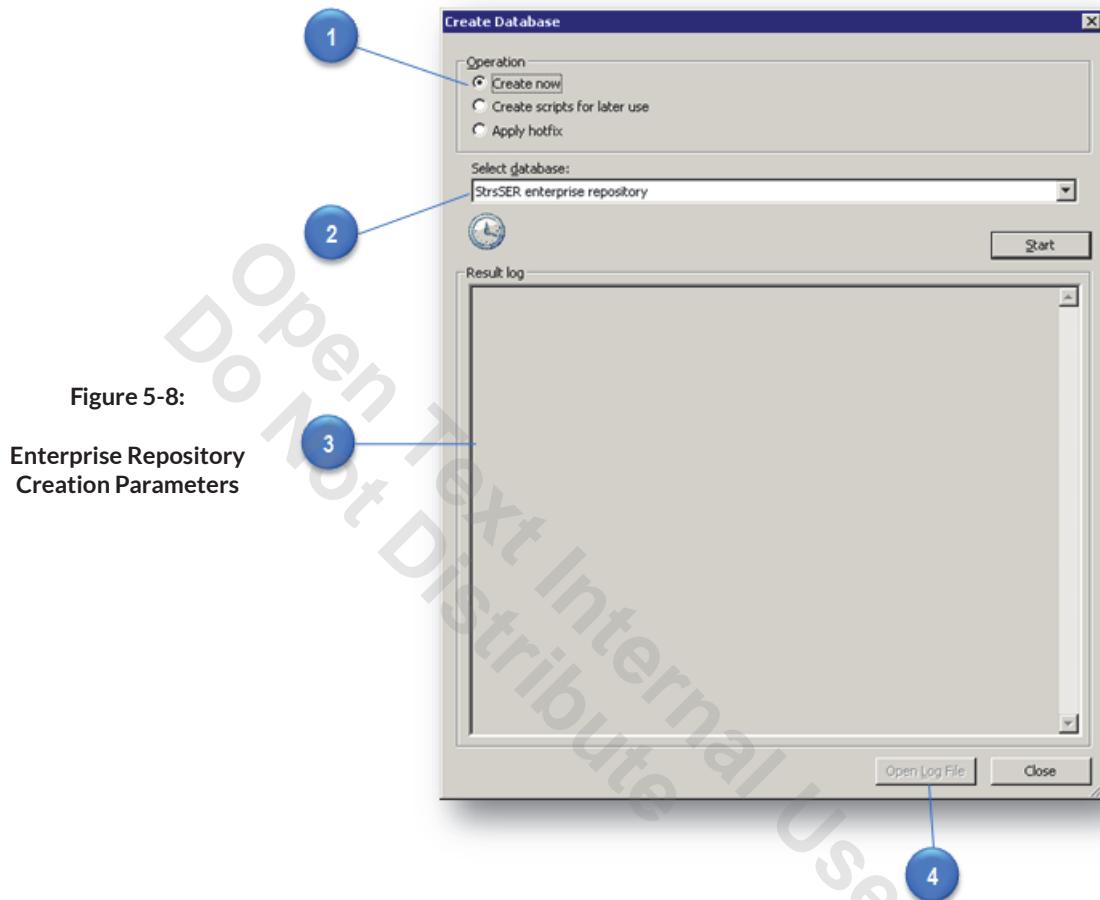
If there is an existing enterprise repository at your company or organization, you connect the site to that enterprise repository. If you are making a new installation of StreamServe Persuasion, you can create a new enterprise repository for the site.

**Settings – Enterprise Repository Configuration dialog**

Database Vendor	Note
Host name	The IP address or host name of the database server. If you use a named instance of SQL Server, you must specify the host name and instance name using the syntax <hostname>\<instancename>. For example: gbg5000\instance1
Port	The port used for communication with the database server. The default port for Microsoft SQL Server is 1433. The default port for Oracle Database is 1521.
Database name	A name for the enterprise repository. For Oracle, you must also specify a Service name.
User name	The user name to access the enterprise repository. The user is automatically created when the enterprise repository is created. For Oracle, the user name is used as schema owner. You cannot use the system administrator as user name (for example, sa for SQL Server).
Password	The password to access the enterprise repository.

The user name, password and database name should comply with the naming standards in your database server.

## Creating the Enterprise Repository



**Creating the database directly from Control Center** You can create the database for the StreamServe Enterprise Repository directly in Control Center (for Microsoft® SQL Server and Oracle® Database). The database administrator user name and password are required for this.

**Creating the database manually** You can also create the database for the enterprise repository manually. This may be required if the company security policy prevents Control Center from connecting to the database, or if you want to have full traceability of the repository creation. See the StreamServe online help for more information.

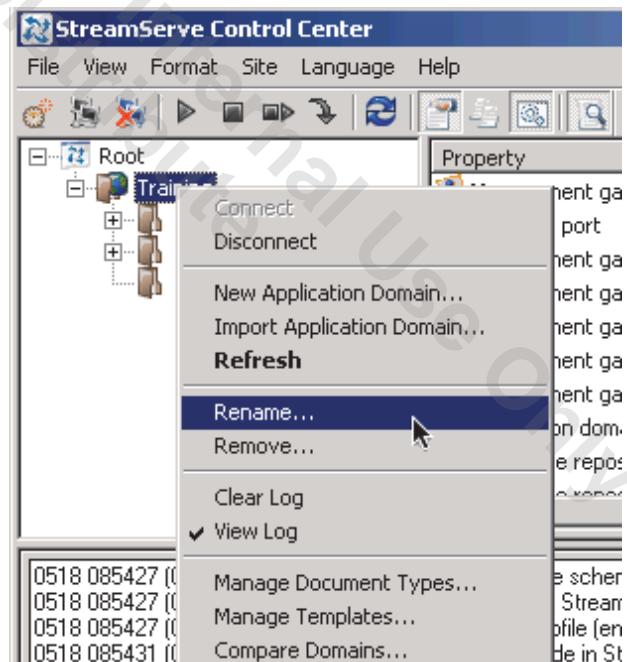
You use the Create Database dialog box to create the database for the enterprise repository.

### Settings – Create Database Dialog

Option	Description
Create now	Runs the scripts to create the repository selected in the drop down list.
Select database	Lists the StreamServe repositories at the site.
Results	Displays a summary of the log file. Select Open Log File to opens the full log in the default text editor.
Open Log File	Opens the full log file.
Create now	Runs the scripts to create the repository selected in the drop down list.

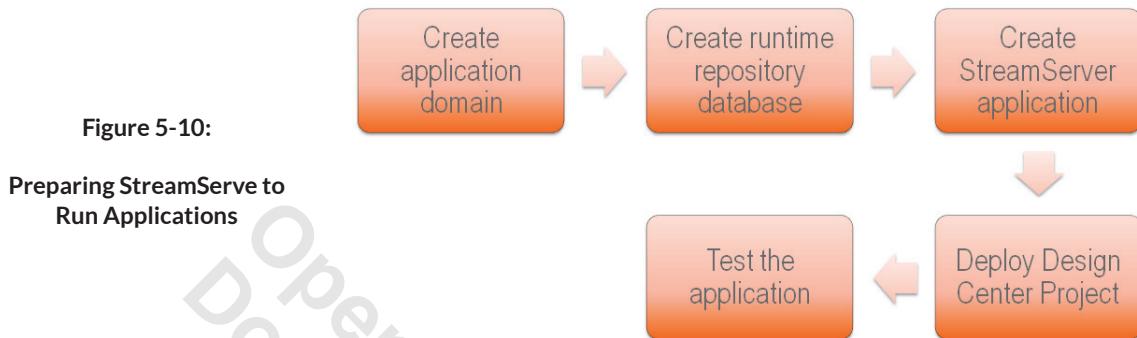
### Renaming the Site

Figure 5-9:  
Renaming the Site



The site node is called `New Site` by default. For best practice, you should rename the site node.

## Preparing to Run StreamServer Applications



The following steps are required to run Design Center Projects:

- Create and configure an application domain.
- Create a runtime repository for the application domain.
- Add StreamServer applications to the application domain.
- Deploy the Design Center Projects to the StreamServer applications.
- Use FastCopy to test your StreamServer applications (optional).

## Application Domain for StreamServer Applications

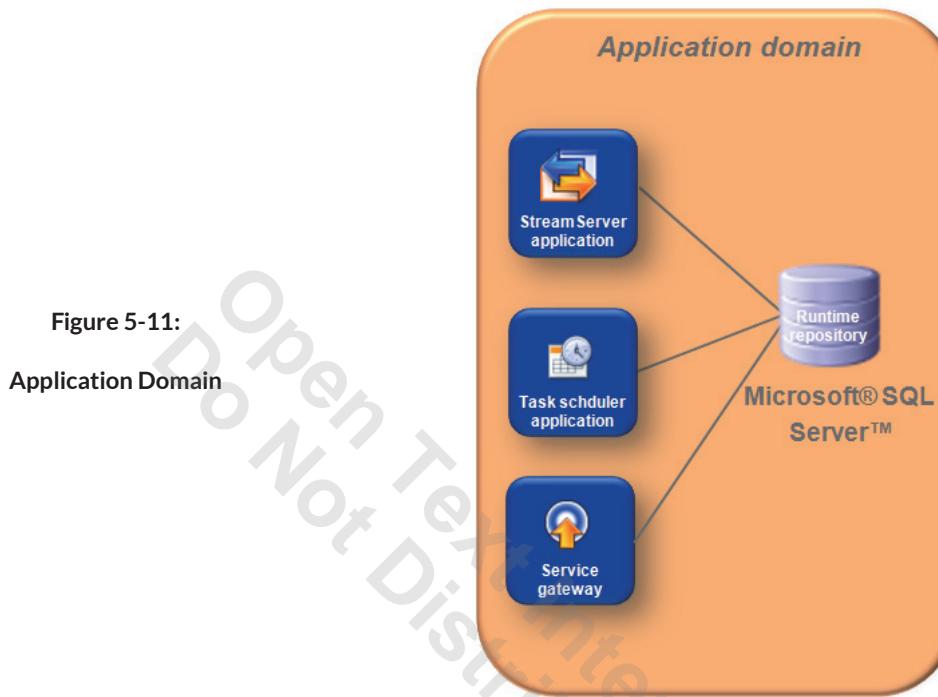


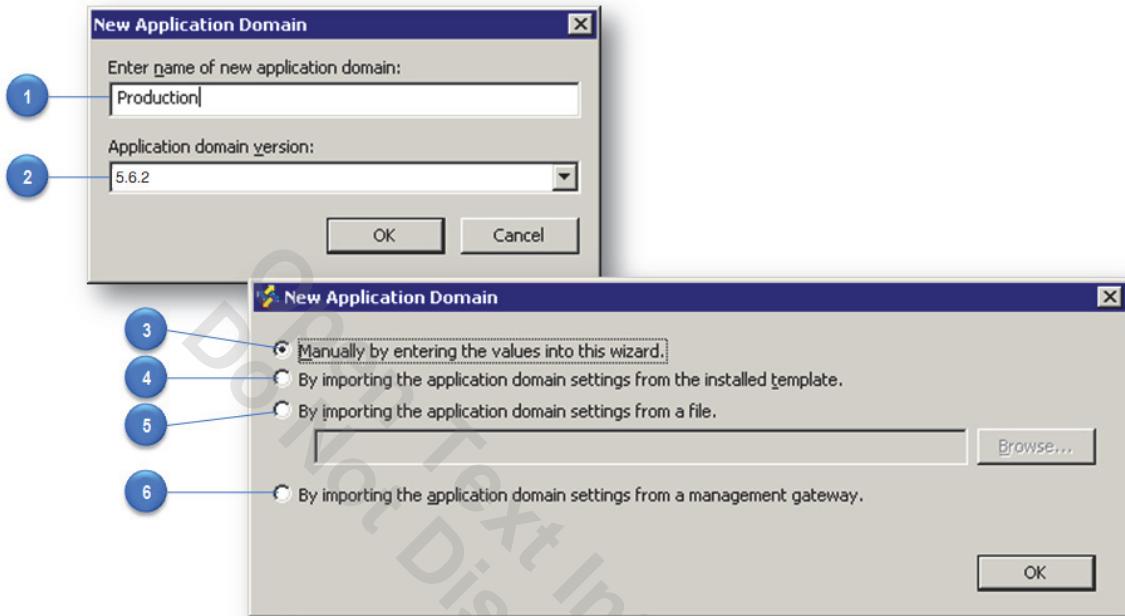
Figure 5-11:  
Application Domain

In this course, we will configure an application domain that can be used to run the following types of applications:

- StreamServer applications
- Service gateway
- Task scheduler

More information about how to configure the application domain for StreamStudio is covered in the StreamStudio Portal Administration e-learning course.

## Creating and Configuring an Application Domain



**Figure 5-12: Application Domain Configuration**

The New Application Domain dialog box is used to add a new application domain and specify how you want to configure the settings in the application domain.

**Settings – New Application Domain Dialog Box**

Option	Description
(1) Enter name of new application domain	The name of the application domain.
(2) Application domain version	The version of the application domain. For example: 5.6.2
(3) Manually by entering values into this wizard	Used to manually enter the settings for the application domain.
(4) Importing from a template	Imports an application domain configuration from the application domain template provided in the installation.  The template includes preconfigured connection settings for a runtime repository in SQL Server and to the OpenDS directory server. You can make changes to the imported configuration using the Application Domain Editor.

Option	Description
(5) Importing from a file	Imports an application domain configuration from a file. The passwords to access the repositories are not imported. You can make changes to the imported configuration using the Application Domain Editor.
(6) By importing the application domain settings from a management gateway	Imports an application domain from a management gateway via the Import from Gateway dialog box.  The user names and passwords to access the repositories are not imported. You can make changes to the imported configuration using the Application Domain Editor.

## Application Domain Editor

The Application Domain Editor is used to configure the following settings for the application domain:

- Application domain administrator
- Connection settings to the repositories
- Connection settings to the user directories
- Recovery actions
- Web Services
- Web Content repository
- Application domain contact

## Administrator Tab

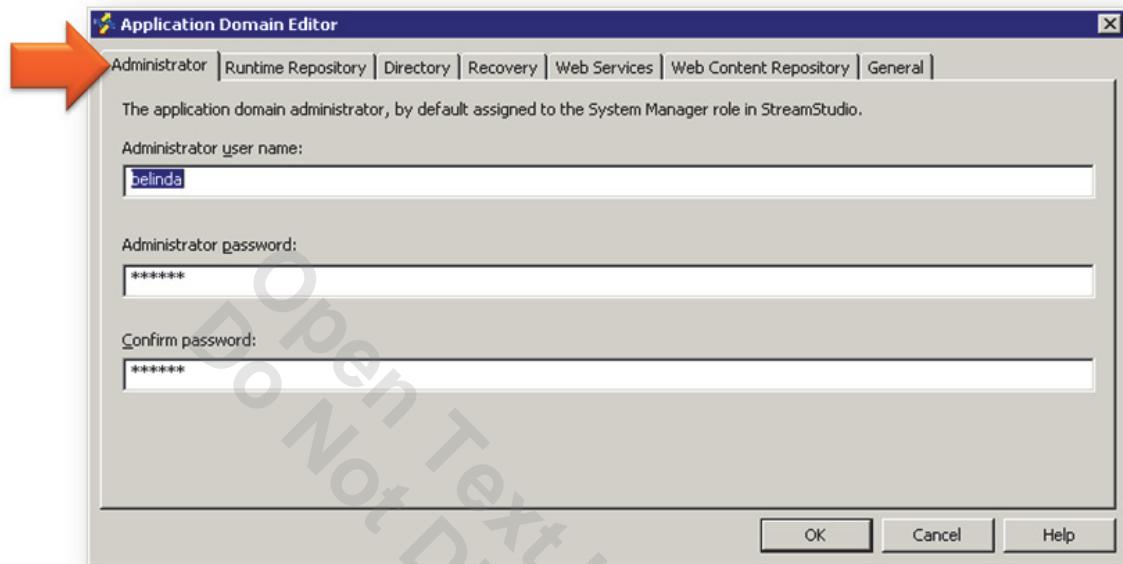
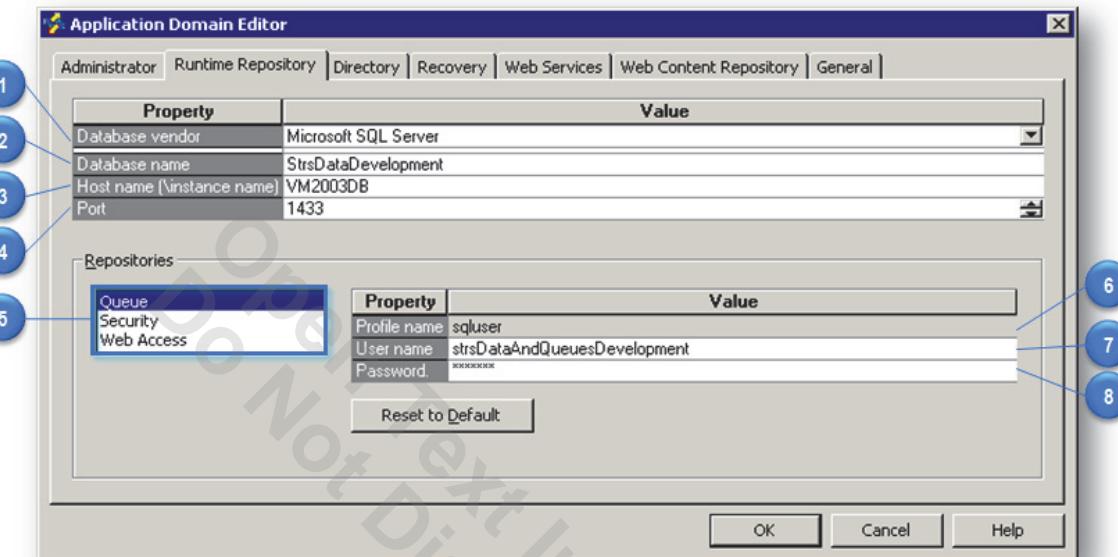


Figure 5-13: Administrator Tab

This tab is used for specifying the application domain administrator.

The application domain administrator is automatically assigned the System Manager role in StreamStudio. The administrator can be used to log on StreamStudio without a user directory.

## Runtime Repository Tab



**Figure 5-14: Runtime Repository Tab**

The Runtime Repository tab is used for specifying the connection settings for the Queue, Security, and Web Access repositories.

### Settings – Runtime Repository Tab

Field	Description
(1) Database vendor	The database vendor used for the runtime repository.
(2) Database name	A name for the runtime repository. The name should comply with the naming standards in your database server.
(3) Host name (instance name)	The IP address or host name of the database server. For named instances of SQL Server, specify both the hostname and instance name of the database. For example: gbg5000\instance1
(4) Port	The port used for communication with the database server.

Field	Description
(5) Repositories - Connection profiles	The connection profiles for the Queue, Security, and Web Access repositories.
(6,7,8) Property area	These users are created in the runtime repository when you create the repository.

## Directory Tab

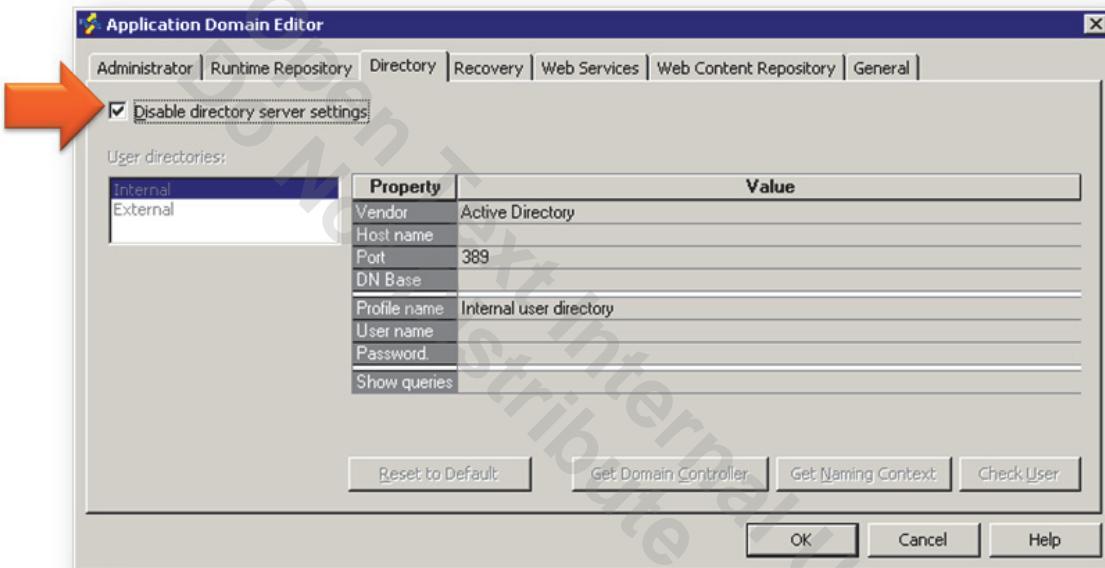


Figure 5-15: Directory Tab

The Directory tab is used for specifying the connection settings to access the internal and external user directories.

Select this option if you run without user directories in your application domain.

## Recovery Tab

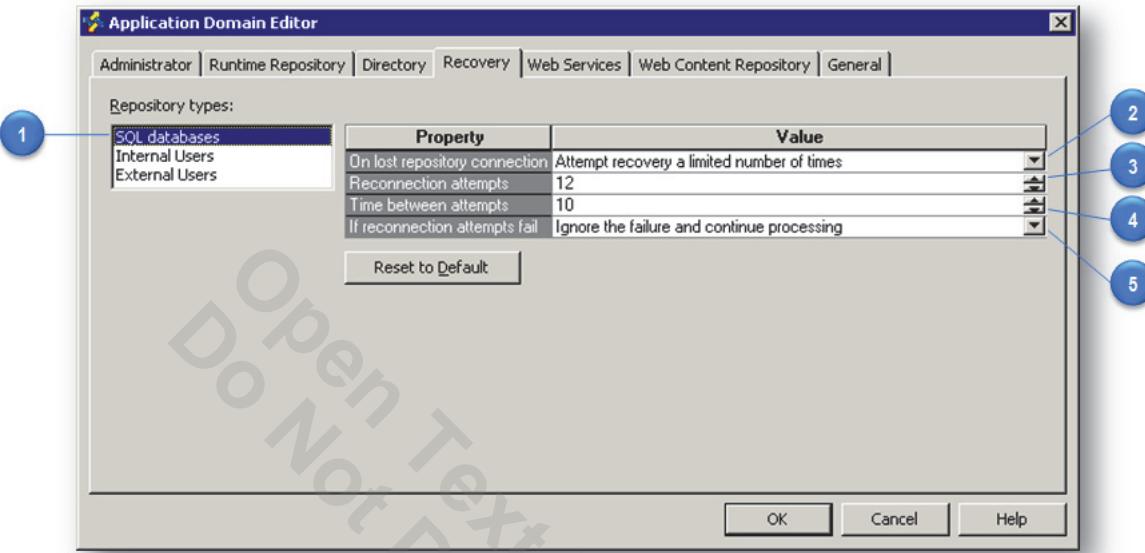


Figure 5-16: Recovery Tab

The Recovery tab is used for specifying the recovery options for the runtime repository and the user directories.

### Settings – Recovery Tab

Field	Description
(1) SQL databases	Specifies the recovery options for the runtime repository.
(2) On lost repository connection	Specifies how StreamServer applications should reconnect to the repository in case of a lost connection.
(3) Reconnection attempts	The number of times StreamServer applications attempt to reconnect the repository.
(4) Time between attempts	The time period between the reconnection attempts
(5) If reconnection attempts fail	<p><b>Ignore the failure and continue processing</b> The StreamServer application continues to process the current job.</p> <p><b>Stop processing on the affected StreamServer</b> The current job is aborted and the StreamServer application is stopped.</p>

## Web Services Tab

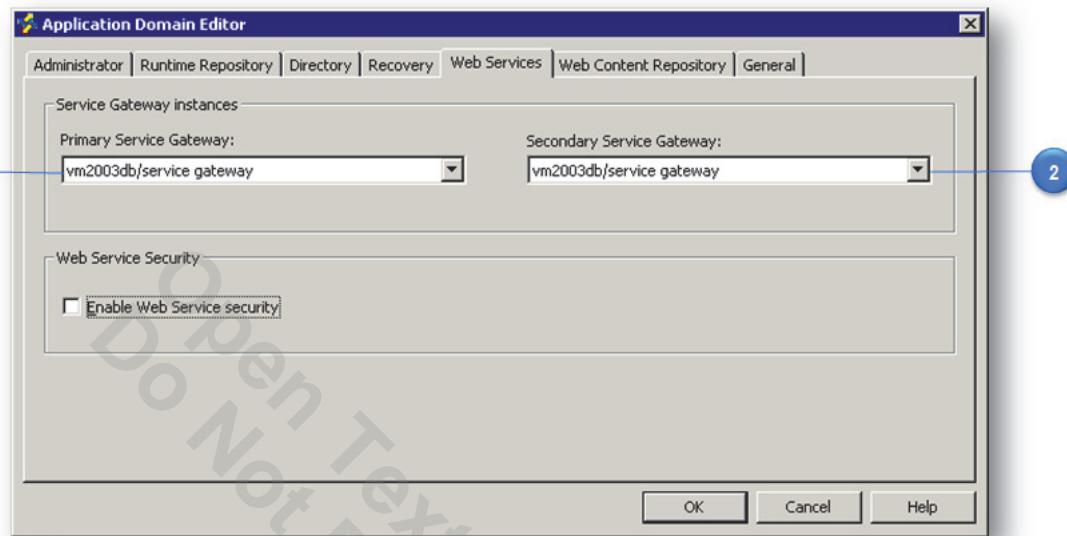


Figure 5-17: Web Services Tab

The Web Services tab is used for specifying the communication settings for the service gateways, and for enabling web service security.

This is required to run StreamStudio web applications. A service gateway is also required if you want to validate job sender or receiver against a user directory.

### Settings – Web Services

Field	Description
(1) Primary Service Gateway	The primary service gateway that is used to host web services.
(2) Secondary Service Gateway	The secondary service gateway, which is used for failover and load balancing reasons.

## Web Content Repository Tab

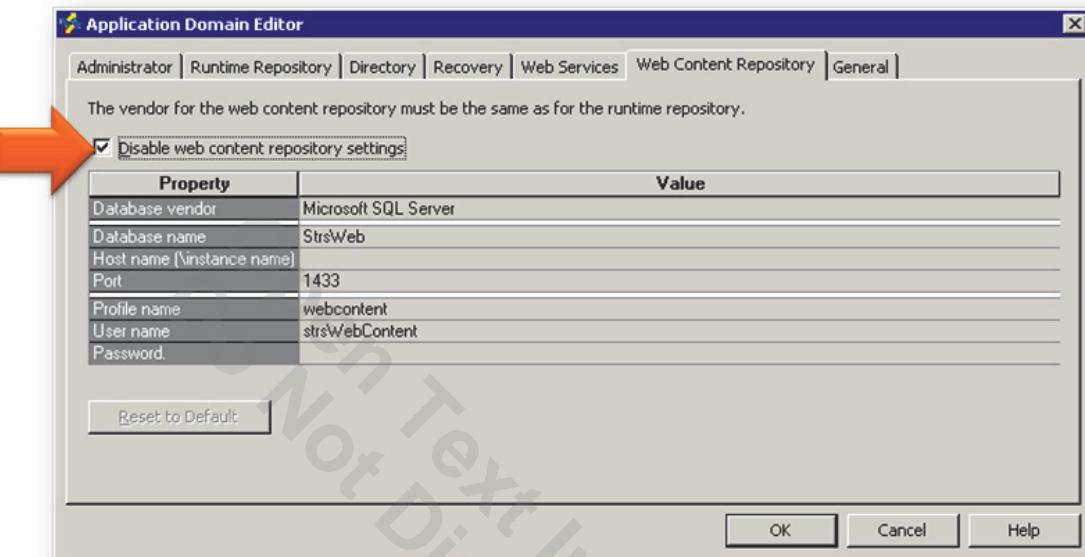


Figure 5-18: Web Content Repository Tab

The Web Content Repository tab is used for specifying the connection settings to access the Web Content Repository.

The web content repository is only required for StreamStudio Composition Center.

Select this option if you are not running StreamStudio Composition Center in the application domain.

## General Tab

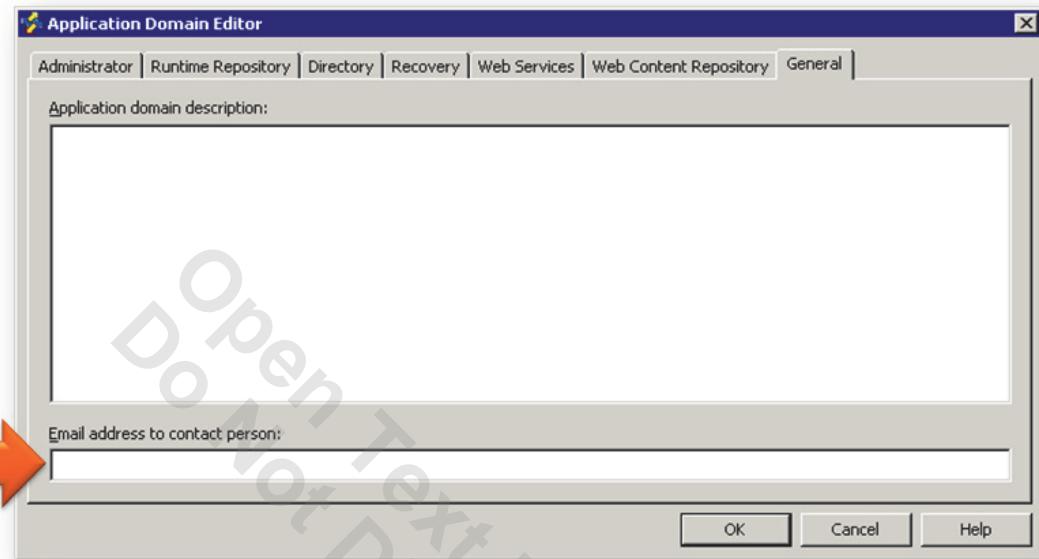


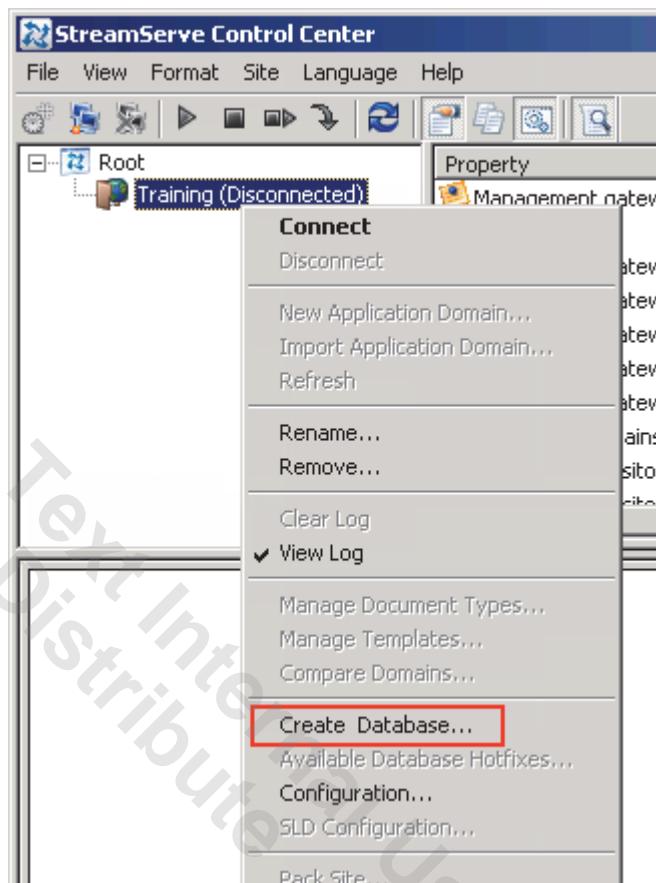
Figure 5-19: General Tab

This tab is used to add a description and an email address for the person responsible for the application domain.

If the surveillance functionality is used, surveillance emails are sent to this address.

## Creating a Runtime Repository

Figure 5-20:  
Creating a Runtime  
Repository



You must create one runtime repository for each application domain.

There are two methods to create a runtime repository:

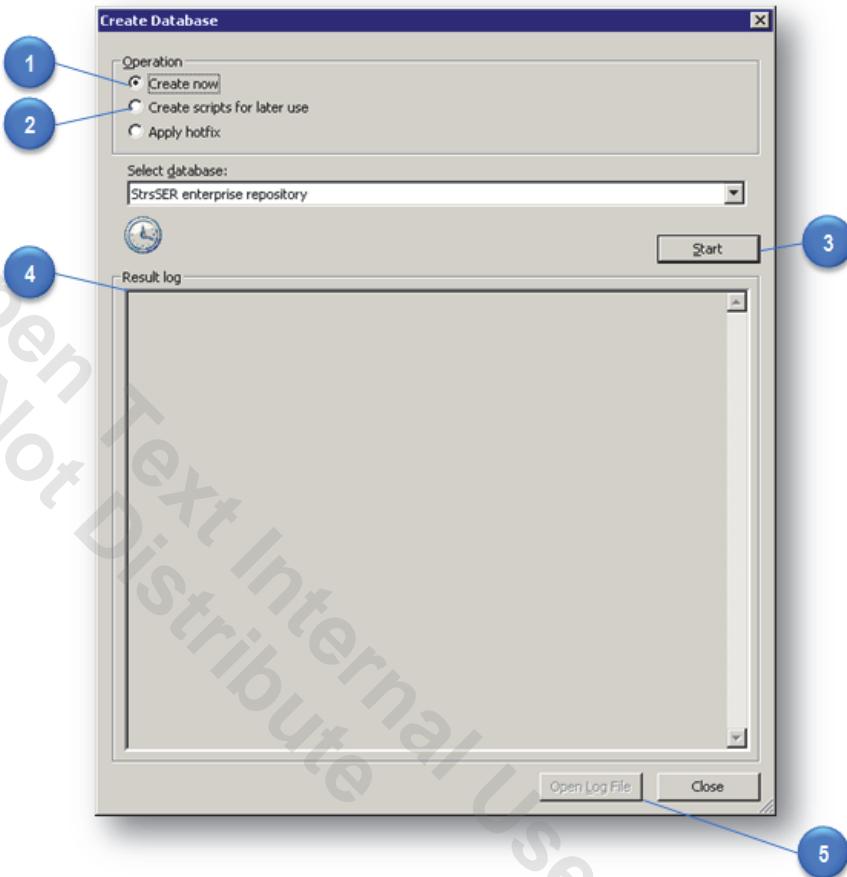
- Create a runtime repository from Control Center.
- Create a runtime repository manually.

A runtime repository that you create from Control Center is sufficient for development and testing purposes.

Before using the repository in a production environment, you must edit the database files to fit the actual conditions (for example, configure file groups, size and file growth parameters, etc).

This course covers how to create a runtime repository from Control Center. For information about how to create a runtime repository manually, see *Creating a runtime repository manually* in the user documentation.

**Create Database Dialog Box**



**Figure 5-21:**

**Create Database Dialog Box**

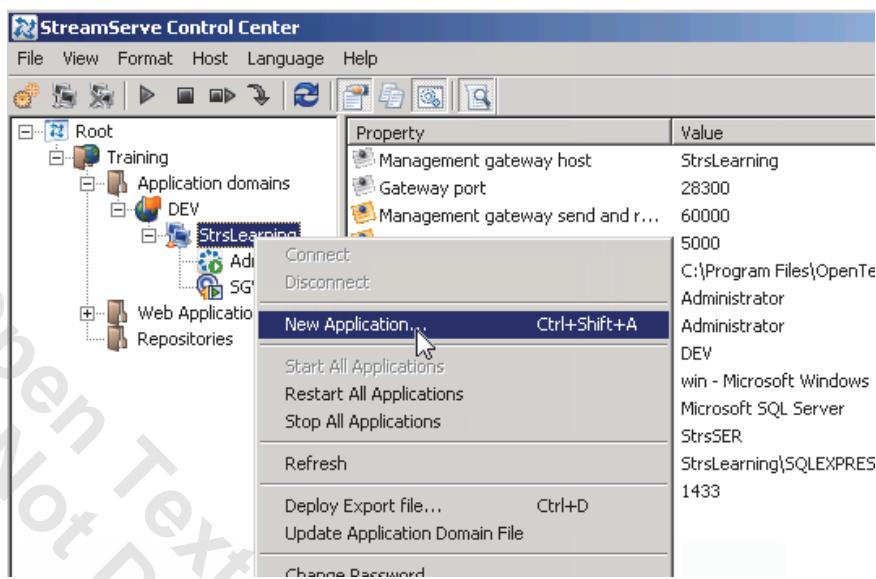
The Create Database dialog box is used to create the repositories for the application domain.

**Settings - Create Database Dialog Box**

Field	Description
(1) Create database	Creates the runtime repository in the database server. The database administrator user name and password are required for this.
(2) Create scripts for later use	Saves the SQL scripts used to create the runtime repository.
(3) Start	Starts the selected operation.
(4) Result log	Displays a summary of the result log.
(5) Open Log File	Opens the full log in a text editor.

## Adding a StreamServer Application

Figure 5-22:  
Adding a StreamServer  
Application



You add StreamServer applications to the application domain.

You need one StreamServer application for each Design Center Project.

You can add applications on your local host, or on remote hosts that are part of the site. The remote host must be configured to use the StreamServe Enterprise Repository that the site is connected to.

## New Application Dialog Box

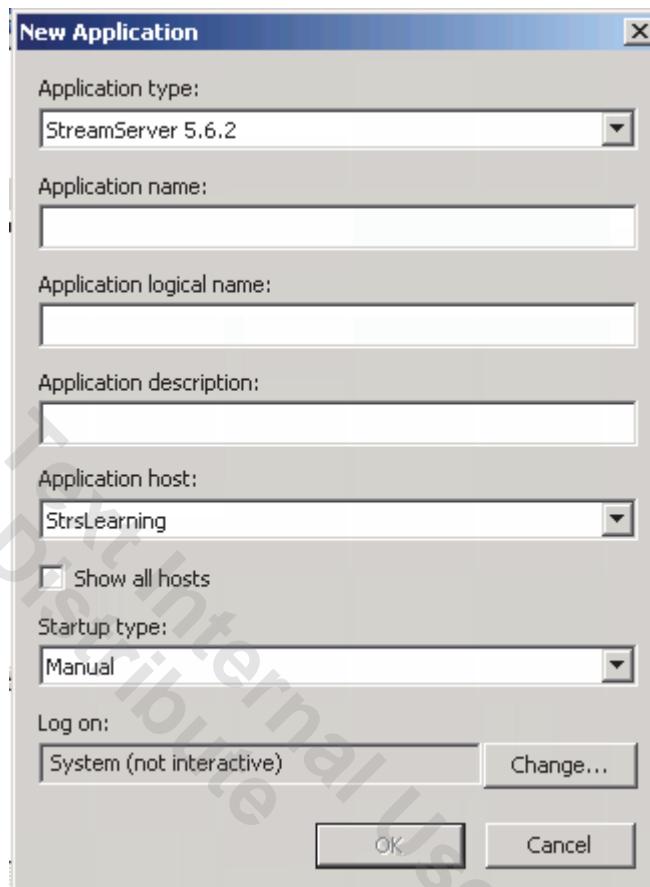


Figure 5-23:  
New Application Dialog  
Box

The New Application dialog box is used to add StreamServer applications.

### Settings - New Application Dialog Box

Field	Description
Application type	The type and version of the application (Archiver, Service Gateway, StreamServer, etc).
Application name	A logical name for the application, that can be used for development purposes.
Application logical name	A description of the application.
Application description	A description of the application.
Application host	The computer used to run the application. Select <b>Show all hosts</b> to display all the hosts that are part of the site.

Field	Description
Startup type	Specifies how the application is started. <b>Automatic</b> - Starts the application automatically when the system (host for the corresponding application) starts. <b>Manual</b> - Starts the application manually. <b>Disabled</b> - Disables the application.
Log on	The account used to run the application.

## Deploying a Design Center Project

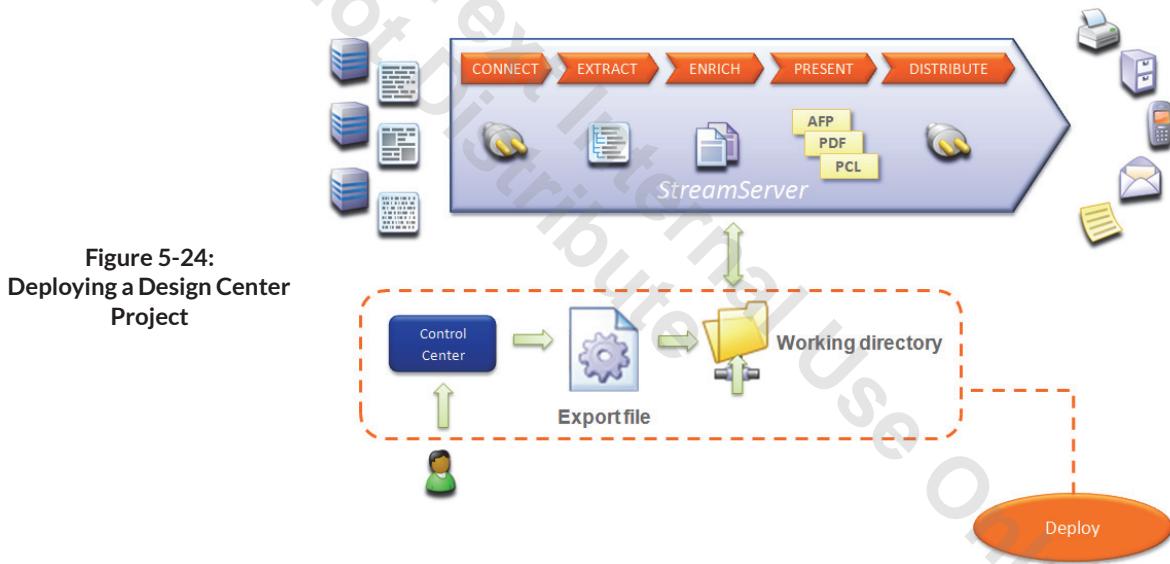


Figure 5-24:  
Deploying a Design Center  
Project

To run a StreamServer application, you must deploy an export file for a Project to an application.

Each time you export a new version of the Project in Design Center, you must redeploy the export file in Control Center.

When you deploy the export file, the contents of the file are extracted to the working directory for the application. More information about working directories is covered in the Disk Structure module.

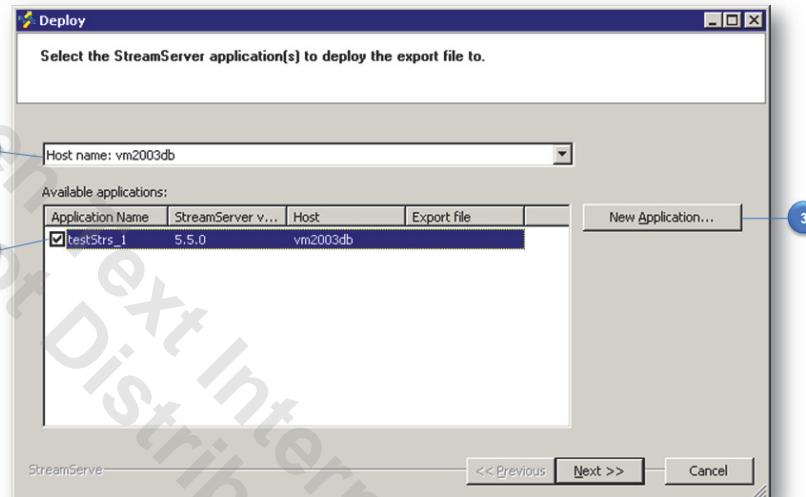
For information about how document types are deployed to the StreamServer repositories, see *Document type deployment overview* in the user documentation.

There are two methods to deploy export files:

- From the file system
- From a version control system

**Deploy wizard –  
Select the  
Applications**

**Figure 5-25:  
Deploy wizard – Select the  
Applications**



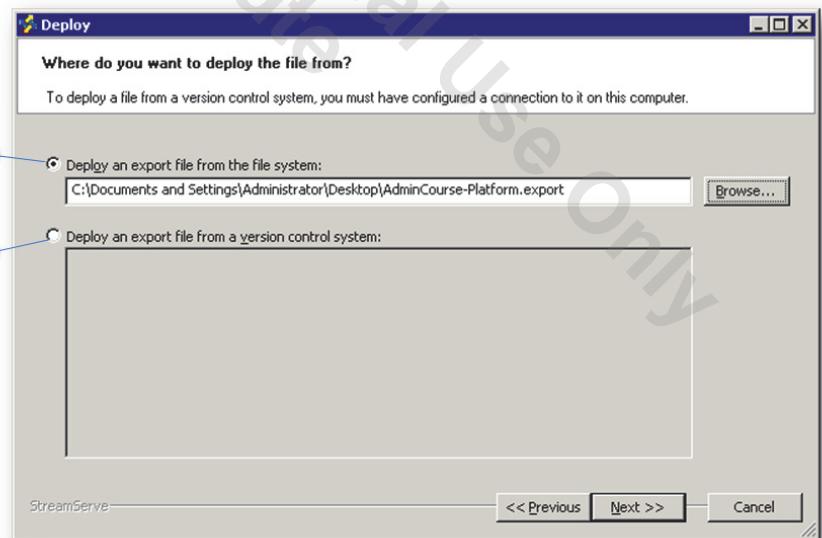
This is used to select the application(s) that you want to deploy the export file to.

You can deploy the export file to an existing StreamServer application or create a new application. You can also deploy a Project to several StreamServer applications simultaneously.

### Settings - Deploy Wizard - Select Application

Field	Description
(1) Host drop-down list	Select the host that will be used to run the application.
(2) Available applications	<p>Lists the existing applications at the site or for the host.</p> <p><b>Application Name</b> - The name of the application.  <b>StreamServer version</b> - The version of the application.</p> <p><b>Host</b> - The host used to run the application.</p> <p><b>Export file</b> - If there is already an export file deployed to the application, the name of the export file.</p>
(3) New application	Opens the New Application dialog box, which is used to create a new application used to run the Project.

### Deploy wizard – Select File



**Figure 5-26:**  
Deploy wizard – Select File

This step is used to select the export file that you want to deploy.

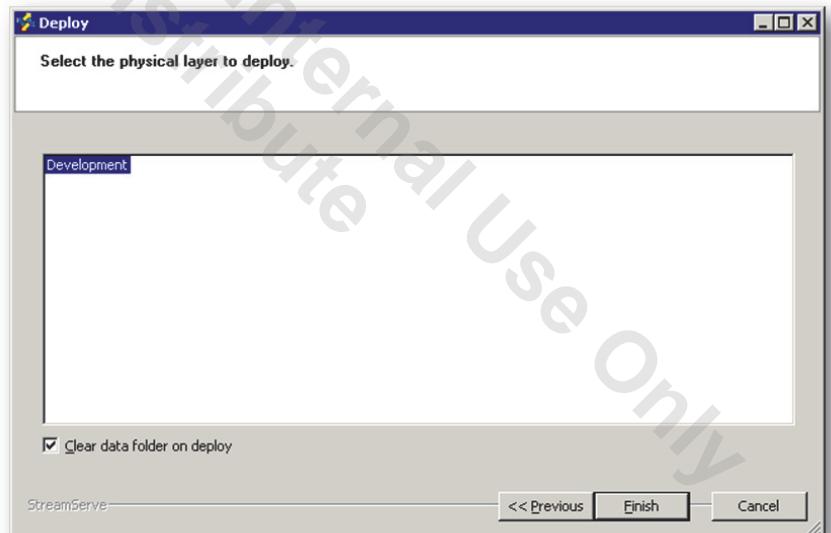
You can connect Control Center to a version control system, which enables you to deploy export files that have been checked-in to the version control system in Design Center.

Connections configured to a version control system in a Design Center instance running on the same computer are automatically available in Control Center.

**Settings - Deploy Wizard -**  
**Select Export File**

Field	Description
(1) Deploy an export from the file system	Used to select an export file from the file system. Click <b>Browse</b> to select the file.
(2) Deploy an export file from a version control system	Used to select an export file from a version control system. You must select the connection profile to connect to the version control system and enter the user name and password.

**Deploy wizard -**  
**Select Physical Layer**



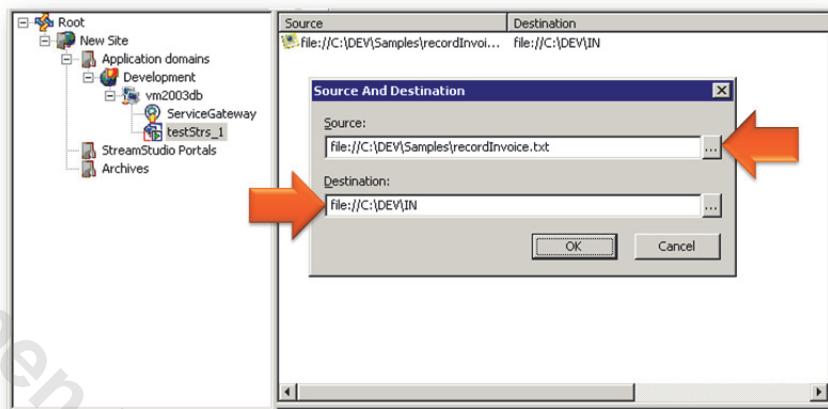
**Figure 5-27:**  
**Deploy wizard – Select Physical Layer**

This step is used to select the layer you want to deploy.

The physical layer is based on the Platform configuration in Design Center. You typically specify one physical layer for Development, one for Testing, and one for Production.

## Testing a StreamServer Application

Figure 5-28:  
Testing the Application



There are different methods to test StreamServer applications. Some examples include:

- Using FastCopy in Control Center.
- Copying files manually to the destination directory.
- Sending input data from the backend systems.

You can use FastCopy to test StreamServer applications with one or more Directory input connectors.

FastCopy copies files from a source directory to a destination directory. The source and destination directories can be located on the local computer or a remote computer.

## Exercise - Configuration After Initial Installation



### **Set the Password for the Management Gateway Administrator**

1. Navigate to **Start > OpenText > Control > Control Center**.
2. In the Welcome to Control Center select the **OK** button.

Figure 5-29:  
Control Center - Welcome  
Page



3. In the Change Password window enter :
  - New password: **streamserve**
  - Confirm new password: **streamserve**

The Configure Enterprise Repository dialog box is displayed.



### Set the Enterprise Repository Connection Parameters

1. In the Enterprise Repository Configuration window enter the following value leaving the default values for the other fields and click the **OK** button:
  - Host name: **StrsLearning\SQLEXPRESS**

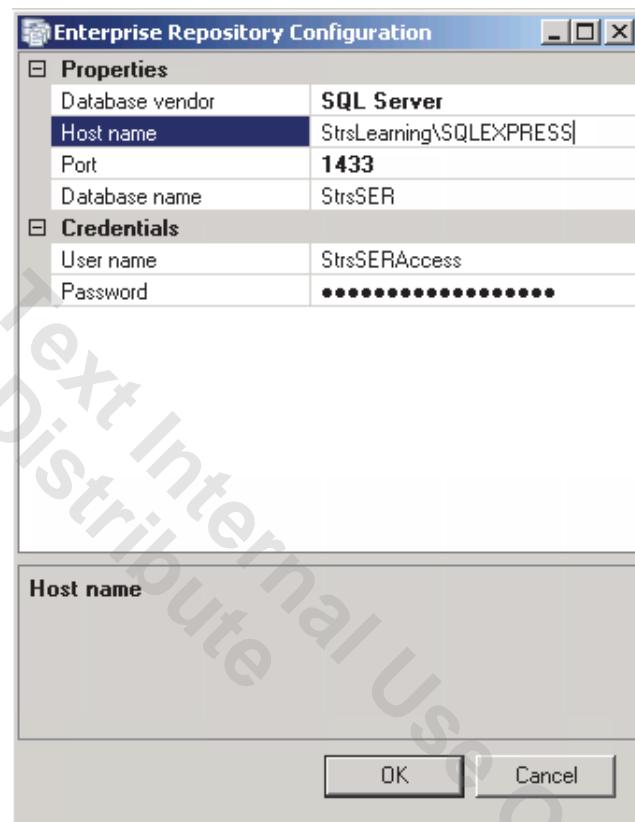


Figure 5-30:

Enterprise Repository Configuration

Control Center tries to connect to the Enterprise Respository database, but it still doesn't exit. You will create it shortly.

2. In the Control Center warning window click the **OK** button.



### Rename the Site

1. In Control Center, right-click **New Site (Disconnected)** and select **Rename** in the pop-up menu.

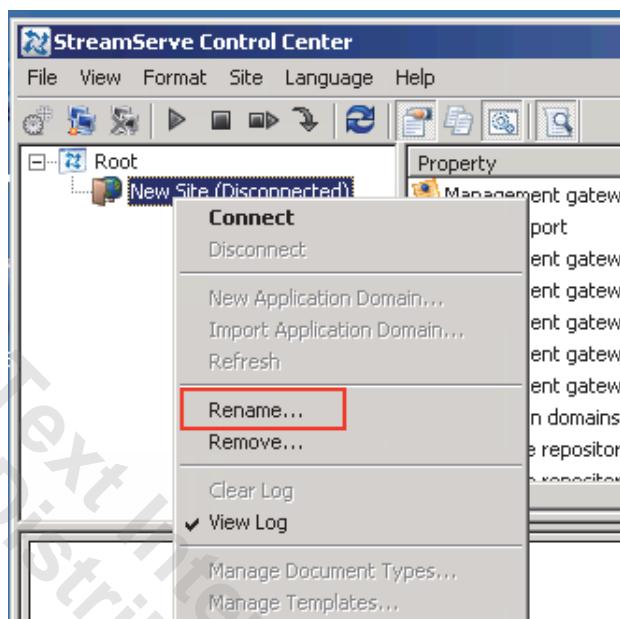


Figure 5-31:  
Renaming the New Site

2. Rename the New Site to **Training** and click the **OK** button.

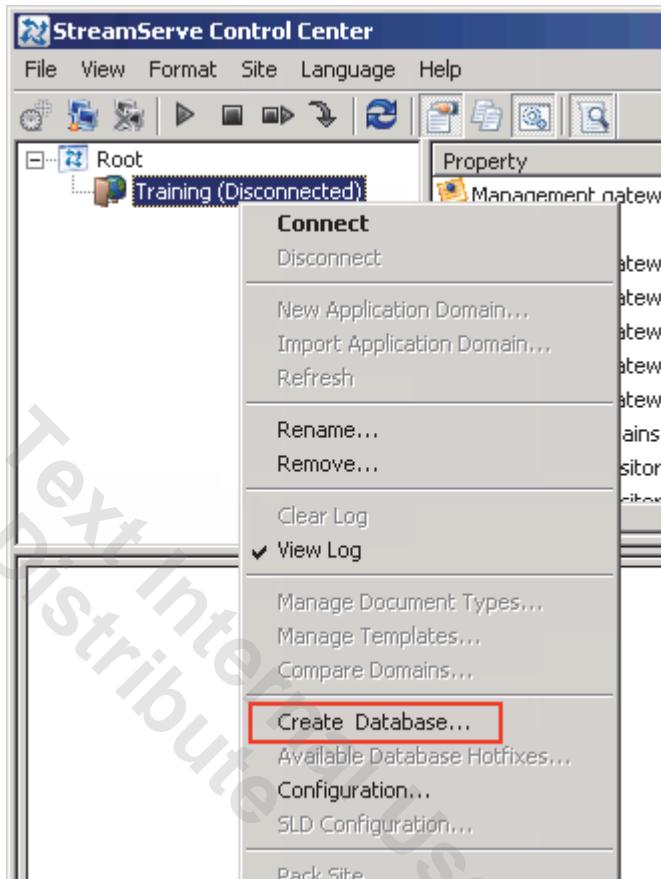


### Create the Enterprise Repository

1. In the Enterprise Repository Configuration window enter the following value leaving the default values for the other fields and click the **OK** button.

2. Right-click **Training (Disconnected)** and select **Create Database** in the pop-up menu.

Figure 5-32:  
Creating the Database



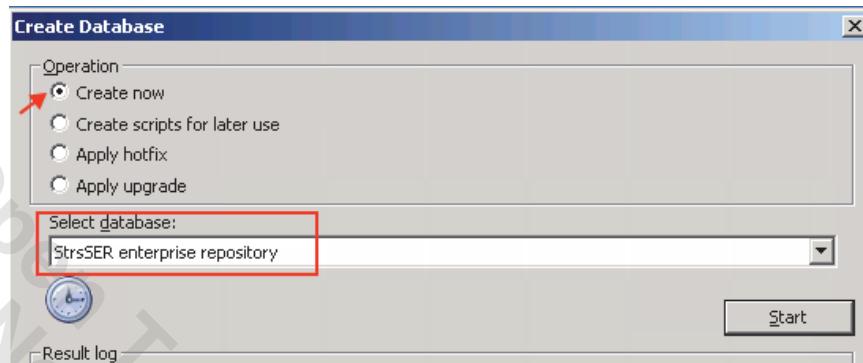
3. In the Connect window enter the following values and click the **OK** button.
  - User name: **Administrator**
  - Password: **streamserve**

4. In the Create Database window enter the following values and click the **Start** button:

- Operation: **Create now**
- Select database: **StrsSER enterprise repository**

Figure 5-33:

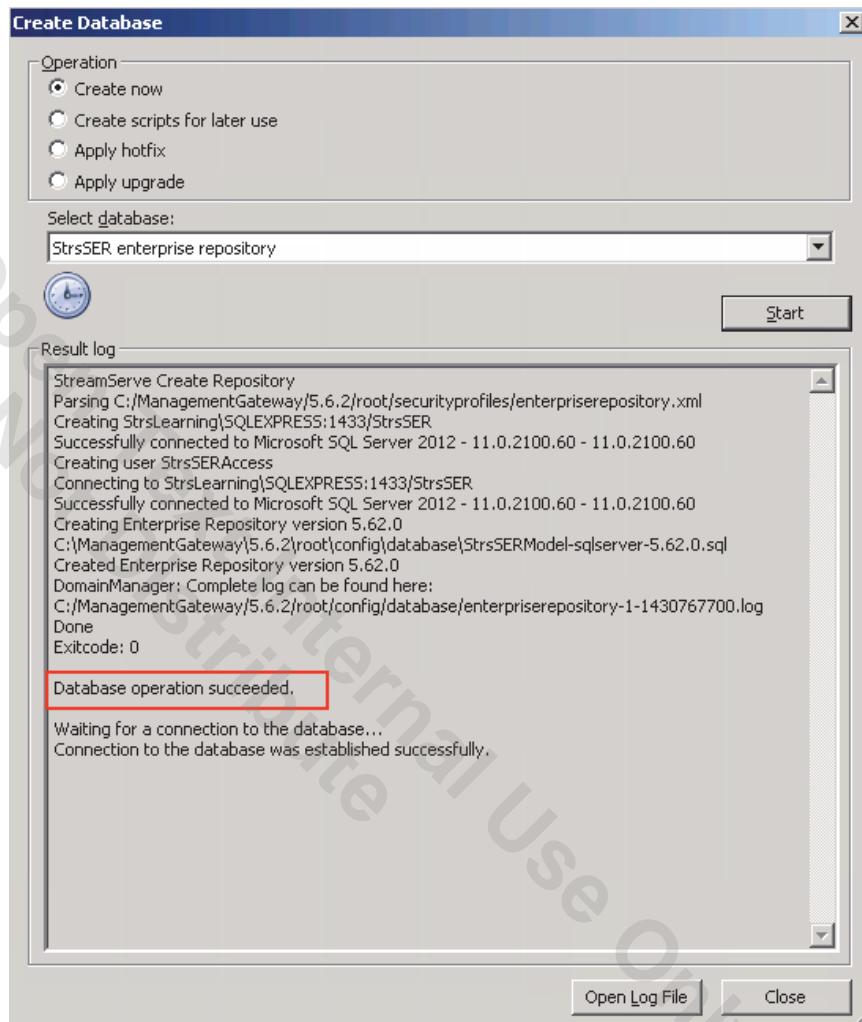
Database Parameters



5. In the Connect window enter the following values and click the **OK** button.

- User name: **sa**
- Password: **streamserve**

After a few seconds the database creation completes, make sure there are no errors.

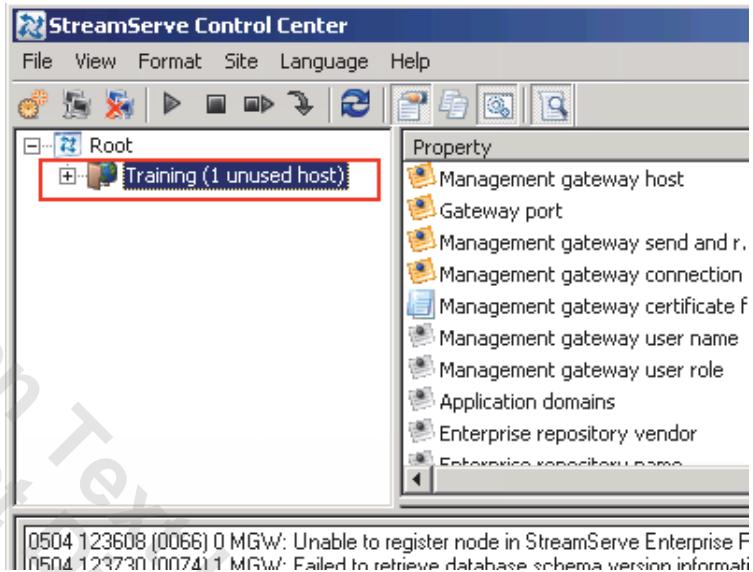


**Figure 5-34:**  
Database Creation Log

6. Click the **Close** button.

You are taken back to Control Center and a connection is established with the Training site:

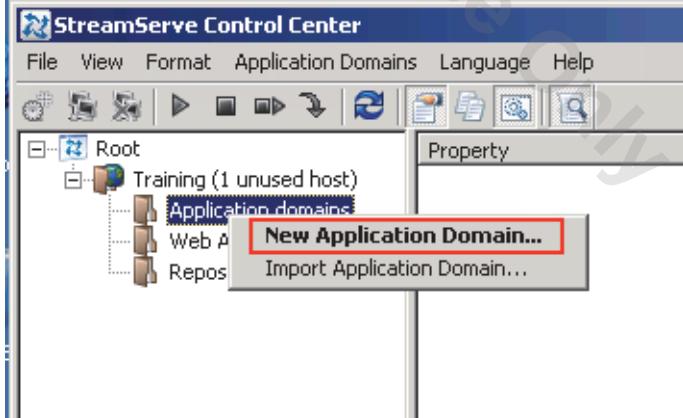
Figure 5-35:  
Training Site Connected



#### Create the Application Domain

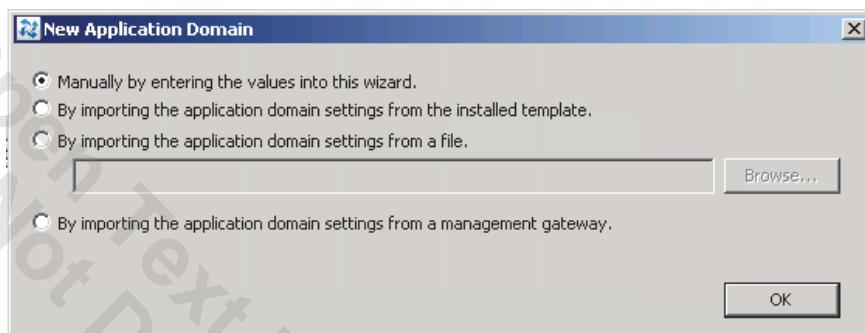
1. In Control Center expand the **Training (1 unused host)** node.
2. Right-click **Application domains** and select **New Application Domain** in the pop-up menu.

Figure 5-36:  
New Application Domain



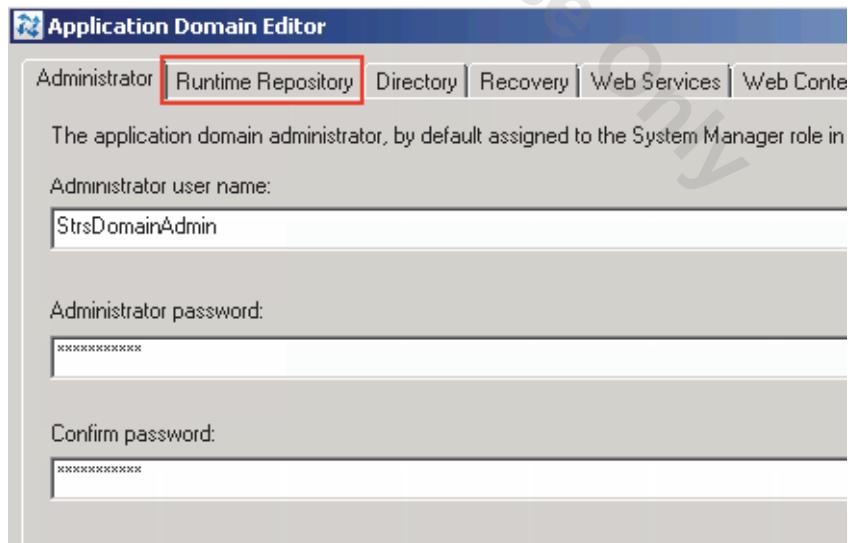
3. In the New Application Domain window enter the following values and click the **OK** button:
  - Application domain: **Development**
  - Application domain version: **5.6.2**
  - Physical layer: **Development**
4. In the New Application Domain window select **Manually by entering the values into this wizard** and click the **OK** button.

Figure 5-37:  
New Application Domain -  
Manual Values



5. In the Application Domain Editor enter the following values and then click the **Runtime Repository** tab:
  - Administrator user name: **StrsDomainAdmin**
  - Administrator password: **streamserve**
  - Confirm password: **streamserve**

Figure 5-38:  
New Domain Parameters



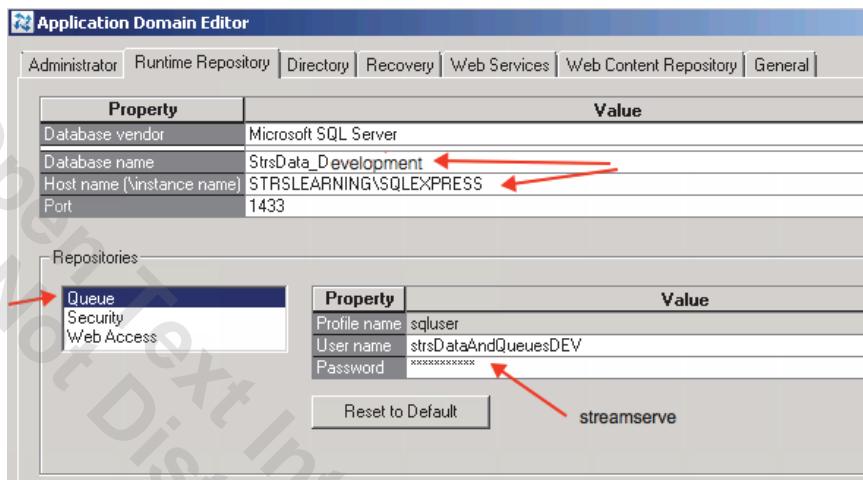
6. In the Runtime Repository tab enter the following values:

- Database name: **StrsData\_Development**
- Host name(\instance name): **STRSLEARNING\SQLEXPRESS**

and making sure that **Queue** is selected in the Repositories panel, enter **streamserve** in the password:

Figure 5-39:

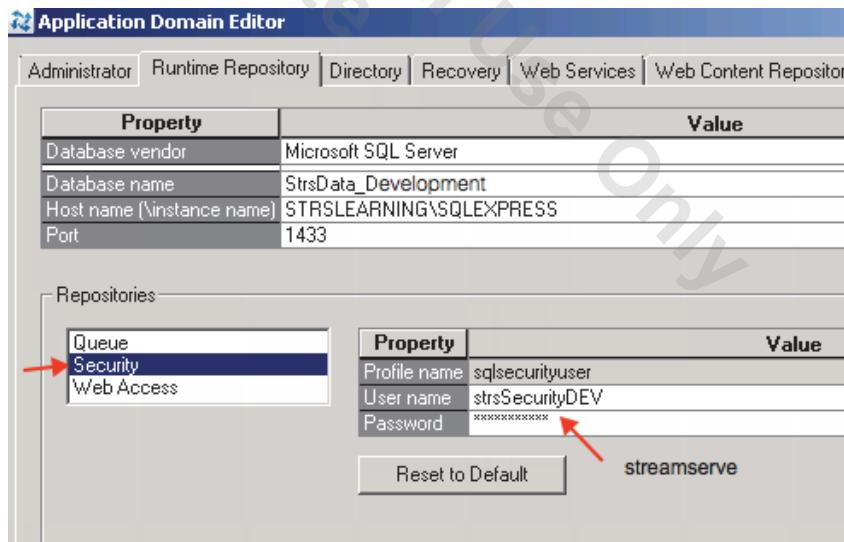
New Domain Runtime Repository Parameters



7. In the Repositories panel select **Security** and enter **streamserve** in the password:

Figure 5-40:

Security Password



8. In the Repositories panel select **Web Access** and enter **streamserve** in the password:

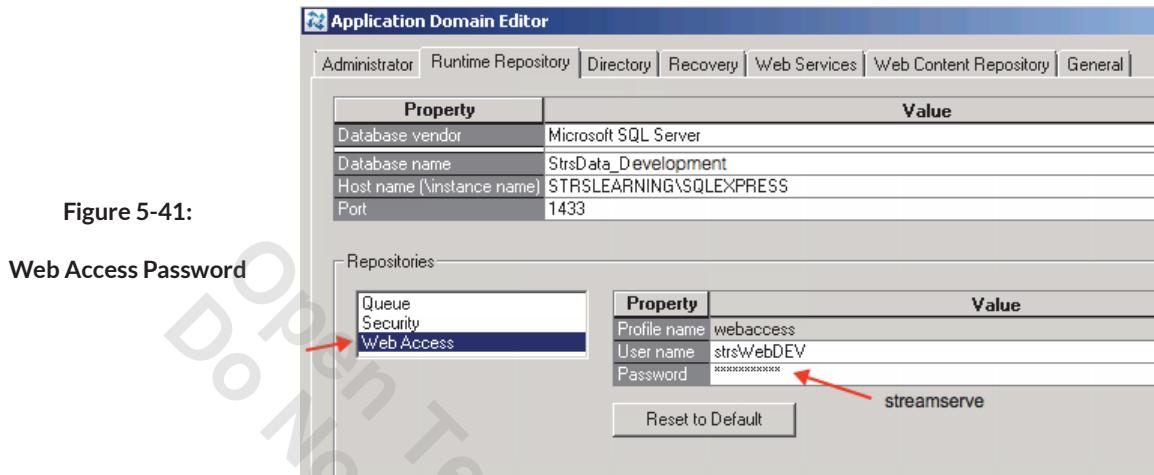


Figure 5-41:

9. Click the OK button.
10. Expand the application domains node and notice that the newly created **Development** application domain is displayed:

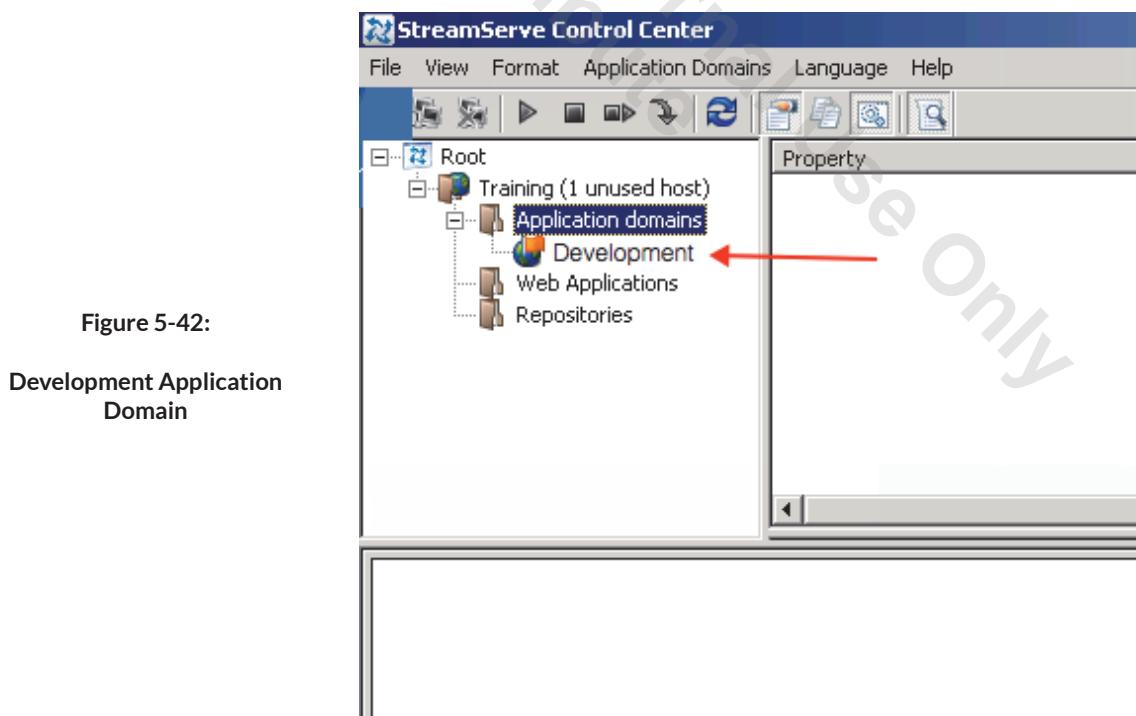


Figure 5-42:

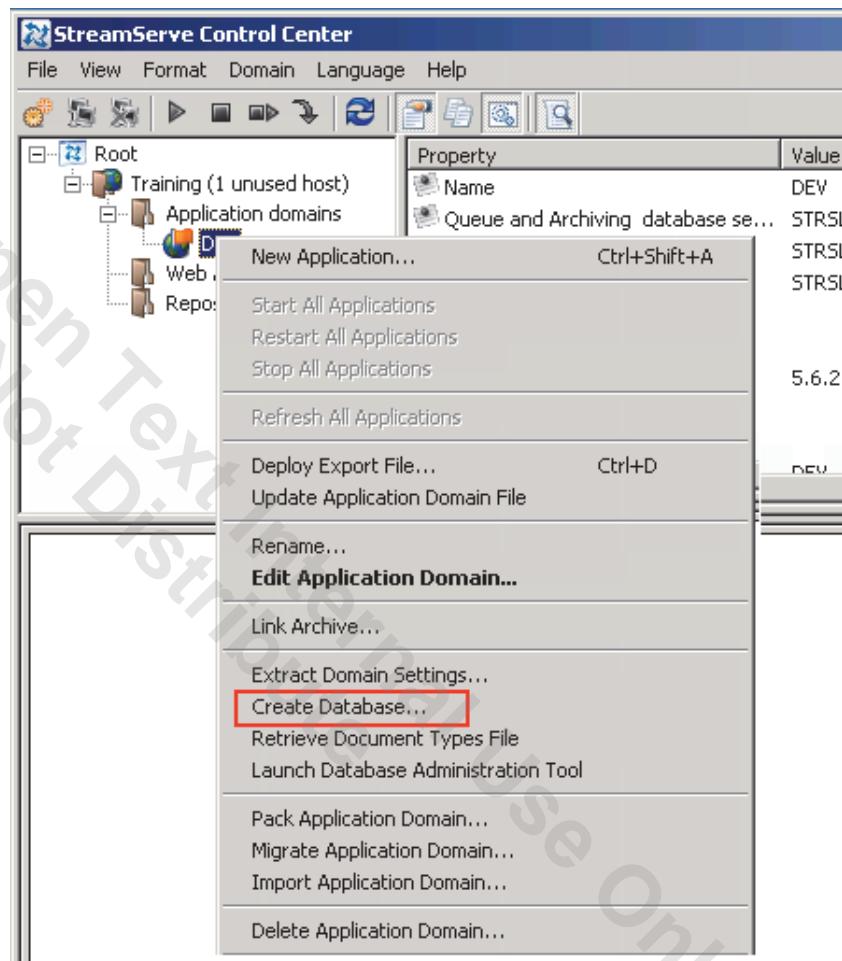
Development Application Domain



### Create the Runtime Repository

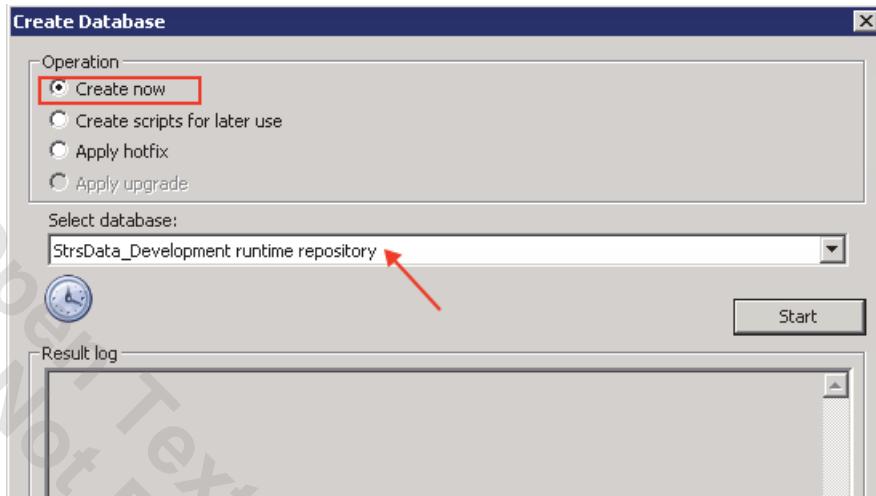
1. Right-click the **Development** application domain and select **Create Database** in the pop-up menu:

Figure 5-43:  
Development Database  
Creation



2. In the Create Database window making sure that **StrsData\_Development** is selected in the database drop-down, select **Create** now and click the **Start** button:

Figure 5-44:  
Creating the Development  
Database



3. In the Connect window enter the following values and click the **OK** button.
  - User name: **sa**
  - Password: **streamserve**
4. Make sure no error is reported in the log and click the **Close** button.

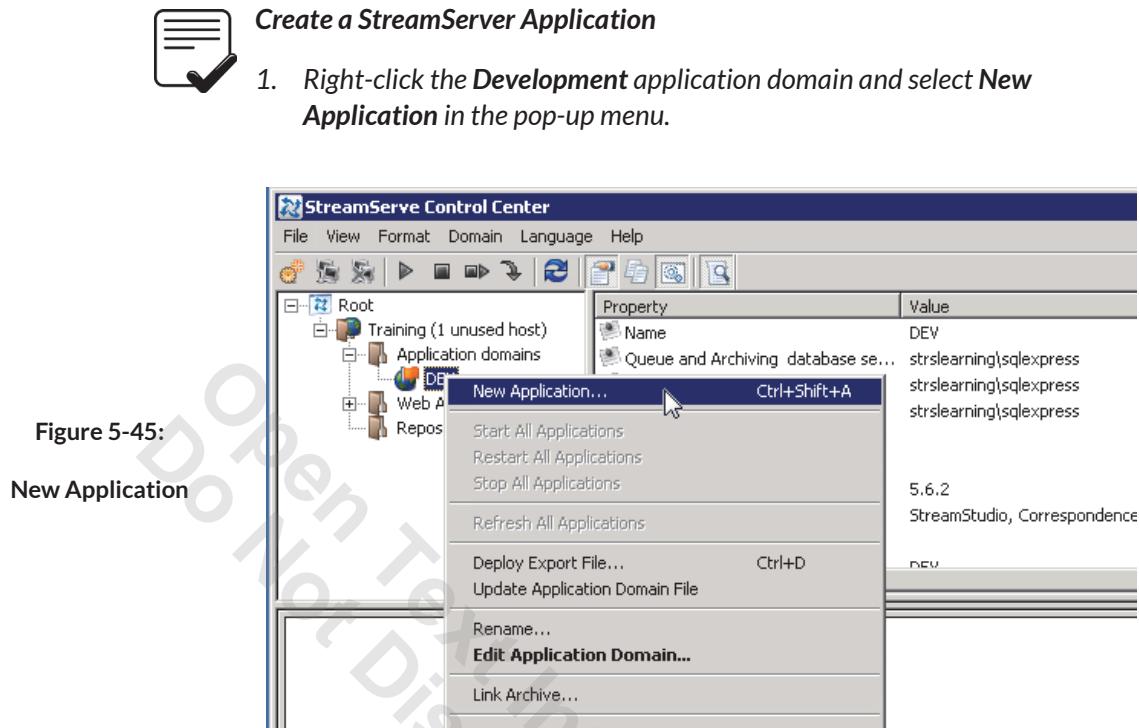


Figure 5-45:  
New Application

1. Right-click the **Development** application domain and select **New Application** in the pop-up menu.
2. In the New Application window enter the following value and click the OK button (leave the default values for the other fields):
  - **Application Name:** AdminCourse

The new AdminCourse application is created and listed under the StrsLearning domain.

Next you will deploy a Design Center project. This step is only to test that the installation is working properly, it is NOT an actual post-installation required step.



### Deploy a Design Center Project

1. Right-click the **AdminCourse** application and select **Deploy Export File** in the pop-up menu:

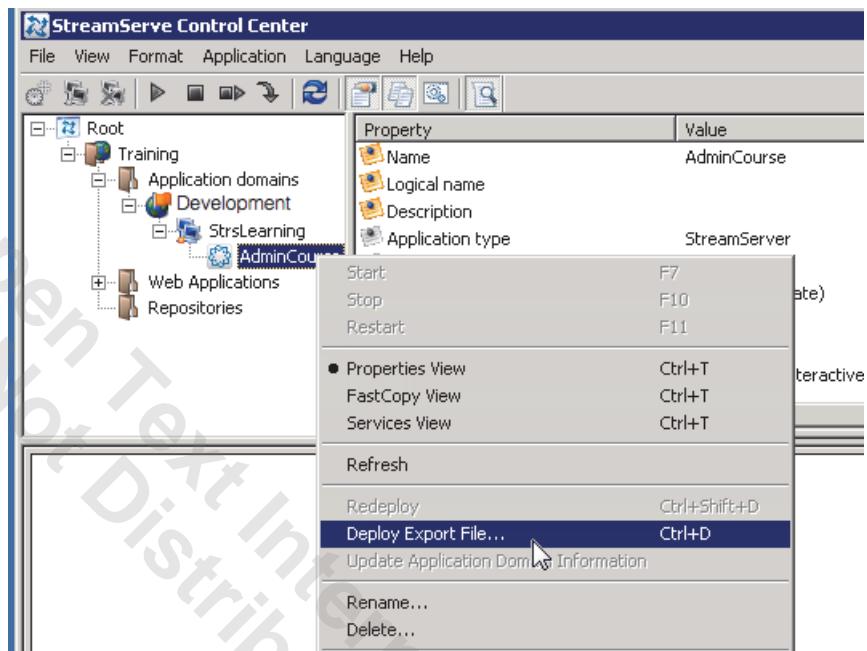


Figure 5-46:

### Deploying Export File

2. In the Deploy window select the **Deploy an export file from the file system** option and then click the **Browse** button.

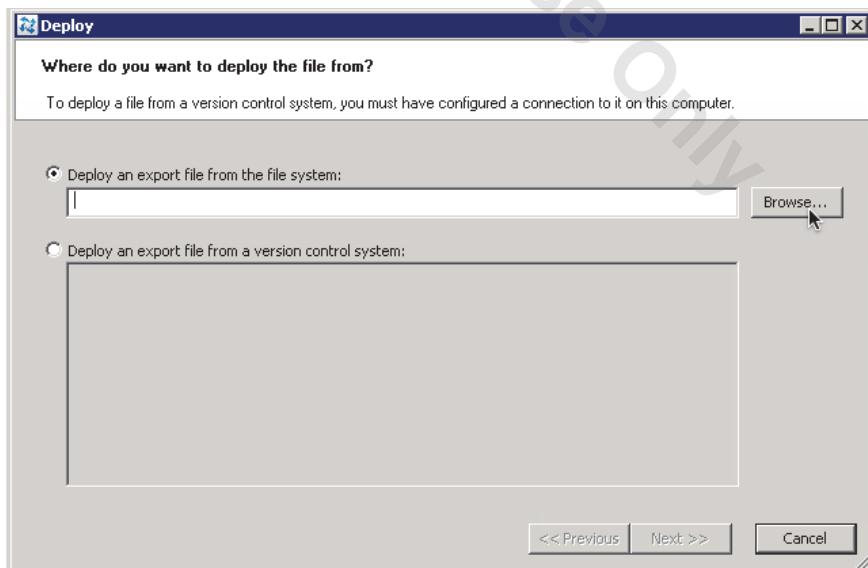


Figure 5-47:

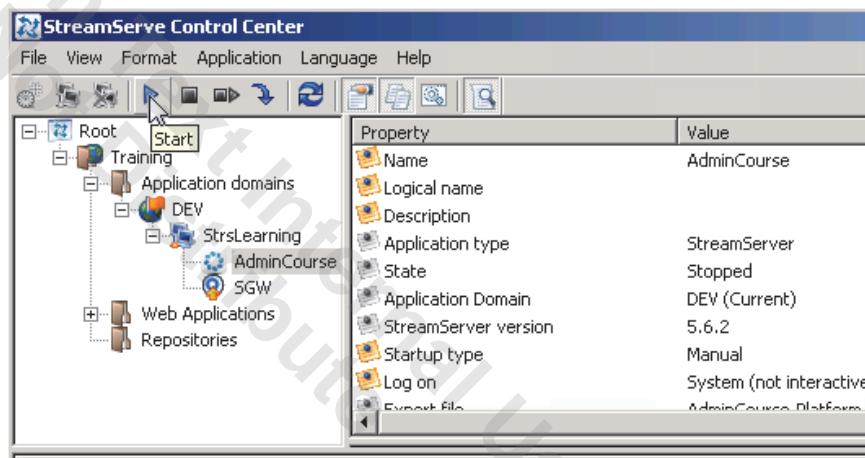
### Deploying Export File

3. Select the **AdminCourse-Platform.export** file located in **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles** and click the **Open** button.
4. Click the **Next** button.
5. In the **Select the physical layer to deploy** window, make sure that **Development** is selected and click the **Finish** button.

An information window displays indicating that the deploy operation was successful.

6. Click the **OK** button.
7. Making sure the **AdminCourse** application is selected, click the **Start** button in the toolbar.

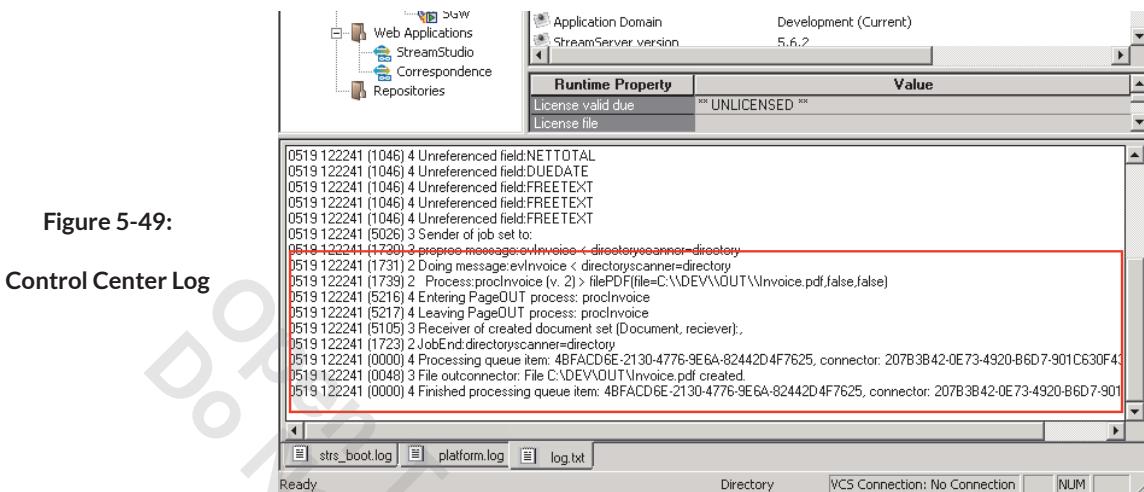
Figure 5-48:  
Starting AdminCourse



#### Test - Run the Project

1. Copy the **recordInvoice.txt** file
  - From: **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles**
  - To: **C:\DEV\IN**

2. View the Control Center log to check that the file has been processed.



3. Check the C:\DEV\OUT folder and make sure that the **Invoice.pdf** file was produced.



### Prepare the Application Domain for Web Applications

1. In Control Center expand Training (1 unused host) > Application domains.
2. Right-click **Development** and select **Edit Application Domain** in the pop-up menu.

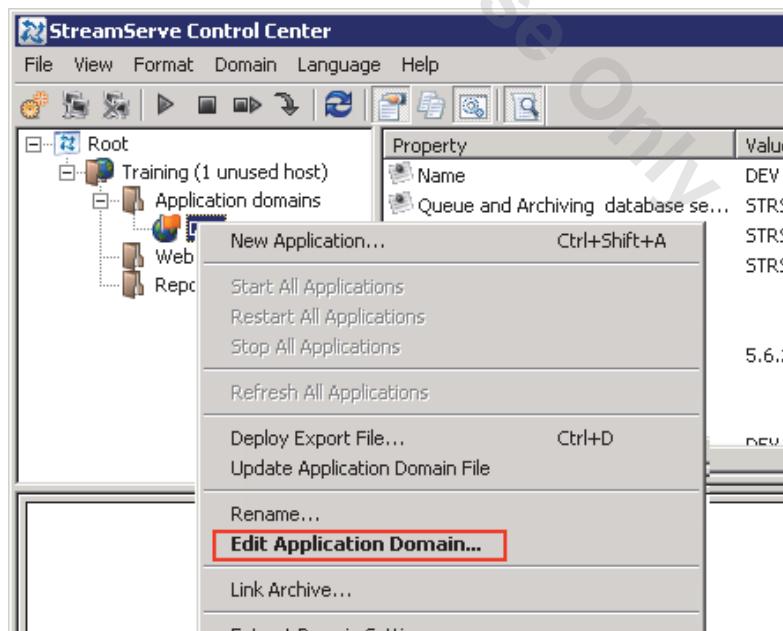


Figure 5-50:

### Edit Application Domain

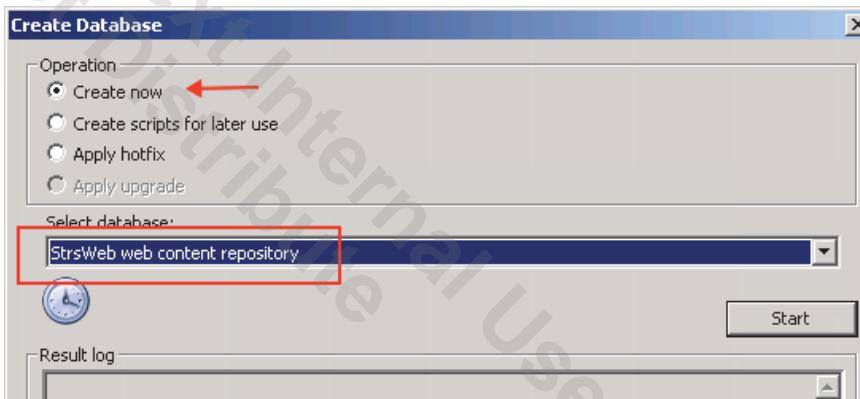
3. In the Application Domain Editor window select the **Web Content Repository** tab, enter the following values and click the **OK** button:
  - Disable web content repository settings: **Unchecked**
  - Host name (*instance name*): **STRSLEARNING\SQLEXPRESS**
  - Password: **streamserve**
4. Click the **Yes** button to confirm the changes made to the application domain.

You are taken back to the Control Center interface.

5. Right-click **Development** and select **Create Database** in the pop-up menu.
6. In the Create Database window enter the following values and click the **Start** button:
  - Operation: **Create now**
  - Select database: **StrsWeb web content repository**

Figure 5-51:

Database Parameters



7. In the Connect window enter the following values and click the **OK** button.
  - User name: **sa**
  - Password: **streamserve**

After a few seconds the database creation completes, make sure there are no errors.

8. Click the **Close** button.

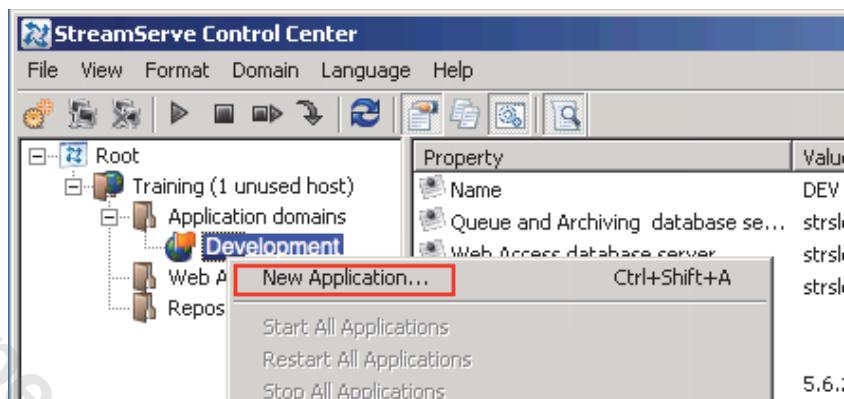


#### Add the Service Gateway Application for StreamStudio

1. In Control Center expand **Training (1 unused host) > Application domains**.

2. Right-click **Development** and select **New Application** in the pop-up menu.

**Figure 5-52:**  
New Application

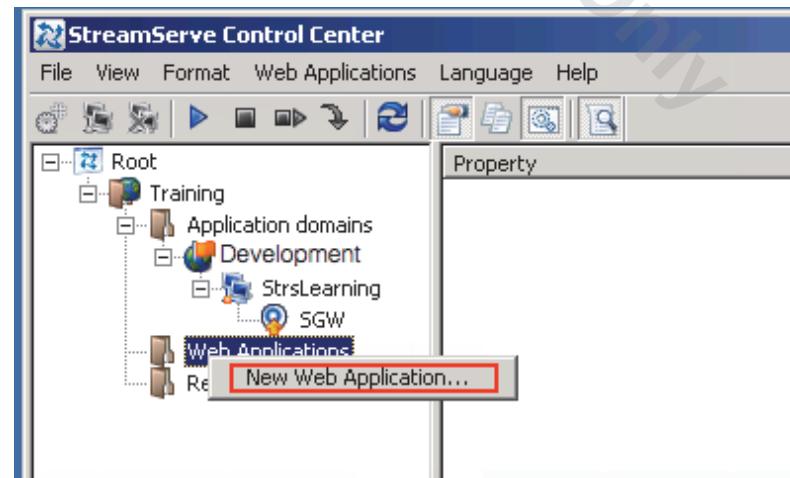


3. In the New Application window enter the following values and click the **OK** button:
  - Application type: **Service Gateway 5.6.2**
  - Application name: **SGW**
4. Click the **Yes** button to confirm the changes made to the application domain.

You are taken back to the Control Center interface.

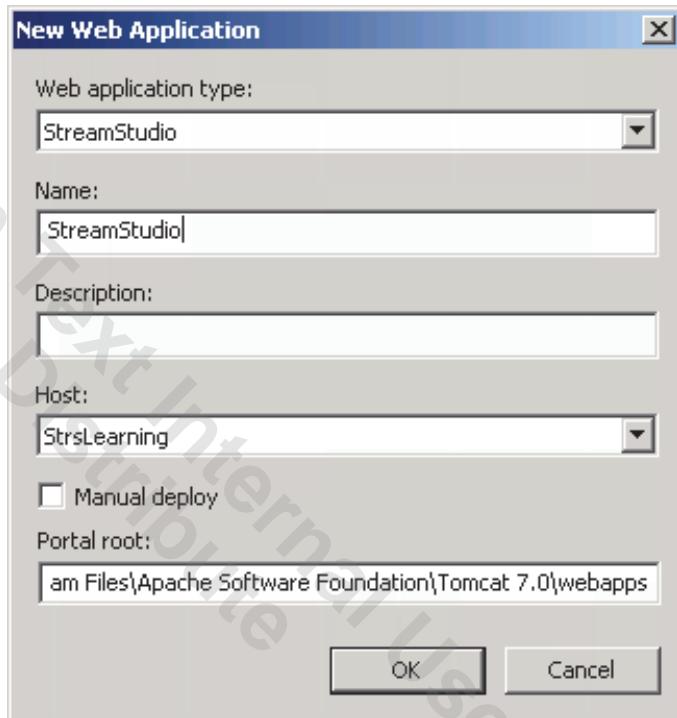
**Create the StreamStudio Web Application**

**Figure 5-53:**  
New Web Application



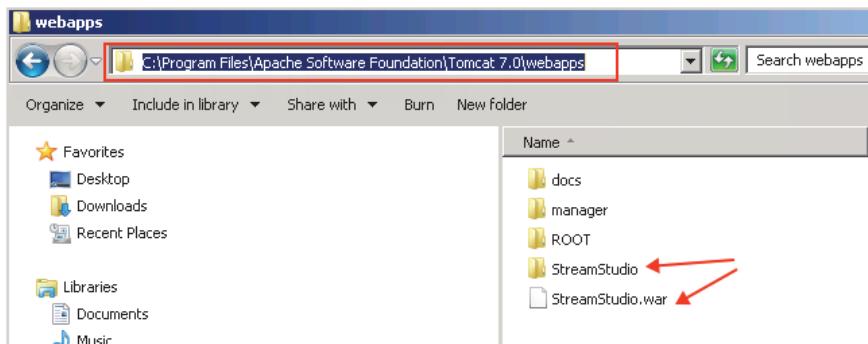
2. In the New Web Application window enter the following values and click the **OK** button:
  - Web application type: **StreamStudio**
  - Name: **StreamStudio**
  - Portal root: **C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps**

Figure 5-54:  
New Web Application  
Data



3. In Windows Explorer navigate to **C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps** and verify that the **StreamStudio.war** was added and deployed.

Figure 5-55:  
StreamStudio Web  
Application Deployed



4. In Control Center right-click **StreamStudio** and select **Link Application Domain** in the pop-up menu.

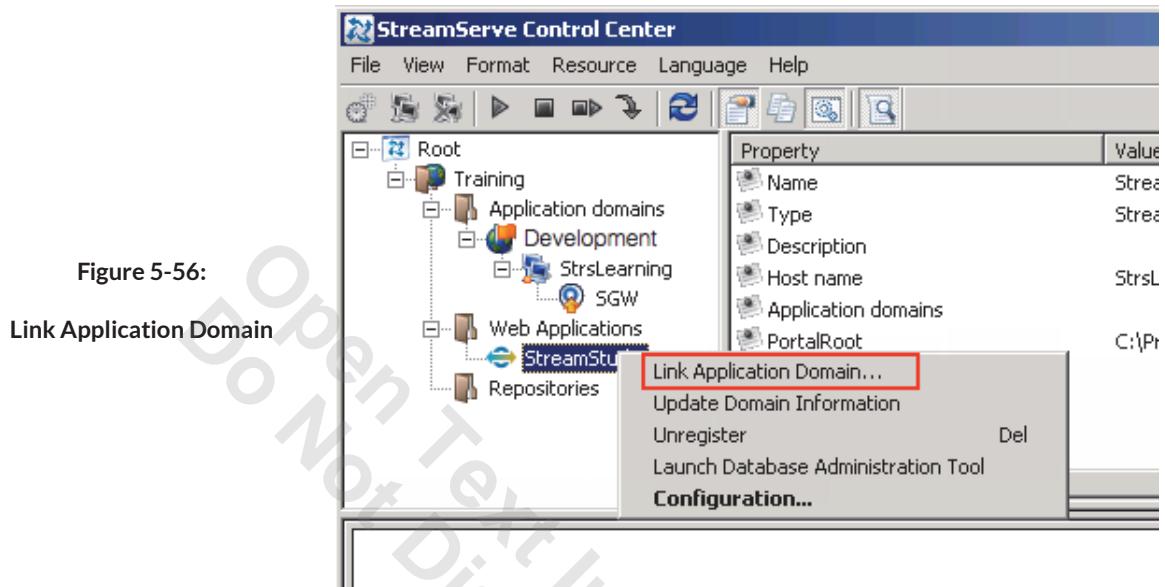


Figure 5-56:  
Link Application Domain

5. In the Link Application Domains window select **Development** in the Available Application Domains, then select the right arrow to move it to the Linked Application Domains panel and click the **OK** button.

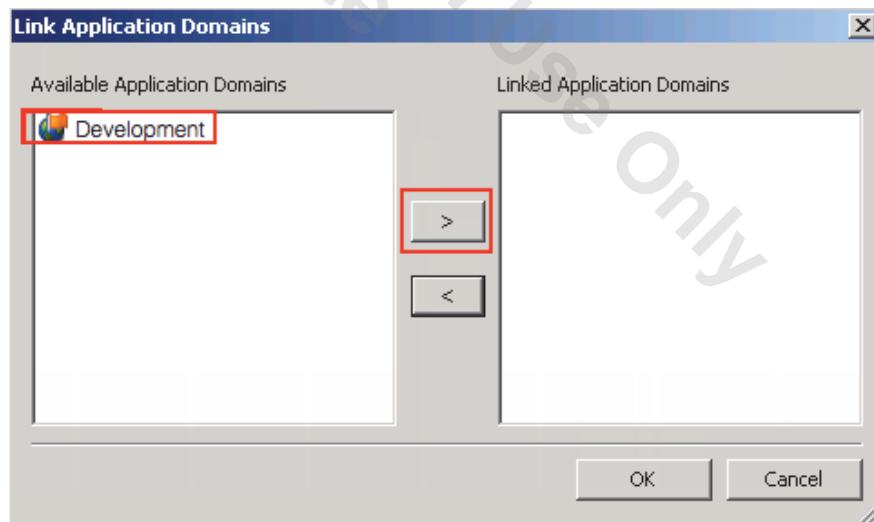
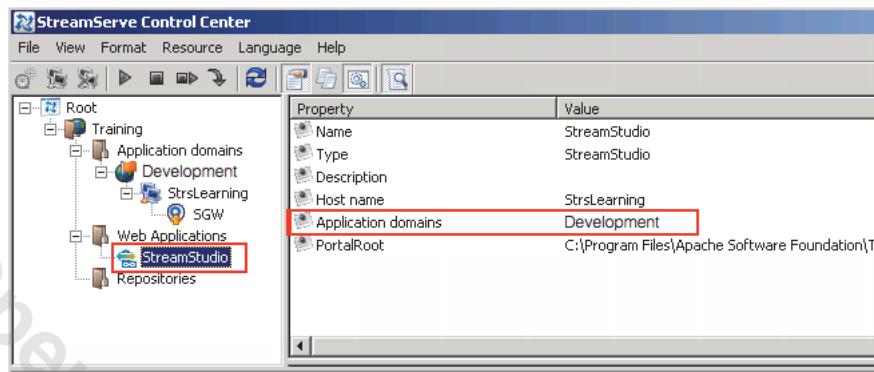


Figure 5-57:  
Linking to the Application  
Domain

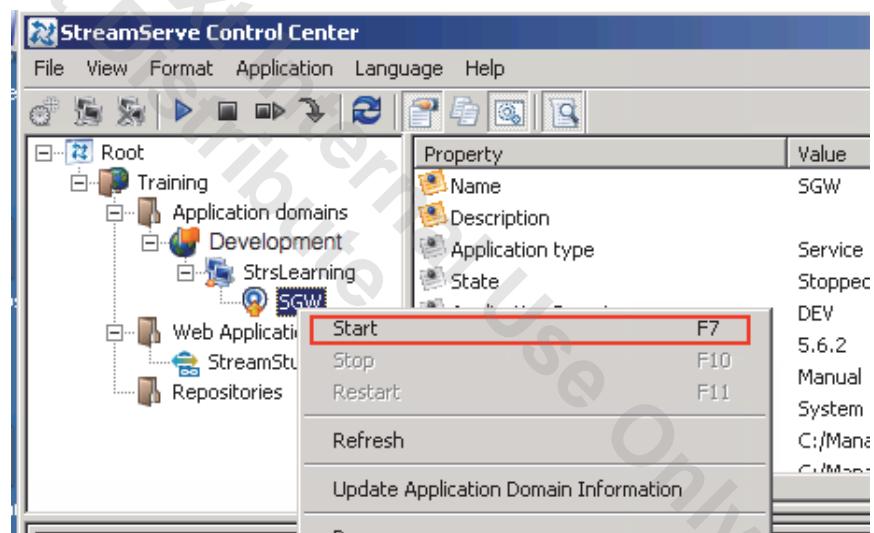
The StreamStudio web application now shows linked to the Development application domain.

**Figure 5-58:**  
**StreamStudio Linked to Development**



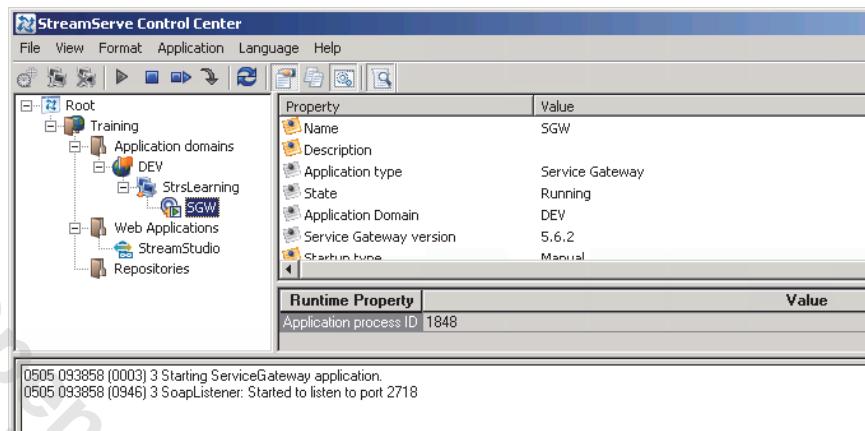
6. Right-click SGW and select **Start** in the pop-up menu.

**Figure 5-59:**  
**Starting SGW**



Make sure the SGW application is started with no errors:

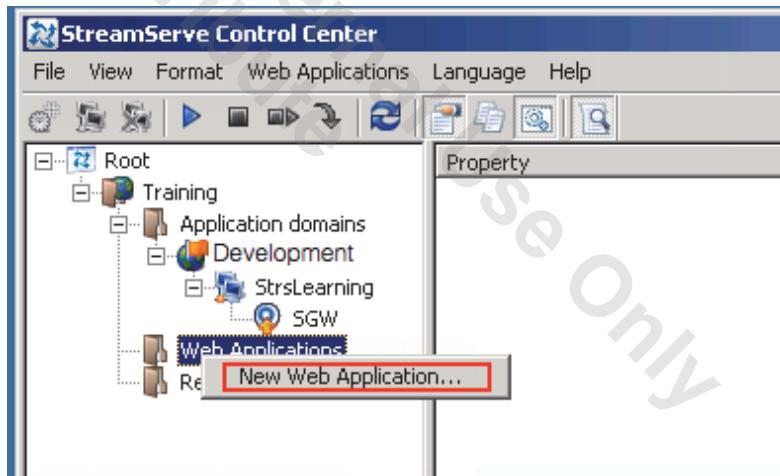
Figure 5-60:  
SGW Started



#### Create the Ad Hoc and Reviewer Web Application

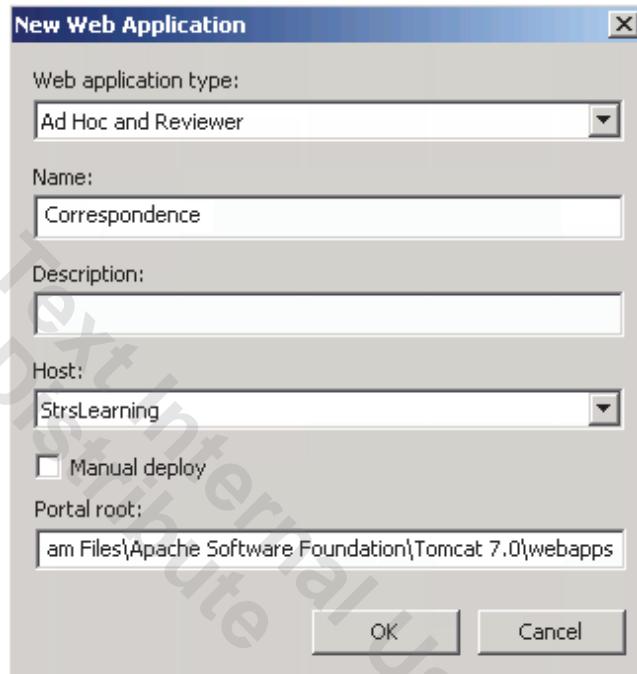
1. Right-click **Web Application** and select **New Web Application** in the pop-up menu.

Figure 5-61:  
New Web Application



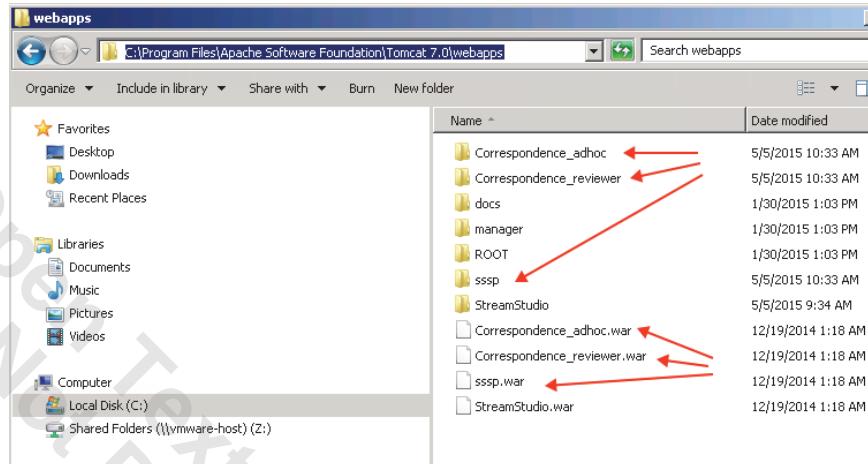
2. In the New Web Application window enter the following values and click the **OK** button:
  - Web application type: **Ad Hoc and Reviewer**
  - Name: **Correspondence**
  - Portal root: **C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps**

Figure 5-62:  
New Web Application  
Data



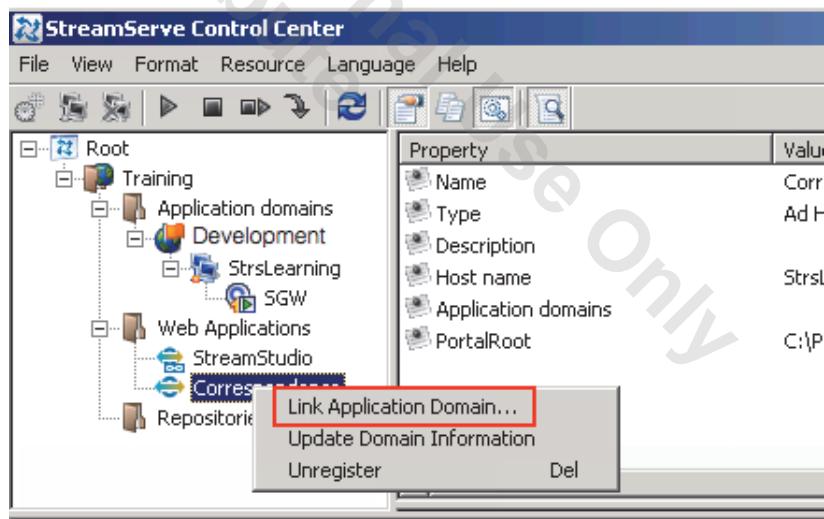
3. In Windows Explorer navigate to **C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps** and verify that the **Correspondence\_adhoc.war**, **Correspondence\_reviewer.war** and **sssp.war** were added and deployed.

**Figure 5-63:**  
Ad Hoc and Reviewer Web Applications Deployed



4. In Control Center right-click **Correspondence** and select **Link Application Domain** in the pop-up menu.

**Figure 5-64:**  
Link Application Domain



5. In the Link Application Domains window select **Development** in the Available Application Domains, then select the right arrow to move it to the Linked Application Domains panel and click the **OK** button.

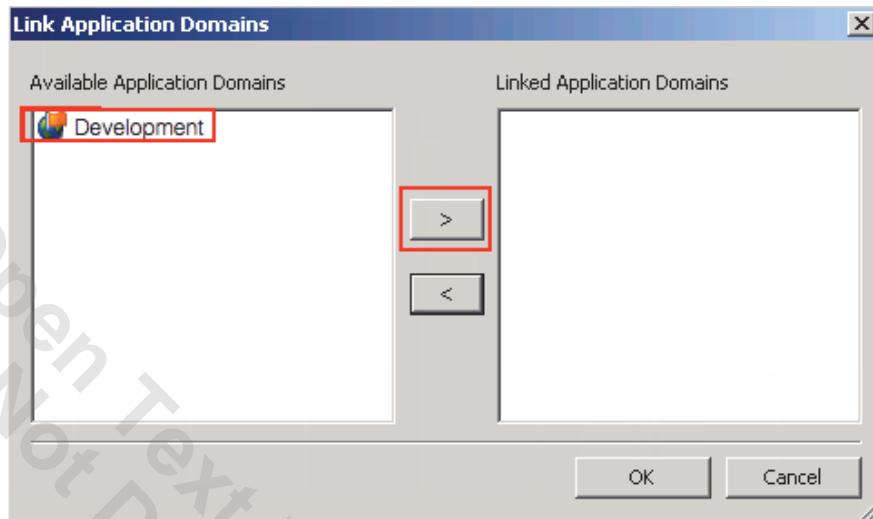


Figure 5-65:  
Linking to the Application  
Domain

The StreamStudio web application now shows linked to the Development application domain:

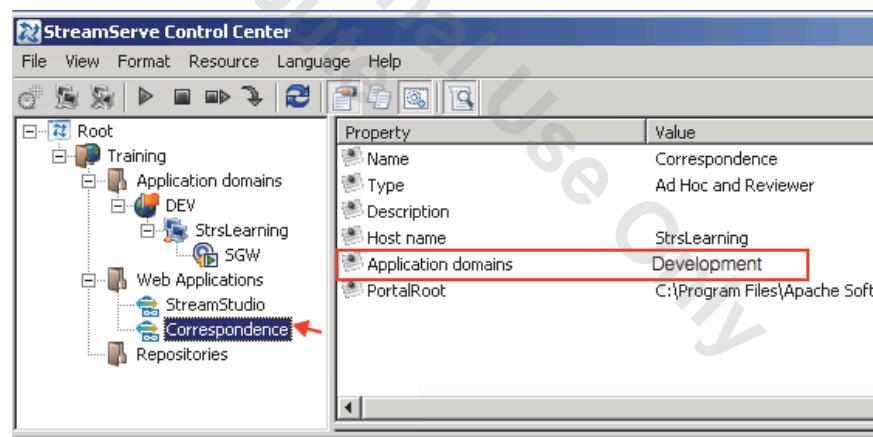


Figure 5-66:  
Correspondence Linked to  
Development

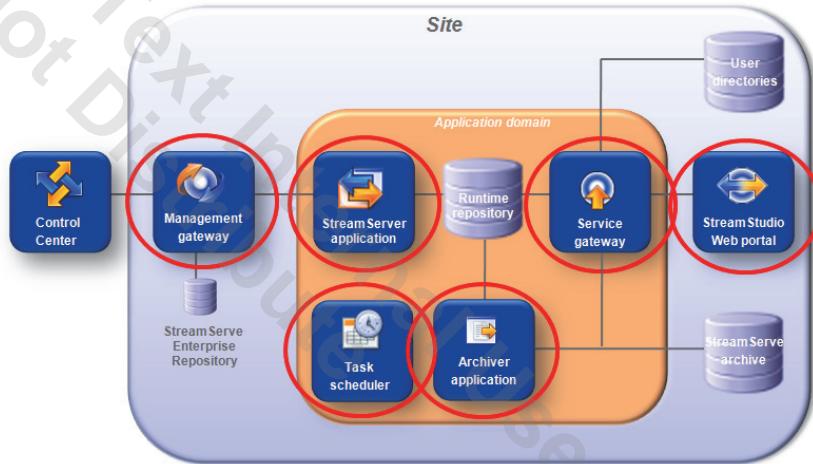
## 6. System Monitoring

On completion of this (section/module), participants should be able to:

- Describe and locate StreamServe components and log files
- Log to the database
- Monitor with Surveillance
- Configure Java Notifications
- Configure Status Messenger
- Configure Reporter

### StreamServe Components and Log Files

**Figure 6-1:**  
StreamServe Components Logs



Log message files are created for the following StreamServe components:

- StreamServer
- Management Gateway
- Service Gateway
- Task Scheduler
- Archiver
- StreamStudio



Logging for StreamStudio and Archiver is not covered in this course. For information about logging for these components, refer to the StreamServe online help.

## StreamServer Application Log Files

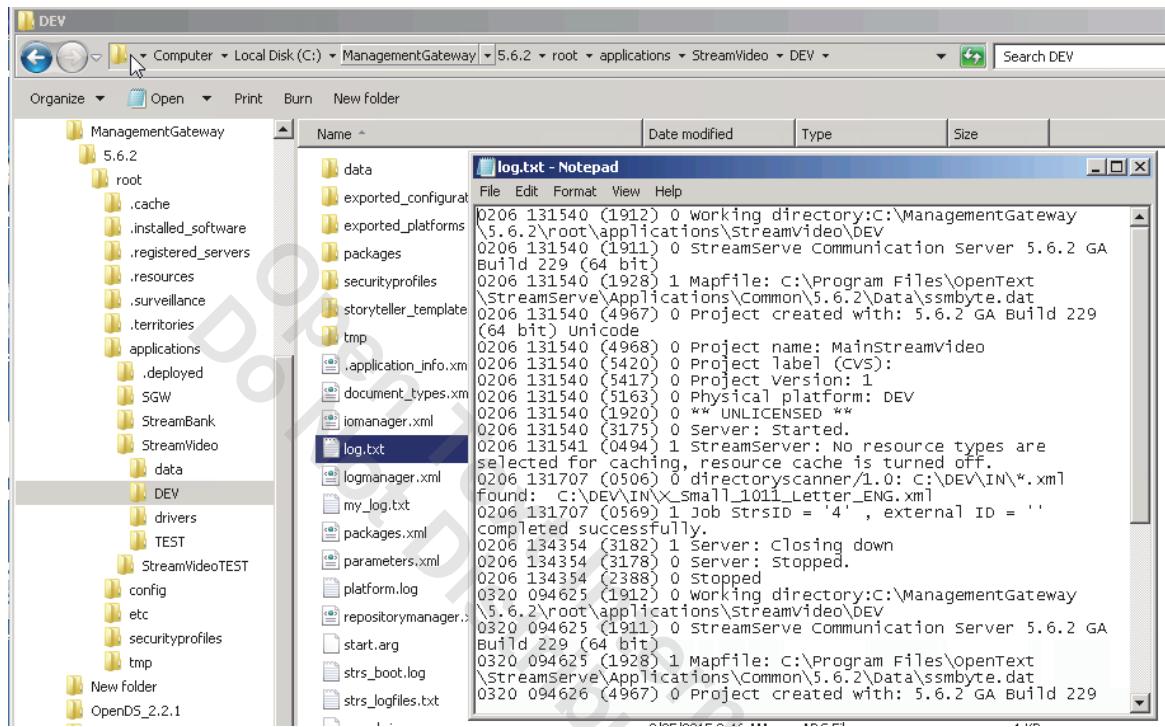


Figure 6-1: StreamServer Application Log Files

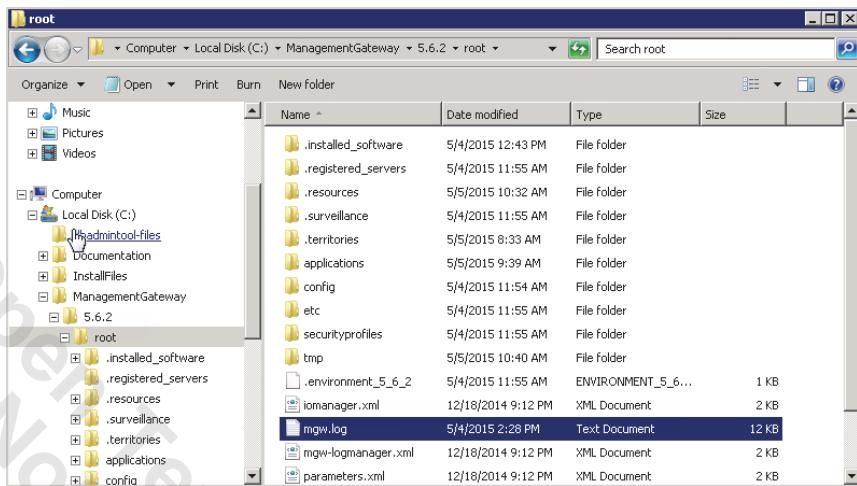
**Information Included** StreamServer applications generate log files with information about start up, run and stop.

**Log Files For Physical layers** Each physical layer has a separate log file, located in the corresponding working directory.

**Log File Name** The default name of the log file is log.txt. The file name can be changed by project developers in Design Center.

## Management Gateway Log File

Figure 6-2:  
ManagementGateway Log



**Information Included** The management gateway generates a log file with information about the connection to Control Center.

**Name and Location** The name of the file is mgw.log and the default location is <InstallDrive>\ManagementGateway\5.6.2\root.

## Service Gateway Application Log File

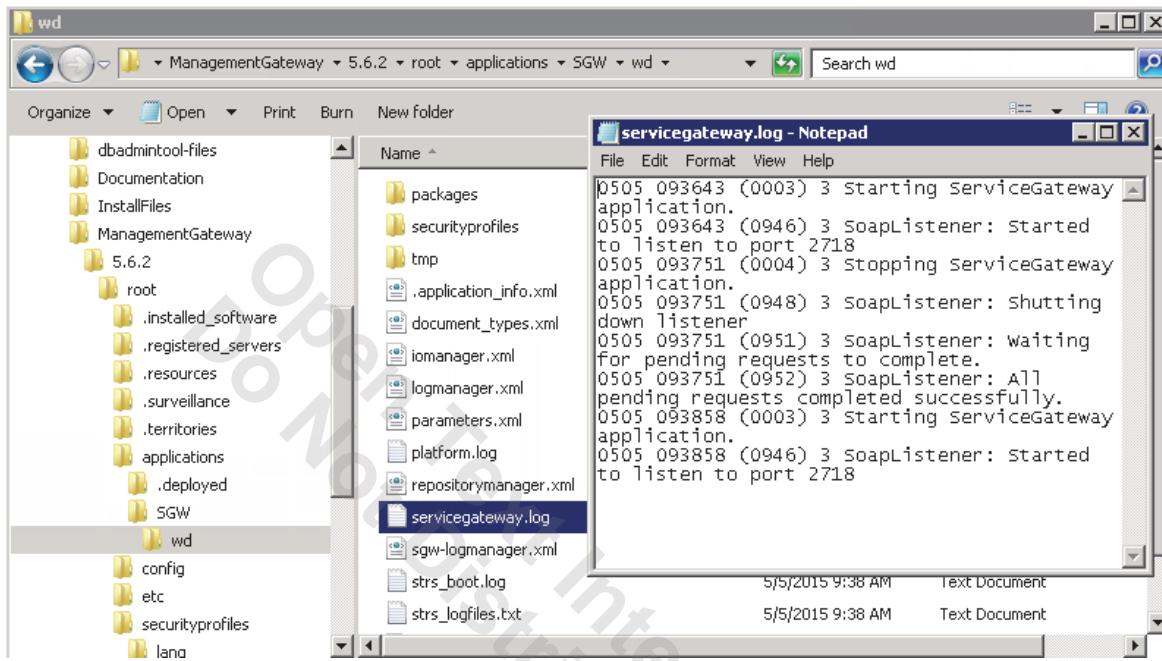


Figure 6-3: Service Gateway Application Log File

**Information Included** The service gateway generates log messages with information about the connection to StreamStudio.

**Name and Location** The name of the log is `servicegateway.log` and the default location is `<InstallDrive>\ManagementGateway\5.6.2\root\applications\<ServiceGatewayApplicationName>\wd`

## Task Scheduler Application Log File

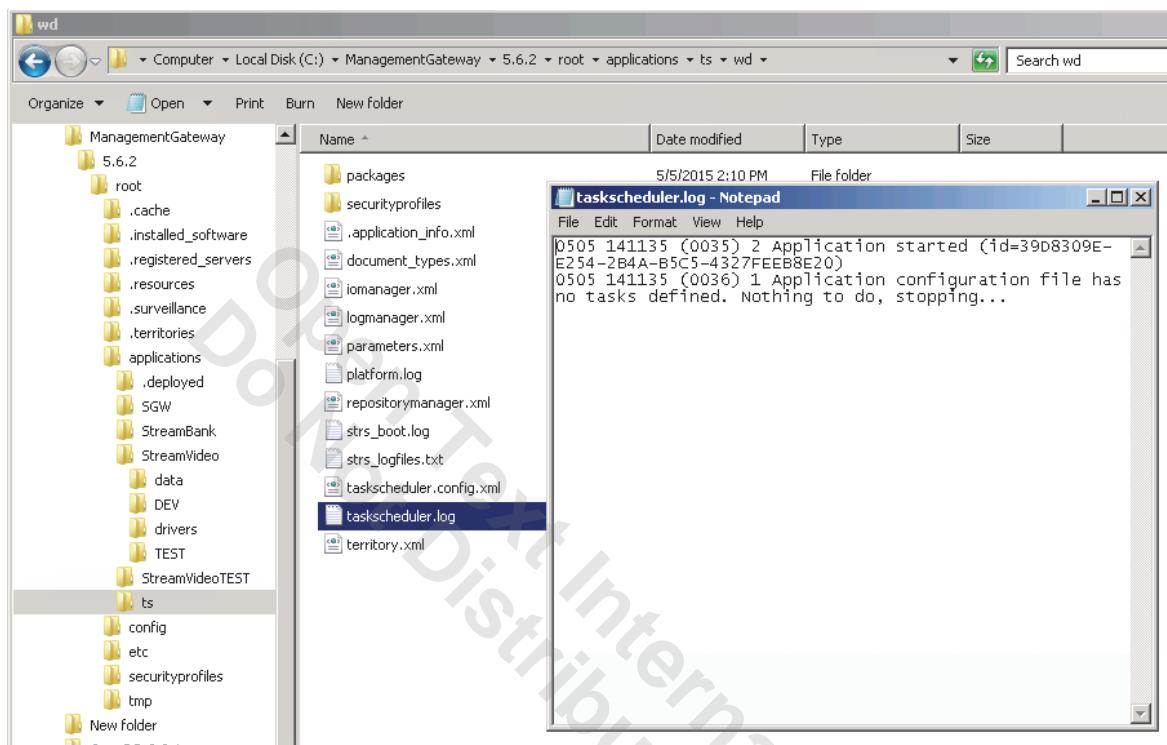


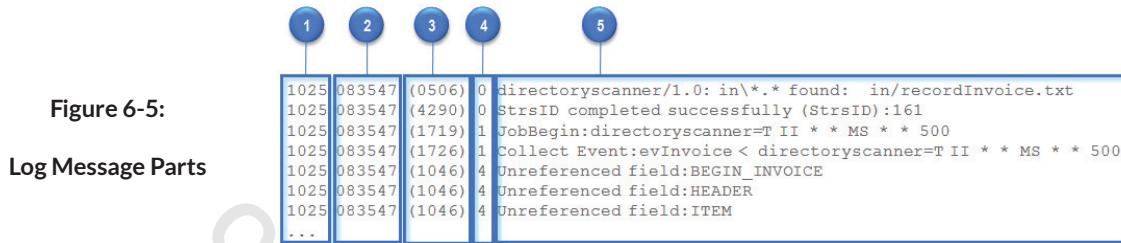
Figure 6-4: Task Scheduler Log File

**Information Included** The Task Scheduler application generates a log file with information about startup, run and stop.

**Name and Location** The name of this log is `taskscheduler.log` and the default location is `C:\ManagementGateway\1.0\root\applications\<ApplicationName>\wd`

## Log Messages

### Log Message Parts



**Figure 6-5:**

**Log Message Parts**

Entry descriptions:

1. The date the log entry was created.
2. The time the log entry was created.
3. The log message ID.
4. The severity level of the log message.
5. The log message.

This log message structure is used for the following StreamServe applications:

- StreamServer
- Management Gateway
- Service Gateway
- Task Scheduler
- Archiver

## Log Messages Included

**Figure 6-6:**  
Log Message Always Included

```

log.txt - Notepad
File Edit Format View Help
0823 132541 (1911) 0 StreamServe Communication Server 5.5.0 LA Build 391
0823 132541 (1928) 1 Mapfile: C:\Program Files\StreamServe\Applications\Streamserver\5.
0823 132541 (1912) 0 Working directory: C:\ManagementGateway\1.0\root\applications\Fundamentals
0823 132541 (4967) 0 Project created with: 5.5.0 LA Build 391 Unicode
0823 132541 (4968) 0 Project name: Fundamentals
0823 132541 (4969) 0 Project version:
0823 132541 (5163) 0 Physical platform: development
0823 132541 (1920) 0 ** UNLICENSED **
0823 132541 (3175) 0 Server: Started.
0823 133104 (0506) 0 directoryscanner/1.0: C:\DEV\IN\*.* found: C:\DEV\IN\fieldInvoice
0823 133104 (4290) 0 Strsib completed successfully (StrsID):1
0823 133138 (0506) 0 directoryscanner/1.0: C:\DEV\IN\*.* Found: C:\DEV\IN\fieldInvoice
0823 133138 (4290) 0 Strsib completed successfully (StrsID):3

```

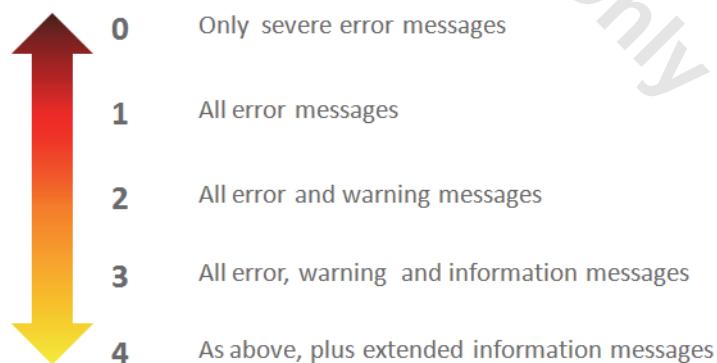
Some message types are always included in the log. For example:

- The active working directory.
- The active Project and physical layer.
- When the application is started and stopped.

Other messages are included based on the log level specified for the application. For example, if a log level of 0 is specified, severe error messages are included. To include all log messages, use log level 99.

## Log Message Severity Levels

**Figure 6-7:**  
Log Levels



**Level 0** Only include severe error messages in the log. This is the recommended option for the production layer due to performance.

- Level 1** Include all error messages in the log. This can be used in the production layer to get more information.
- Level 2** As the above, plus warning messages. This can also be used in the production layer to get even more information.
- Level 3** As the above, plus information messages. This is the recommended option for the development and test layers.
- Level 4** This can be used in the development and test layers to get more information. Note that this option may complicate the search for relevant error messages, due to all information messages displayed in the log.

### Log Examples

These log examples illustrate four scenarios where the file recordInvoice.txt is retrieved via a Directory input connector, and the output file Invoice.pdf is delivered via a File output connector.

**Scenario 1 – Severe messages only, successful processing** In this scenario, the log level is set to 0.

```
1025 083132 (0506) 0 directoryscanner/1.0: in\*.* found: in/recordInvoice.txt  
1025 083132 (4290) 0 StrsID completed successfully (StrsID):155
```

**Scenario 2 – All messages, successful processing** In this scenario, the log level is set to 4. The bold message lines show the difference between scenario 1 and scenario 2.

```
1025 083547 (0506) 0 directoryscanner/1.0: in\*.* found: in/recordInvoice.txt  
1025 083547 (4290) 0 StrsID completed successfully (StrsID):161  
1025 083547 (1719) 1 JobBegin:directoryscanner=T II * * MS * * 500  
1025 083547 (1726) 1 Collect Event:evInvoice < directoryscanner=T II * * MS * *  
500  
1025 083547 (1046) 4 Unreferenced field:BEGIN_INVOICE  
1025 083547 (1046) 4 Unreferenced field:HEADER  
1025 083547 (1046) 4 Unreferenced field:ITEM  
...  
1025 083547 (5026) 3 Sender of job set to:anonymous  
1025 083547 (1730) 3 preproc event:evInvoice < directoryscanner=T II * * MS * *  
500  
1025 083547 (1731) 1 Doing Event:evInvoice < directoryscanner=T II * * MS * * 500  
1025 083547 (1739) 2 Process:procInvoice > filePDF(RecordIN_2006-10-25_083547)  
1025 083547 (5105) 3 Receiver of created document set (Document,  
reciever):,anonymous  
1025 083547 (1723) 1 JobEnd:directoryscanner=T II * * MS * * 500  
1025 083547 (0048) 3 File outconnector: File C:\DEV\OUT\Invoice.pdf created.
```

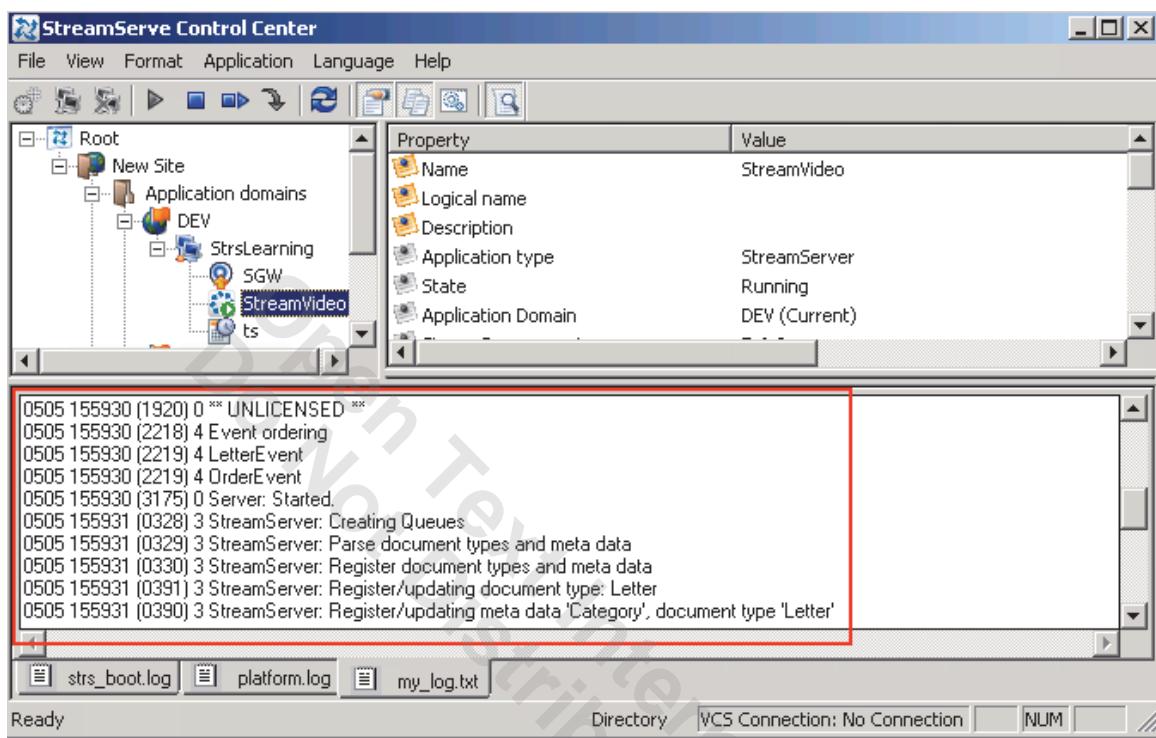
**Scenario 3 – Severe messages only, failed to process** In this scenario, the log level is set to 0. The output is not delivered because the output directory was not found. Note that the log result in this scenario is the same as in scenario 1 because the error is not severe.

```
1025 084345 (0506) 0 directoryscanner/1.0: in\*.* found: in/recordInvoice.txt  
1025 084345 (4290) 0 StrsID completed successfully (StrsID):167
```

**Scenario 4 – All messages, failed to process** In this scenario, the log level is set to 4. The output is not delivered because the output directory was not found. The bold message lines show the difference between scenario 3 and scenario 4.

```
1025 083923 (0506) 0 directoryscanner/1.0: in\*.* found: in/recordInvoice.txt  
1025 083923 (4290) 0 StrsID completed successfully (StrsID):164  
1025 083923 (1719) 1 JobBegin:directoryscanner=T II * * MS * * 500  
1025 083923 (1726) 1 Collect Event:evInvoice < directoryscanner=T II * * MS * *  
500  
1025 083923 (1046) 4 Unreferenced field:BEGIN_INVOICE  
1025 083923 (1046) 4 Unreferenced field:HEADER  
1025 083923 (1046) 4 Unreferenced field:ITEM  
...  
1025 083923 (5026) 3 Sender of job set to:anonymous  
1025 083923 (1730) 3 preproc event:evInvoice < directoryscanner=T II * * MS * *  
500  
1025 083923 (1731) 1 Doing Event:evInvoice < directoryscanner=T II * * MS * * 500  
1025 083923 (1739) 2 Process:procInvoice > filePDF(RecordIN_2006-10-25_083923)  
1025 083923 (5105) 3 Receiver of created document set (Document,  
reciever):,anonymous  
1025 083923 (1723) 1 JobEnd:directoryscanner=T II * * MS * * 500  
1025 083923 (0613) 1 File outconnector: Unable to append/copy/move from temporary  
file to C:\DEV\OUT\Invoice.pdf  
1025 083923 (0201) 1 Failed to process queue item 01E8DBF6-6943-4DD0-9960-  
1E10D97D1C75
```

## Logs in Control Center



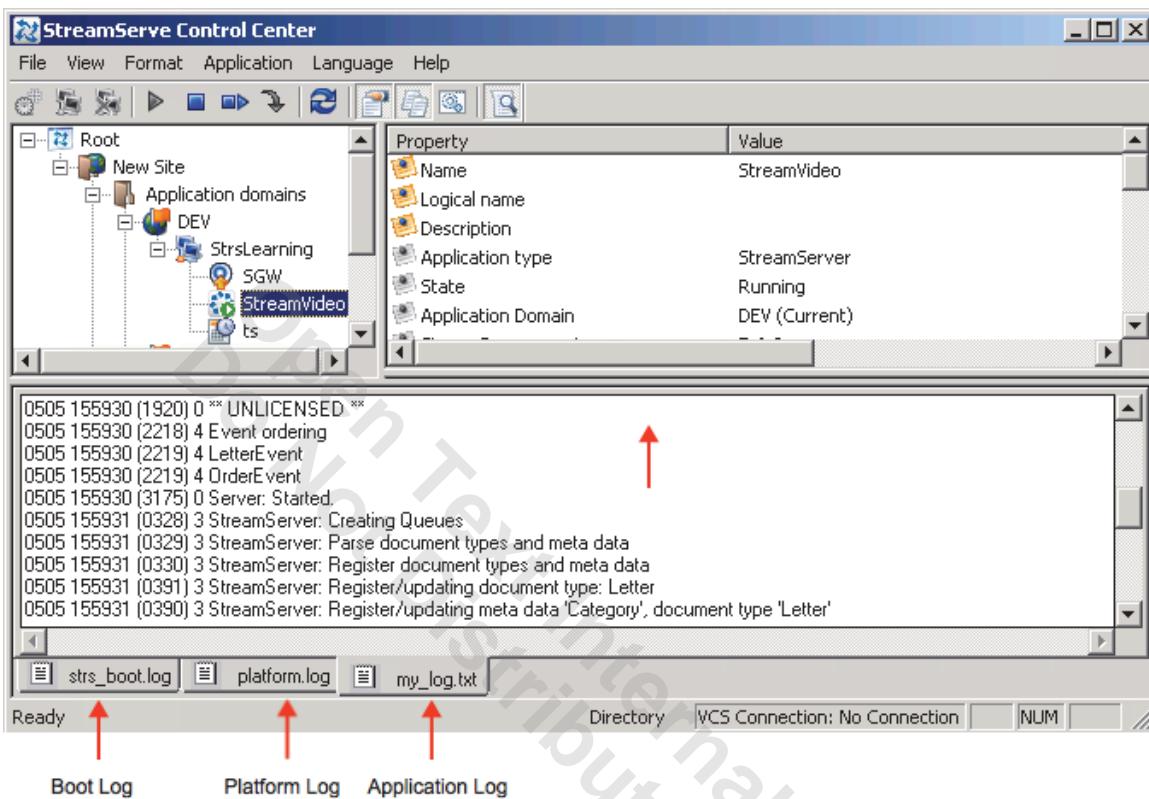
**Figure 6-8: Logs in Control Center**

You can view and manage the logs for applications in Control Center.

The logs are viewed on separate tabs in the Control Center log view. For example, when a service gateway is selected in the tree view, the log view displays the log files relevant to that application.

Each tab contains the same information as the corresponding log file. However, if you clear the information displayed on the log tab, it does not affect the corresponding log file.

## Boot, Platform and Application Logs



Each application (StreamServer, management gateway, service gateway, Task Scheduler and Archiver) generates three separate logs (location indicated next):

- Boot log – early startup messages
- Platform log – low level details
- Application log – runtime log messages

### Management Gateway Logs

**Boot Log** <ManagementGatewayRoot>/strs\_boot.log

**Platform log** <ManagementGatewayRoot>/platform.log

**Application log** <ManagementGatewayRoot>/mgw.log

## Service Gateway Logs

**Boot Log** <Service gateway wd>/strs\_boot.log

**Platform log** <Service gateway wd>/platform.log

**Application log** <Service gateway wd>/servicegateway.log

## StreamServer Logs

**Boot Log** <StreamServer wd>/strs\_boot.log

**Platform Log** <StreamServer wd>/platform.log

**Application Log** <StreamServer wd>/<logfile>

Where <logfile> is the log file name specified in Design Center. The default is log.txt.

## Task Scheduler Logs

**Boot Log** <Task scheduler wd>/strs.log

**Platform Log** <Task scheduler wd>/platform.log

**Application Log** <Task scheduler wd>/taskscheduler.log

## Setting Log Levels

Log levels may be set at the following levels:

### Platform Log Levels

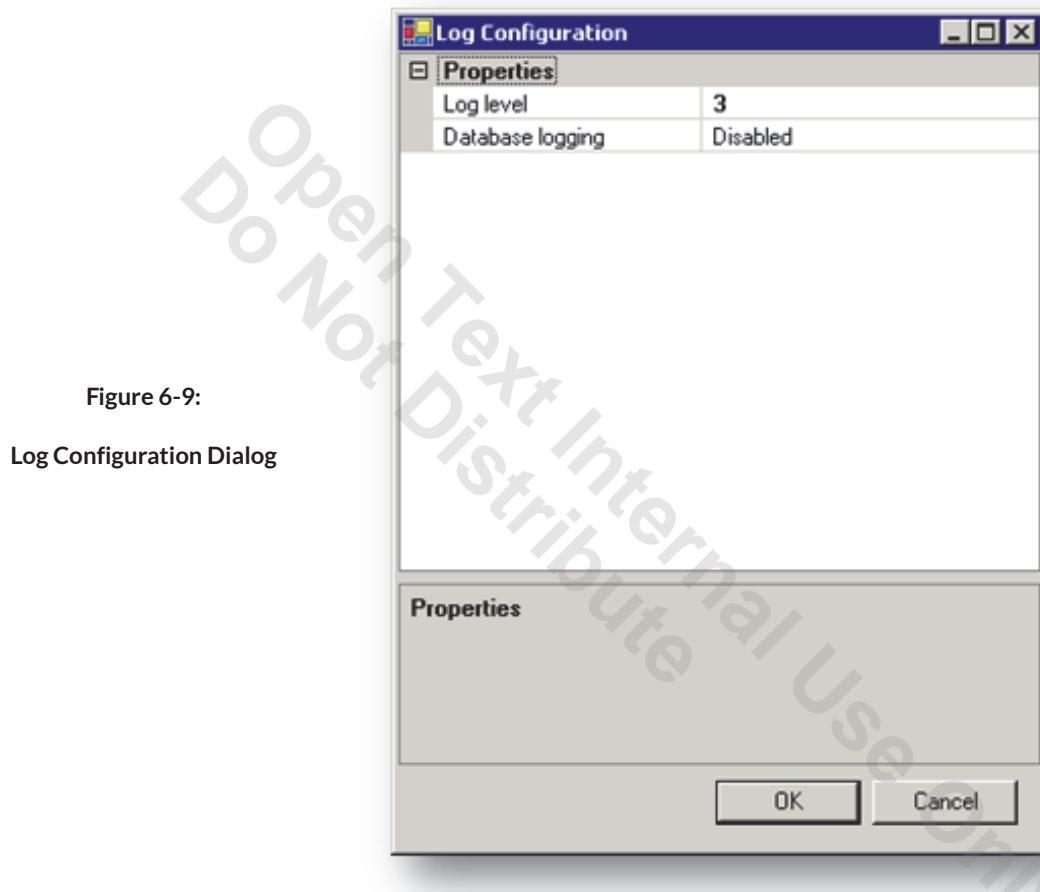


Figure 6-9:

Log Configuration Dialog

You can specify the platform log level for the following StreamServe components in Control Center:

- StreamServer
- Task Scheduler
- Service gateway
- Archiver

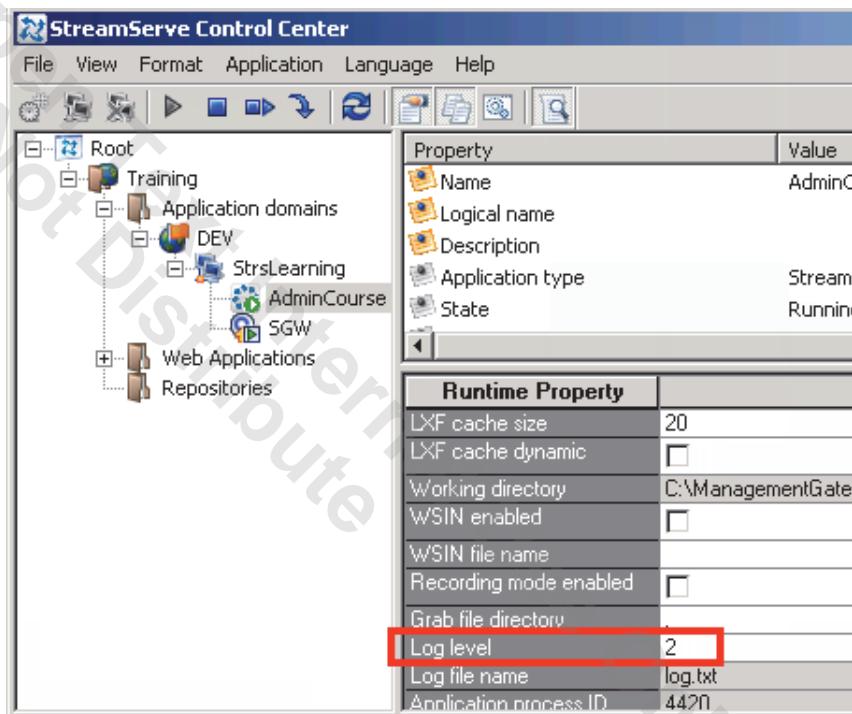
To specify the platform log level:

1. Stop the application.
2. In the Control Center tree view, right-click the application and select Log Configuration. The Log Configuration dialog box opens.
3. Change Log level to the appropriate level and click OK.
4. Start the application.

### Setting StreamServer Application Log Level

Figure 6-10:

StreamServer Application Log Level



The log level for StreamServer application logs is specified in Design Center. Each time you start a StreamServer application, the log level is set to the same value as specified in Design Center.

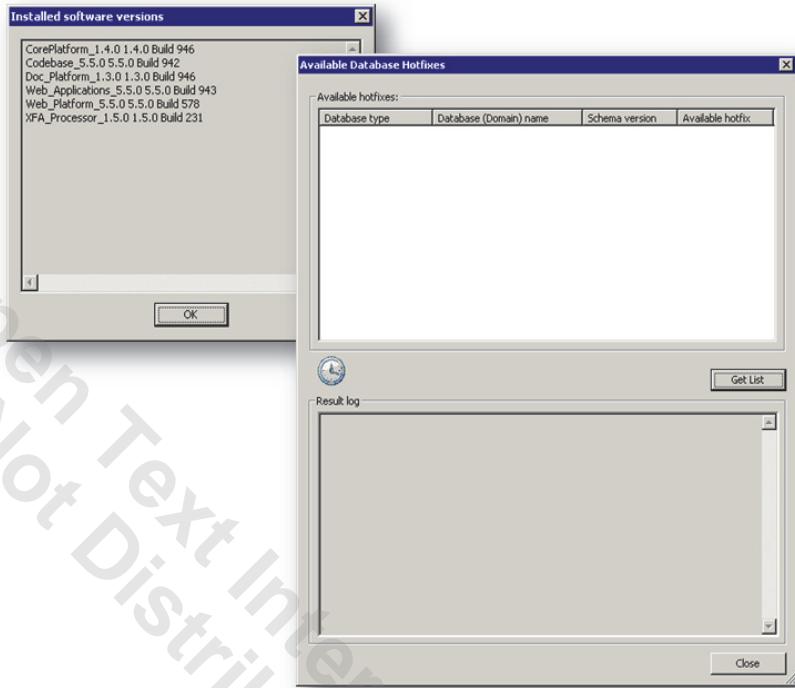
You can change the log level for a running StreamServer application from Control Center, without having to change the log level in Design Center and export/redeploy to the StreamServer application.

To change the log level for a running StreamServer application:

1. Select the StreamServer application node in the tree view.
2. In the Runtime Property view, change Log level to the appropriate level.

## Information Used to Log Support Incidents

Figure 6-11:  
Information Used to Log  
Support Incidents



You may be required to provide the following information when you log an incident with StreamServe Support:

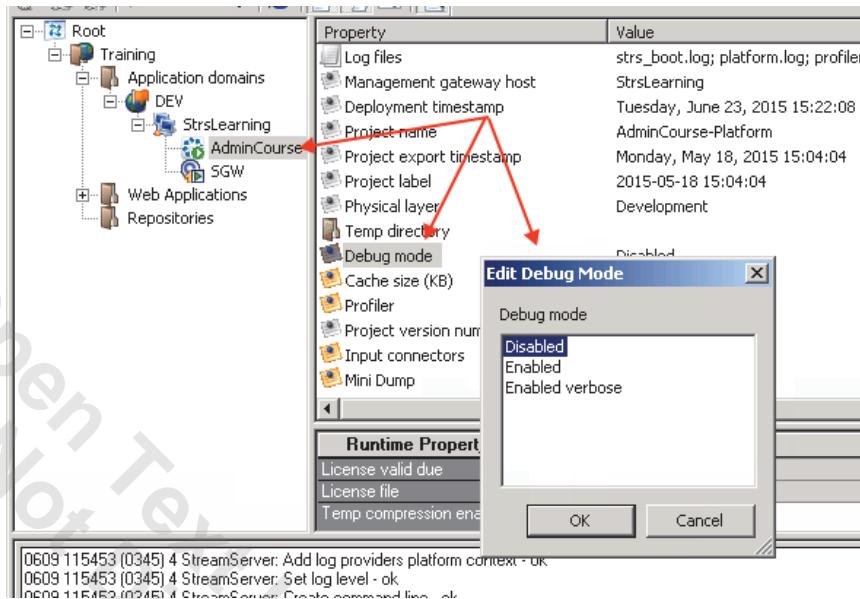
- StreamServe components with version information installed on a computer.
- List of the repositories at your site, together with their current schema versions and the latest available hotfixes.

This window displays the StreamServer components on the computer.

This Window lists the repositories at the site, along with status and version information.

## Debugging Applications

**Figure 6-12:**  
Debugging Applications



You can enable debugging and include all available log information in the logs. This applies to all logs (boot log, platform log, and application log), for the selected application.

To enable debugging:

1. Select the application node in the tree view.
2. Stop the application.
3. In the Properties view, double-click Debug. The Edit Debug mode dialog box opens.
4. Select Debug and click OK.
5. Start the application.

## Logging to the Database

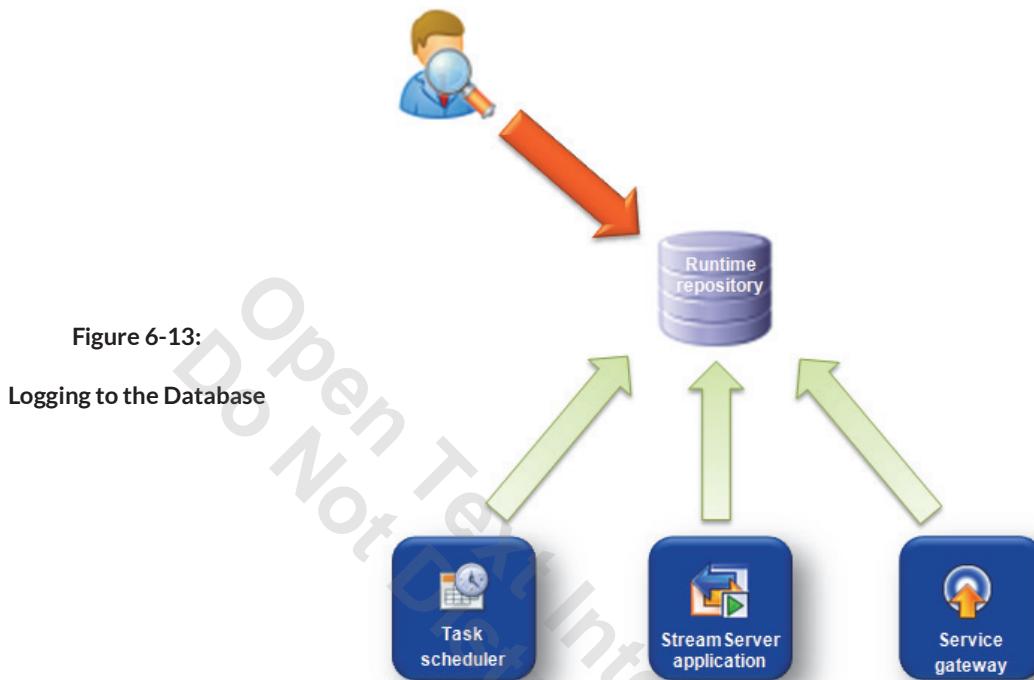


Figure 6-13:  
Logging to the Database

You can enable database logging for each application, which means log messages are stored in the runtime repository.

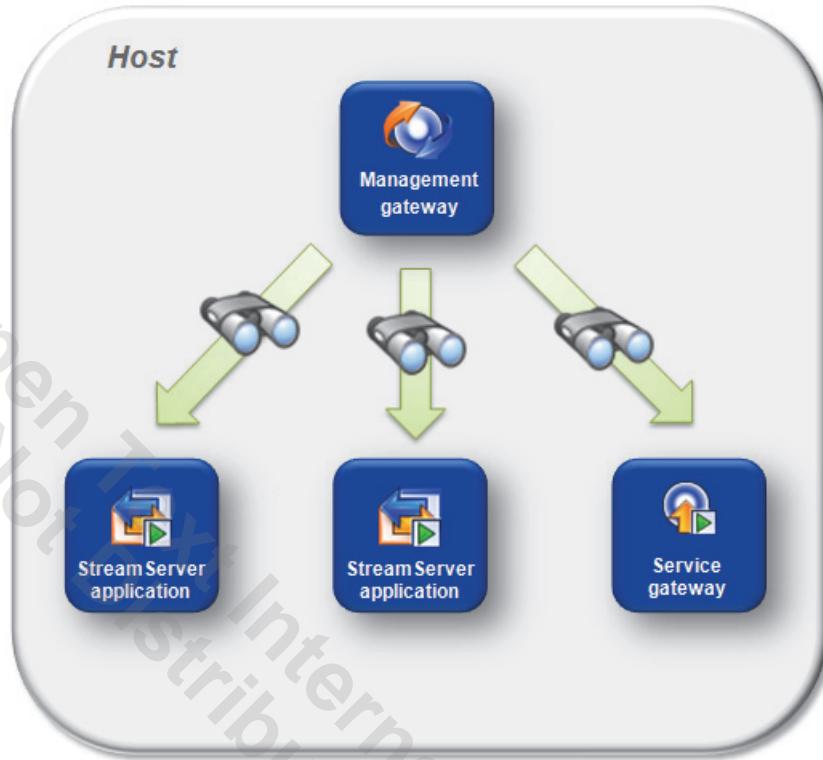
Additional entries are included in log messages when you log to the database (e.g. External log ID, Thread ID). Logging to the database also provides better control of the logs than using log files, since you can examine the logs from several applications using date, job ID, etc, as search criteria.

If logging to the database is enabled for an application, a Database log tab is displayed in Control Center. This tab shows the log information stored in the runtime repository.

In order to display the information on the Database log tab, a service gateway must be connected to the same application domain as the application. The service gateway must also be running.

## Monitoring with Surveillance

Figure 6-14:  
Monitoring with  
Surveillance



You can use the surveillance functionality to monitor the status of the following StreamServe applications:

- StreamServer
- Service gateway
- Task Scheduler
- Archiver

Surveillance monitors the preferred state of an application. The preferred state of the application is set when the application is started or stopped from Control Center.

Example – monitoring the preferred state

For example, if you start an application from Control Center, the preferred state is set to Started. Then, if applications are started or stopped outside of Control Center, for example from Windows Control Panel, the preferred state does not change.

Starting and stopping the application (service) from Windows Control Panel means changing the current state - not the preferred state.

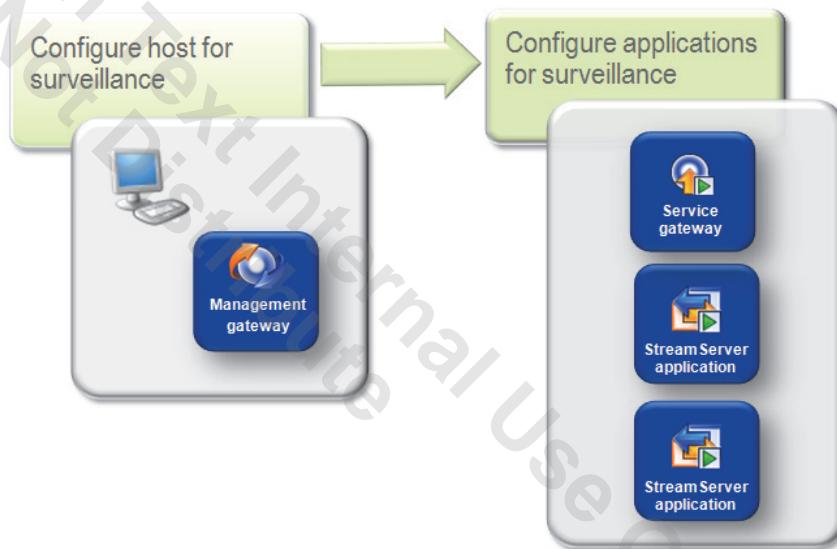
If an application that is monitored and that has a preferred state of Started goes down, you can use the surveillance functionality to try to restart the application.

If restarting is not successful, it is also possible to:

- Start an external program, which can try to restart the application. For example, by sending an SNMP trap to a Network Management System.
- Send an email to the application domain contact.

## Configuring Surveillance

Figure 6-15:  
Configuring Surveillance

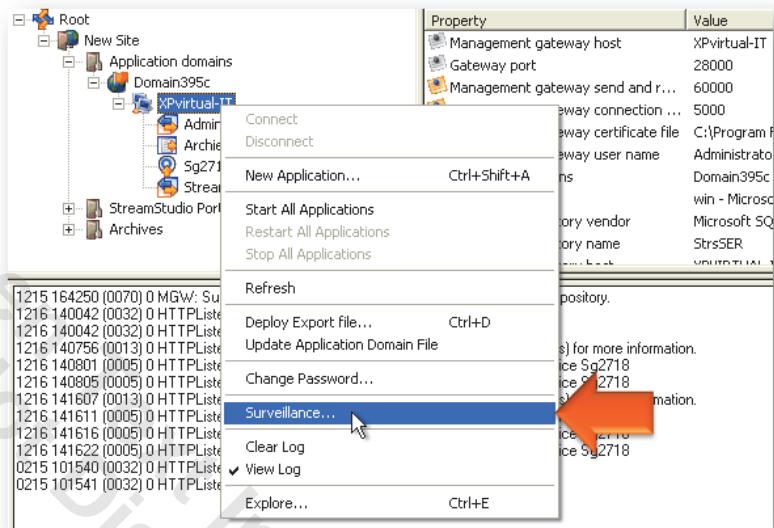


To configure surveillance to monitor an application you must:

1. Configure failure actions for the host.
2. Enable surveillance for an application and specify which failure actions to take.

## Configuring Surveillance Actions for the Host

**Figure 6-16:**  
Configuring the Host



You must specify the surveillance actions that can be taken for the StreamServe applications on each host.

To configure the surveillance actions for the application on host

1. Right-click the host node and select Surveillance. The Host Surveillance dialog box opens.
2. Configure the surveillance options.
3. Click OK to save the changes.

**Host Surveillance Dialog Box** This dialog box is used to define surveillance actions that can be taken for applications on the host.

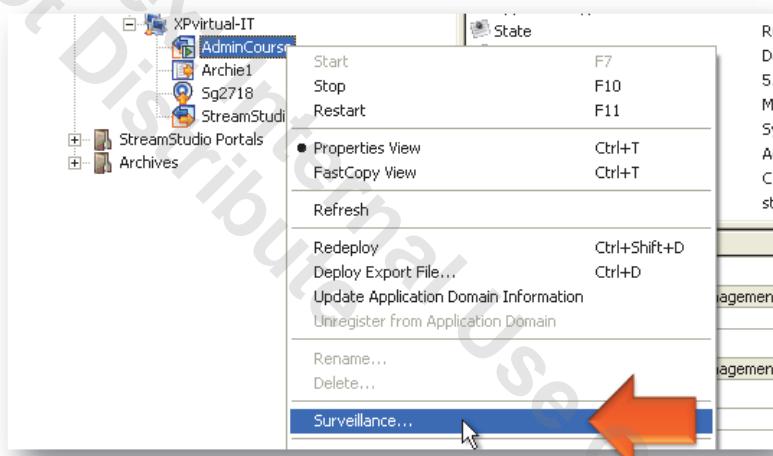
If the preferred state of an application cannot be reestablished by the surveillance actions, an email can be sent to the application domain contact. The application domain contact is defined in the Application Domain Editor, on the General tab.

Setting	Description
Mail server	The IP address or host name of the mail server used to send surveillance emails.
Mail domain	The domain name for the mail server. For example: streamserve.com

Setting	Description
Return address	The return address for surveillance emails. This address is displayed in the sender field
Alternative email address	The email address to use if no contact is specified for the application domain, or if the email sent to the application domain contact fails.
Mail time-out (ms)	The time the management gateway waits for a response from the mail server. This is specified in milliseconds.

### Configuring surveillance for Specific StreamServe Applications

Figure 6-17:  
Application Surveillance



You must enable surveillance for each application that you intend to monitor.

You must also specify which of the failure actions are taken for the application. For example, send an email if the preferred state of the application cannot be reestablished.

To enable surveillance for an application and specify actions taken:

1. Right-click the application and click Surveillance. The Application Surveillance dialog box opens.
2. Click Enabled to activate surveillance.
3. Configure the surveillance options.
4. Specify the failure actions.
5. Click OK to save the changes.

Application Surveillance  
Dialog Box

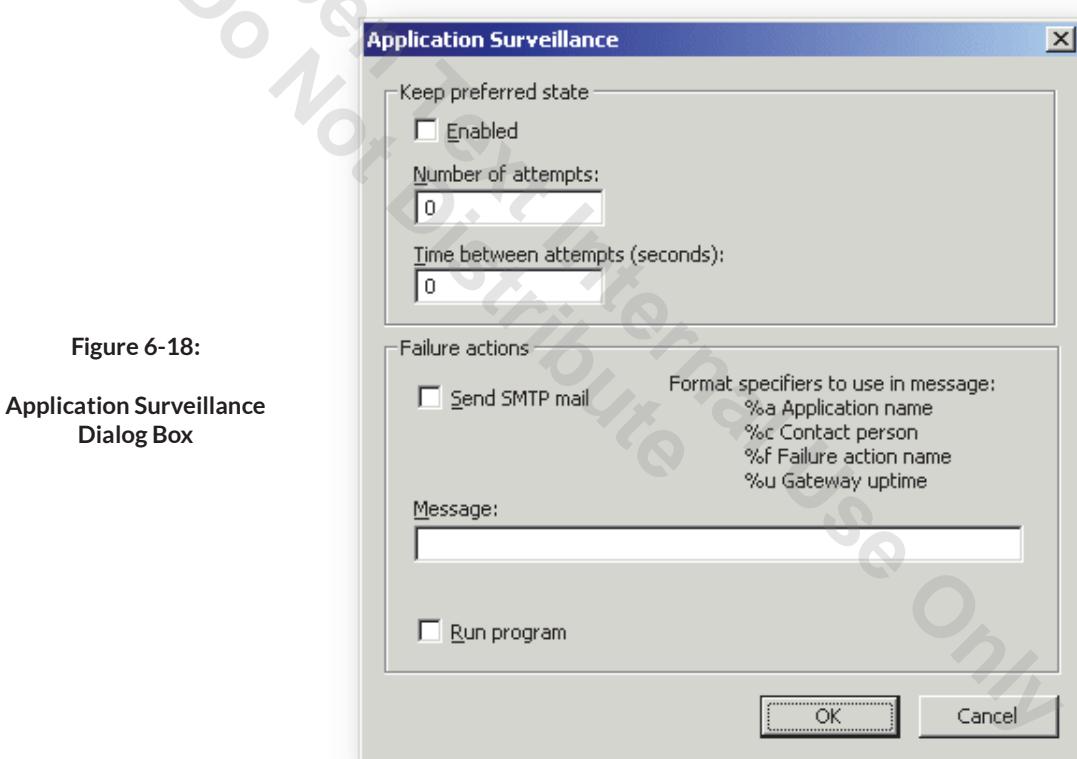


Figure 6-18:  
Application Surveillance  
Dialog Box

The Application Surveillance dialog box is used to enable surveillance for an application, and to select the surveillance actions can be taken for the application.

Setting	Description
Enabled	Switches on the surveillance functionality for the application.
Number of attempts	The number of attempts made to reestablish the preferred state of the application. For example, to restart the application if it goes down.
Time between attempts	The interval between the attempts. This is specified in seconds.
Send SMTP mail	Sends an email if the preferred state of the application cannot be reestablished. For example, if the application cannot be restarted. This email is sent using the options specified at host level. See <i>Host Surveillance dialog box</i> in the user documentation.
Run program	Executes an external program if the preferred state of the application cannot be reestablished. For example, if the application cannot be restarted. The path to the program and the startup options are specified at host level. See <i>Host Surveillance dialog box</i> in the user documentation.

## StreamServe Notifications Overview

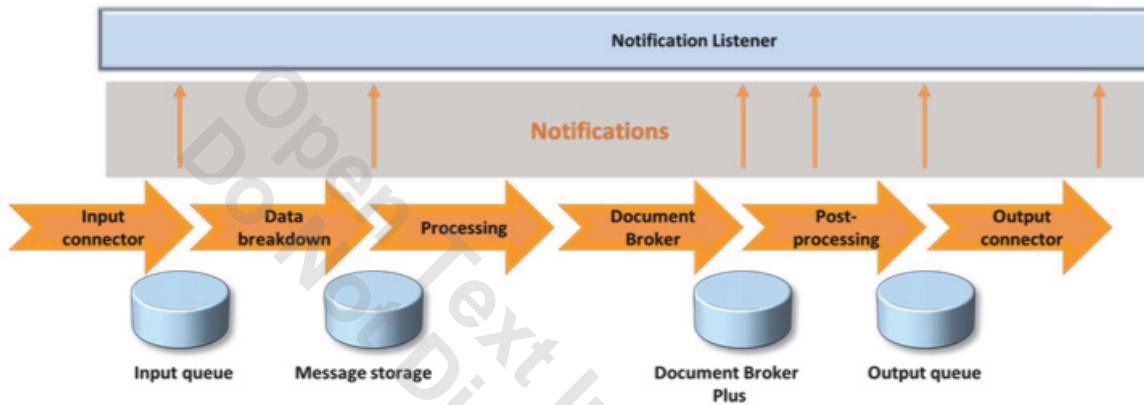
The Notifications API provides the possibility to build your own solutions using the notifications published by StreamServe applications.

To use the Notifications API, a programmer must develop a Java notification listener. The Notifications SDK helps programmers get started with the development and implementation of the notification listener.

StreamServer applications publish notifications when running jobs. Some examples of notifications include completed top jobs, and documents paused in Ad Hoc Correspondence and Correspondence Reviewer.

You can subscribe to different notifications, depending on the requirements of the solution you want to build.

Adding notifications to a StreamServer application may affect the overall performance. Several of the different StreamServer notifications call the notification listener using the StreamServer job thread. It is important to realize that any processing done inside a notification listener affects the StreamServer job processing time. OpenText does not recommend adding heavy weight processing or blocking methods to your notification code.



**Figure 6-19: Notifications Published by StreamServer Applications During Job Processing**

## StreamServer Notification Types

- |   |  |
|---|--|
| <b>Job Related Notifications</b>        | For example, the StreamServer publishes a notification when it starts processing a top job and another notification when it finishes processing the top job.             |
| <b>Document Broker Notifications</b>    | For example, the Post-processor publishes a notification before it creates a new logical document and another notification when it finishes creating a logical document. |
| <b>Composition Center Notifications</b> | For example, StreamServer publishes a notification each time a Composition Center-enabled StoryTeller Process uses a resource.   |
| <b>Sheet Layout Notifications</b>       | For example, StreamServer publishes a notification each time a new sheet is started.   |

**Preparing the StreamServe Application to Use Notifications**

In order to enable Java Notifications for a StreamServe notifications you need to:

- Enable Java.
- Specify location of the notification listener Java files.
- Subscribe to StreamServe notifications.

**Enable Java** You must enable Java for each StreamServe application that should subscribe to notifications using a Java notification listener.

To enable Java for a StreamServe application:

1. In Control Center, right-click the application and select Java Configuration.
2. Depending on the vendor of the JRE or JDK on the computer, select Oracle or IBM.

**Specify Location of the Notification Listener Java Files**

You must specify where the Java files for your notification listener are located. This section describes three ways to do this.

- Option 1: this option is only applicable to StreamServer applications. Add the Java files to the resource set in Design Center. When you deploy the export file, Java files are placed in Data/Java folder in the working directory of your StreamServer application.
- Option 2: In the working directory of your StreamServe application, create a subfolder called Java and place your Java files there.
- Option 3: Place your Java files in another directory and add path to the files to Java class path as a System environment variable or in the Java configuration dialog box for the StreamServe application.

**Subscribe to StreamServer Notifications** In order to subscribe a notification in StreamServer, you need to provide a pre-configured subscription XML file.

```

<?xml version="1.0" encoding="UTF-8"?>
<strs xmlns="http://schemas.streamserve.com/kernel/1.0">
    <modules>
        <module modulepath="kernel">
            <component type="JobEventListener">
                factory="http://schemas.streamserve.com/uid/service/javanotificationlistenerservice/1.0"
                runtime="http://schemas.streamserve.com/uid/javaruntime/1.0"
                <configuration>
                    <java xmlns="http://schemas.streamserve.com/uid/javacomponent/1.0">
                        <class>streamserve.jobstatcollector.JobEventListener</class>
                    </java>
                </configuration>
            </component>
        </module>
    </modules>
    <containers>
        <container type="http://schemas.streamserve.com/uid/component/servicecontainer/1.0">
            <name
value="http://platform.streamserve.com/uid/container/JobEventListenerServiceNotificationContainer/1.0" />
            <configuration>
                <servicecontainer xmlns="http://schemas.streamserve.com/uid/component/servicecontainer/1.0">
                    <servicecontroller type="http://schemas.streamserve.com/uid/component/servicecontroller/1.0">
                        <configuration>
                            <servicecontroller xmlns="http://schemas.streamserve.com/uid/component/servicecontroller/1.0">
                                <services>
                                    <service type="JobEventListener">
                                        <properties>
                                            <name value="BatchJobEventListener" />
                                        <instancing
values="http://schemas.streamserve.com/uid/component/multiuseservicefactory/1.0" />
                                        <registration value="http://schemas.streamserve.com/uid/type/service/registration/local"
/>
                                        </properties>
                                        <subscriptions>
                                            <!-- Add your subscriptions here -->
                                            <subscription type="simple">
                                                <topic>streamserve.notification.streamserver.beginbatch</topic>
                                            </subscription>
                                            <subscription type="simple">
                                                <topic>streamserve.notification.streamserver.endbatch</topic>
                                            </subscription>
                                        </subscriptions>
                                    </service>
                                </services>
                            </servicecontroller>
                        </configuration>
                    </servicecontroller>
                </servicecontainer>
            </configuration>
        </container>
    </containers>
</strs>
```

Figure 6-20: Notifications Published by StreamServer Applications During Job Processing

## Exercise 1 - Change the Log Level at Runtime

In this next exercise you will change the log level of the AdminCourse application while the application is running. The following steps are required:

- Update the log level property of the application and view the log.
- Restart the application and view the log.

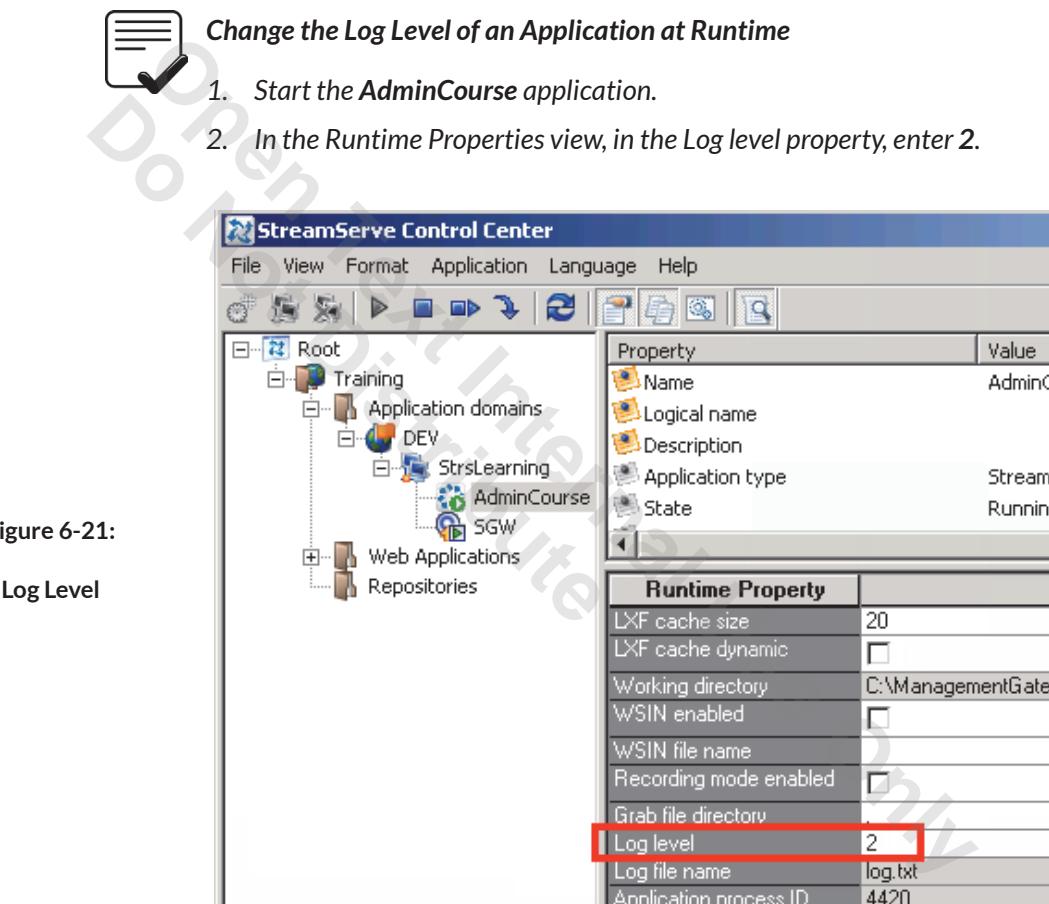


Figure 6-21:

Log Level

3. Send a job to the **AdminCourse** application.
4. View the log messages in the Log view.
5. Repeat steps 2 to 4, testing the log levels 3 and 4.
6. Restart the **AdminCourse** application.

The log level is reset to its original value.

## Exercise 2 – Configure Surveillance to Restart an Application

In this exercise, you will configure surveillance for the AdminCourse application and simulate an error with the application. The following steps are required:

- Enable the Keep preferred state surveillance functionality for the AdminCourse application.
- Simulate an application error and view the results.

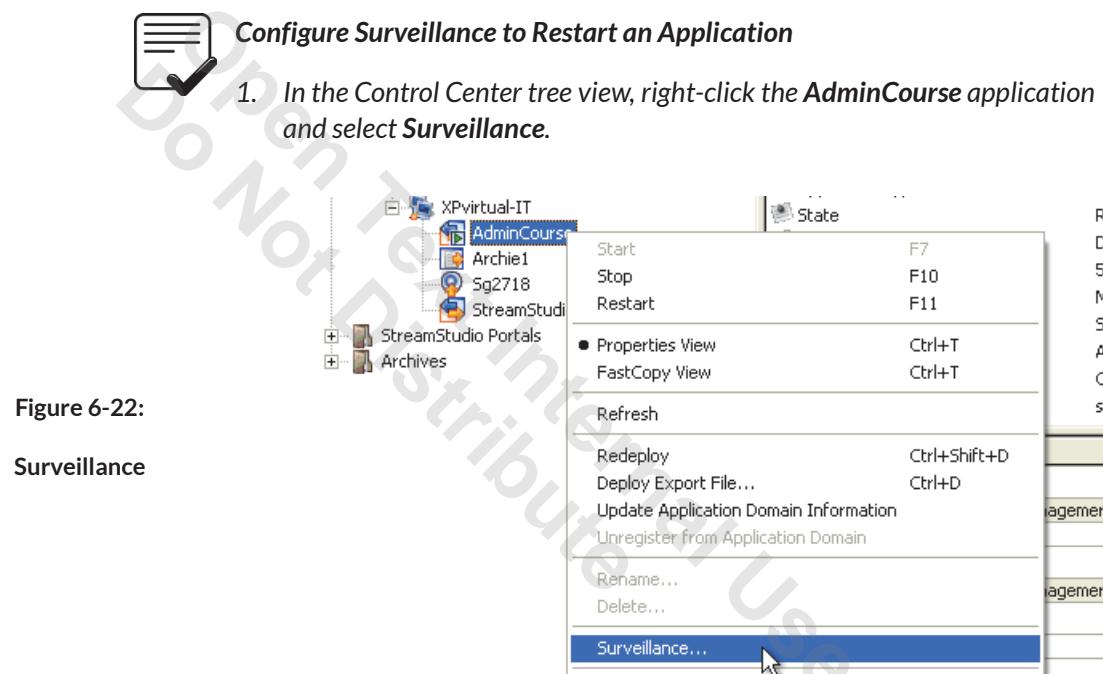


Figure 6-22:

Surveillance

The Application Surveillance dialog box opens.

2. In the Keep preferred state area, select **Enabled**.
3. In the number of attempts field, enter **3**.

4. In the Time between attempts field, enter **15**.

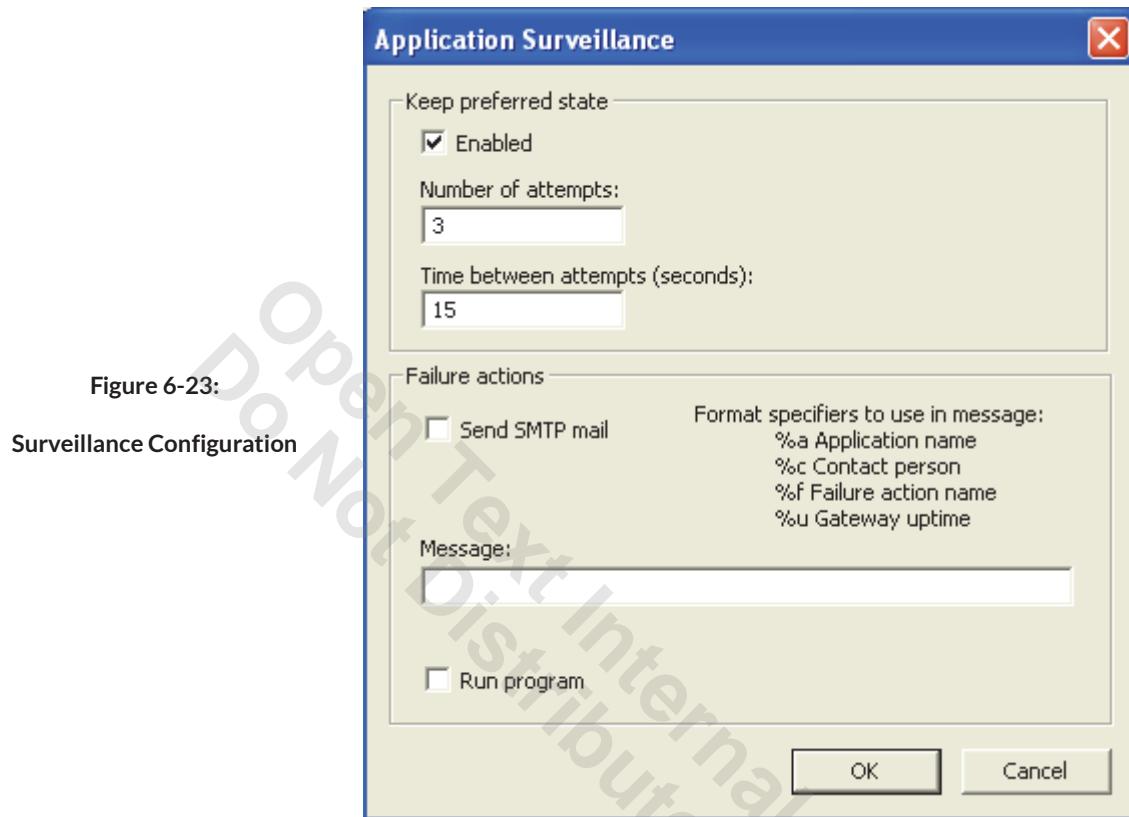


Figure 6-23:  
Surveillance Configuration

5. Click **OK**.

Next you will simulate an application error.

6. Open the **Control Panel** and select **Administrative Tools > Services**.
7. Select the service for the **AdminCourse** application and stop the service.
8. Return to **Control Center**, wait 20 seconds and click the **Refresh** icon.

The application is automatically restarted.

## Exercise 3 – Configure Surveillance to Start a Program

In this exercise you will configure surveillance to run a BAT file that creates a text file when the AdminCourse application goes down. The following steps are required:

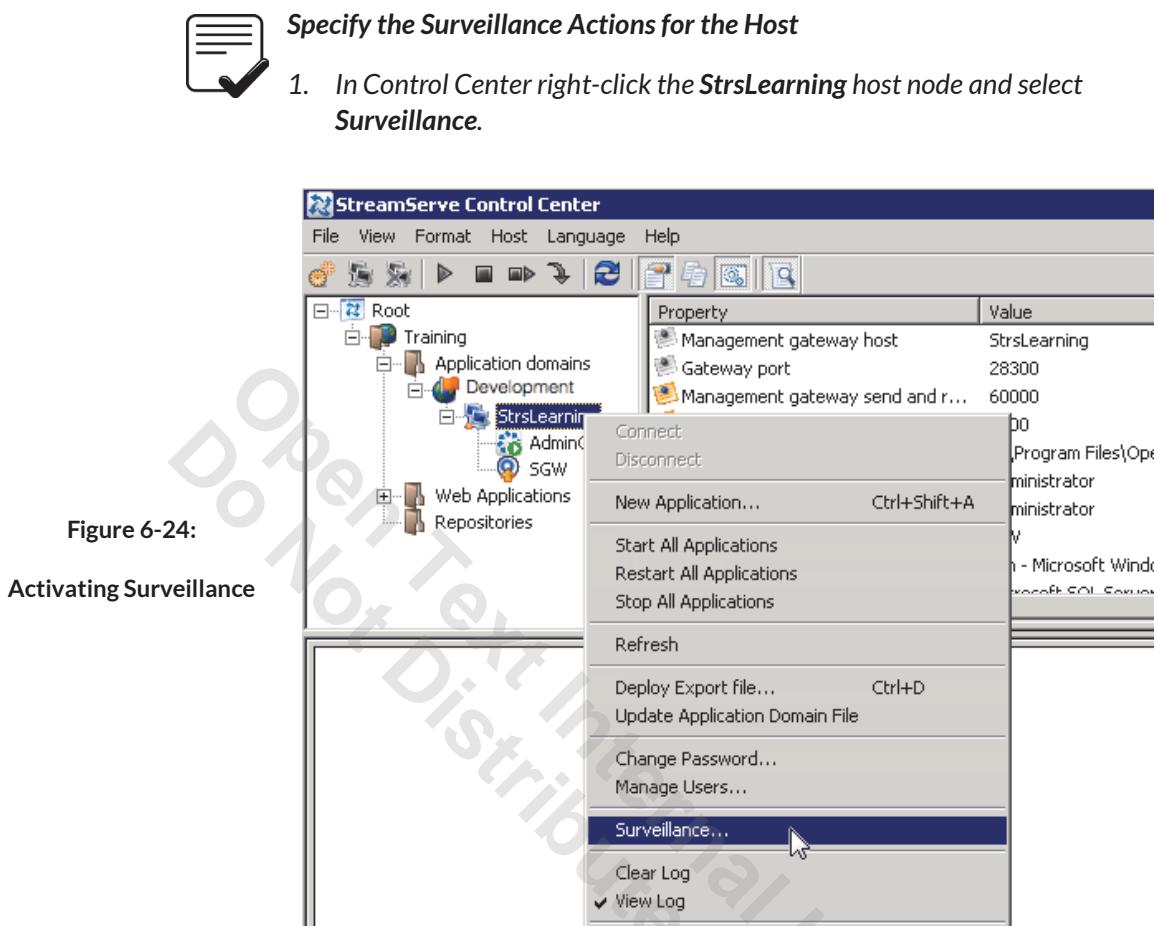
- Create a BAT file that will be run when the application goes down.
- Configure the surveillance actions for your computer.
- Specify which of the failure actions should be taken for the AdminCourse application.
- Simulate an error with the application.



### **Create a BAT File to Be Run When the Service Is Down**

1. Create a text file and save it as **C:\MySurveillance.bat**.
2. Edit the file and enter the following code:

```
echo The application %1 is down  
>c:\MyNotifications.txt
```



**Figure 6-24:**  
Activating Surveillance

The Host Surveillance dialog box opens.

2. In the Host Surveillance window enter the following values:
  - Program: **C:\MySurveillance.bat**
  - Program arguments: **%a**
3. Click **OK**.

Next you will select the failure actions (i.e. send an email) to take for the AdminCourse application. In order to test this functionality, you can reduce the number of restart attempts and time between the restart attempts.



#### Select the Surveillance Actions for the Application

1. Right-click the **AdminCourse** application and select **Surveillance**.
2. In the number of attempts field, enter **1**.
3. In the Time between attempts field, enter **3**.

4. In the Failure action area, select **Run program** (make sure **Send SMTP mail** is not selected).
5. Click **OK**.
6. Open the Control Panel and select **Administrative Tools > Services**.
7. Open the **Properties** dialog box for the **AdminCourse** application and change the Startup type to **Disabled** and click **Apply**.
8. Stop the **AdminCourse** service and click **OK**.

A text file with information that the AdminCourse application is down is created under C:\. You may have to wait up to 60 seconds after the service was stopped until the file is created.

## Exercise 4 – Configure Surveillance to Send an Email

In this exercise you will configure surveillance to send a notification email when the AdminCourse application goes down. The following steps are required:

- Update the contact details in the default application domain.
- Configure the surveillance actions for your computer.
- Specify which of the failure actions should be taken for the AdminCourse application.
- Simulate an error with the application.



### **Update the Contact Details in the Application Domain**

1. In Control Center right-click the **Development** application domain and select **Edit Application Domain**.

The Application Domain editor opens.

2. In the Application Domain Editor window select the **General** tab.
3. In the **Email address to contact person** field, enter **admin@thecompany.com**. and click **OK** to save changes.
4. Click the **Yes** button to update the application domain.
5. Make sure that the **AdminCourse** application is Enabled and started.



### **Specify the Surveillance Actions for the Host**

1. In Control Center right-click the **StrsLearning** host node and select **Surveillance**.

2. In the Host Surveillance window enter the following values:

- Mail server: **localhost**
- Mail domain: **thecompany.com**
- Return address: **noreply@thecompany.com**
- Alternative email address: **mike.jones@thecompany.com**
- Mail time-out: **30**
- Subject : **The application %a is down.**

3. Click **OK**.

In this step, you will select the failure actions (i.e. send an email) to take for the AdminCourse application. In order to test this functionality, you can reduce the number of restart attempts and time between the restart attempts.



#### Select the Surveillance Actions for the Application

1. Right-click the **AdminCourse** application and select **Surveillance**.
2. In the **Number of attempts** field, enter **1**.
3. In the **Time between attempts** field, enter **1**.
4. In the **Failure action** area, select **Send SMTP mail**.
5. Click **OK**.
6. Open the Control Panel and select **Administrative Tools > Services**.
7. Open the **Properties** dialog box for the **AdminCourse** application and change the Startup type to **Disabled** and click **Apply**.
8. Stop the **AdminCourse** service and click **OK**.
9. Open Mozilla Thunderbird and check admin's inbox.

An email with The application AdminCourse is down should arrive into admin's inbox.

## Exercise 5 – Configuring Java Notifications

In the next exercise you are going to use Java Notifications. Typically, a developer creates the classes for the required notifications along with the subscription XML file. You will simply configure StreamServe to use this custom-developed Java notification. We have a simple example listener included in the training image that could be used to subscribe to a defined set of notifications and dump them to a file in XML format.

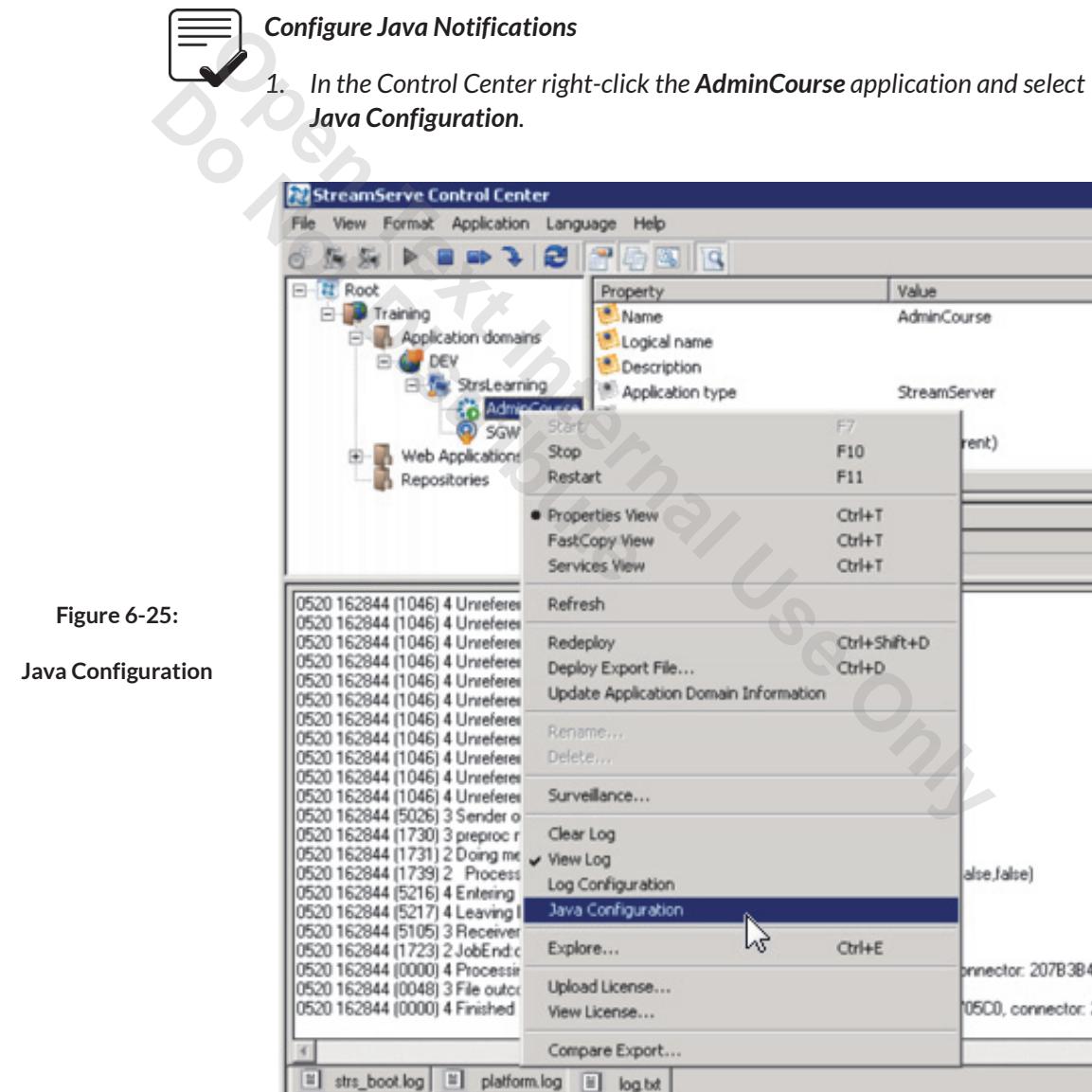


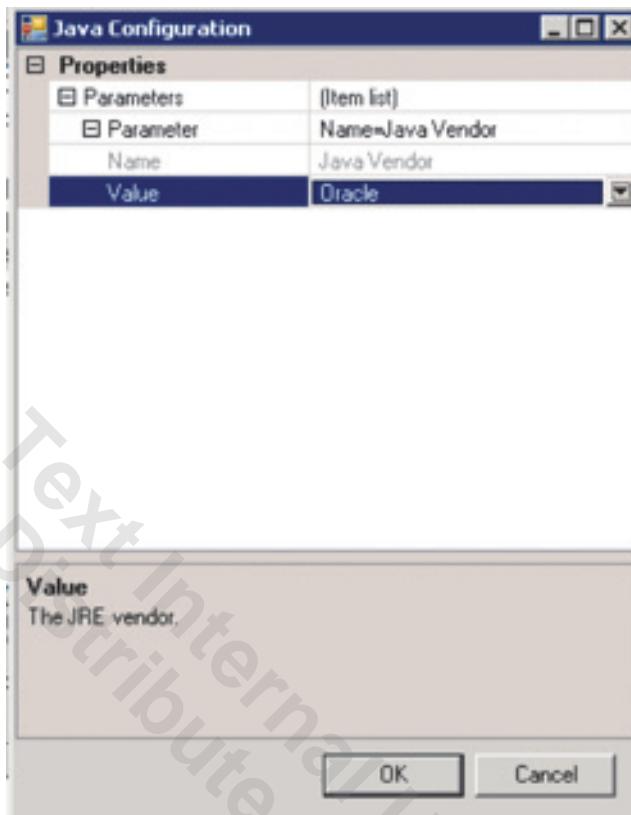
Figure 6-25:  
Java Configuration

The Java Configuration window opens.

2. In the Java Configuration window, select **Oracle** for the Value field and click the **OK** button.

Figure 6-26:

Java Vendor

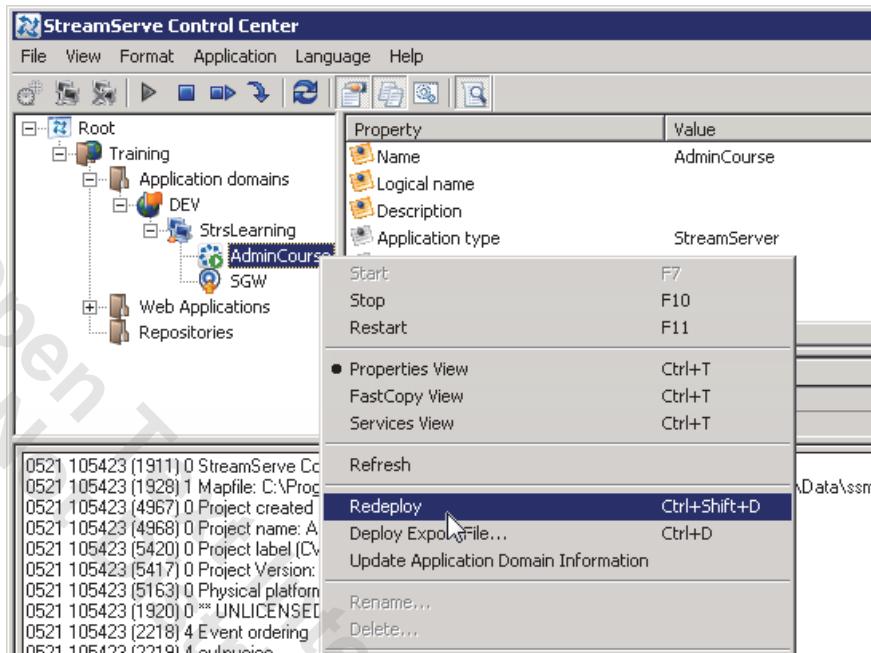


Next you will simply copy the subscription and the notification implementation java files to the application's working directory.

3. Create a folder called **java** under  
**C:\ManagementGateway\5.6.2\root\applications\AdminCourse\data**
4. Copy **jstrscs.jar** from and to the locations indicated below:
  - From: **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles**
  - To: **C:\ManagementGateway\5.6.2\root\applications\AdminCourse\data\java**
5. Copy **notifications.xml** from and to the locations indicated below:
  - From: **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles**
  - To: **C:\ManagementGateway\5.6.2\root\applications\AdminCourse\Development\packages**

6. In the Design Center right-click the **AdminCourse** application and select **Redeploy**.

**Figure 6-27:**  
Application Redeployment



The Application is stopped, redeployed and started. At this point the application will use the provided java notification implementation.

7. To test the notification copy **recordInvoice.txt** from and to the locations indicated below:
  - From: **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles**
  - To: **C:\DEV\IN**



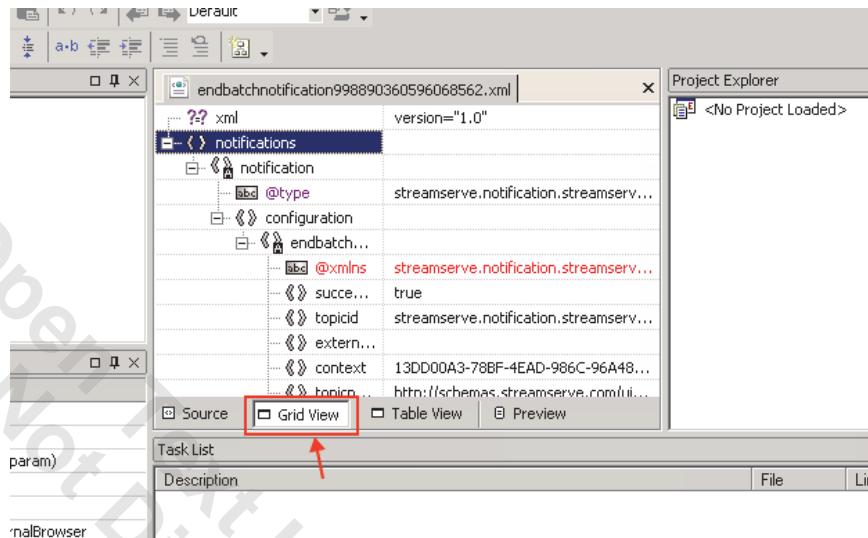
Copy the files 3 or 4 times to get multiple notifications.

After a few seconds you will find one notification XML file in **C:\ManagementGateway\5.6.2\root\applications\AdminCourse\Development\notifications** for each batch processed.

8. Navigate to **C:\ManagementGateway\5.6.2\root\applications\AdminCourse\Development\notifications**, right click any of the notifications (xml files) and select **Edit with XmlPad**.
9. Make sure the Grid View is selected.

You can view the details about this notification. For this particular notification type (endbatch notification) you can see whether the batch ended successfully or not.

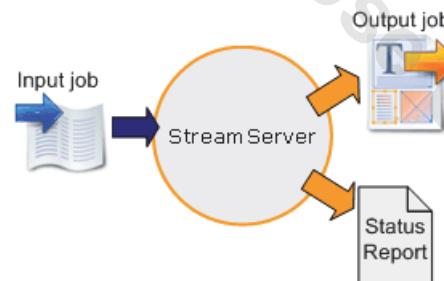
**Figure 6-28:**  
Notification XML



## Status Messenger Overview

### Status Reports

**Figure 6-29:**  
Status Report



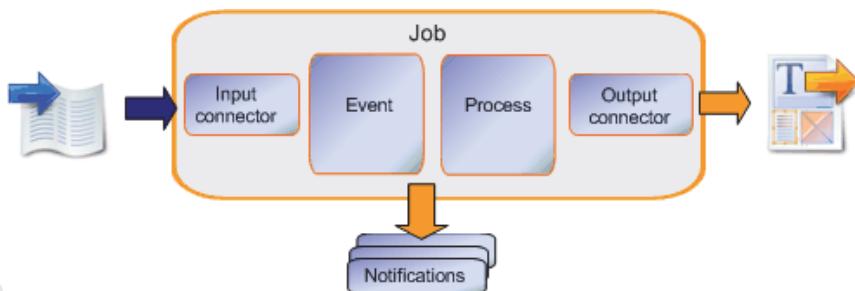
Status Messenger is a StreamServer component that generates status reports for jobs.

For example, if you use StreamServer to create invoices, you can configure Status Messenger to email a status report if the output for a job cannot be delivered.

Each status report created by Status Messenger contains information for one StreamServer job.

## Notifications

Figure 6-30:  
Notifications



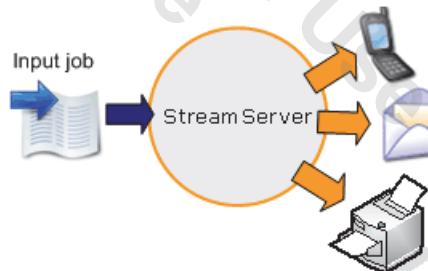
Status Messenger creates status reports based on notifications. Notifications are messages that are created by StreamServer applications during job processing.

For example, notifications are created during jobs when:

- The job is placed in the input queue by the input connector.
- The Events and Processes begin and end.
- The output is delivered by the output connector.

## Why Use Status Messenger?

Figure 6-31:  
Messenger Use

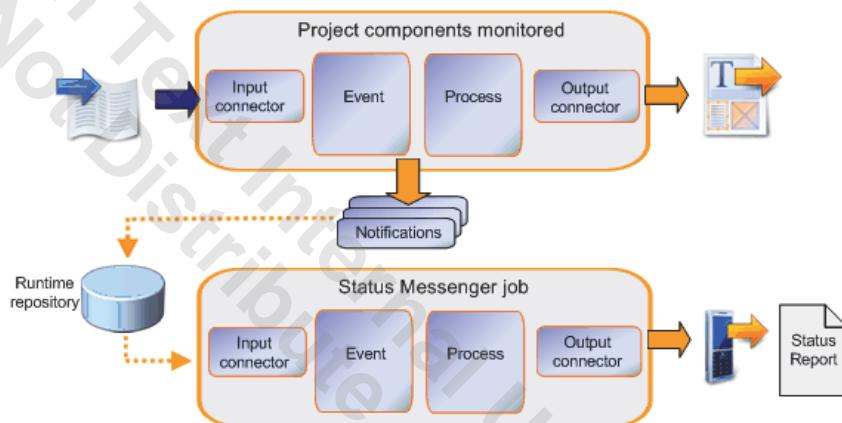


- Monitor Jobs & Multi-Channel Reports** You can use Status Messenger to monitor jobs and generate status reports in any of the StreamServe output channels. For example, you can generate status reports that are sent via email or SMS.
- Notifications Severity Level** You can generate status reports that include notifications with a particular severity level. For example, status reports that only include warning or error messages.
- Reports for Specific Components** You can also generate reports that include notifications for specific Project components. For example, status reports that only include messages related to Events or Processes.

- Non-Job Related Information** Notifications for non-job related log messages can not be included in status reports. For example, when a StreamServer starts or stops.
- Reports for Specific Log Messages** You can generate status reports if specific log messages are generated during jobs. For example, to create a status report if the log message 3153 - Processing Stopped is generated during a job.
- Reports for Specific Jobs** You can also generate status reports if specific log messages are generated during specific jobs. For example, you can specify to generate a status report if the log message 3153 - Processing Stopped is generated during a specific job ID. In this case, the Job ID is an external job ID.

## How It Works

Figure 6-32:  
How It Works



Status Messenger generates notifications for the components of the Project (Processes, Events, etc) you want to monitor.

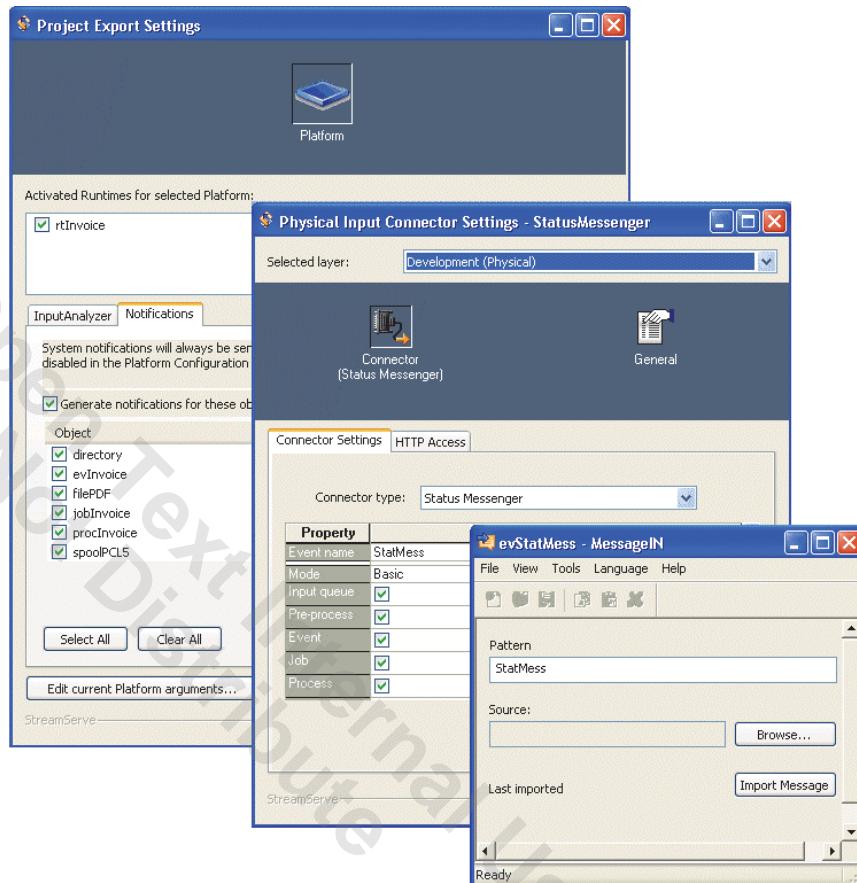
The notifications are saved with information such as the severity level, message text, generation time and the component the notification relates to.

The Status Messenger input connector retrieves the notifications you want to include in the status report.

A Message with a MessageIN Event is used to extract the notification data and create a status report. The status report is sent using a Process and output connector.

## Status Messenger Configuration

Figure 6-33:  
Configuration



You configure Status Messenger in Design Center:

- Project Export settings dialog bog
- Status Messenger input connector

- MessageIN Event

Step	Description
1	Disable the report function for other applications.
2	Enable notifications in the Platform.
3	Specify the Project components to monitor.
4	Configure the Status Messenger input connector.
5	Create a Message with a MessageIN Event.
6	Configure a Process to create the status report.
7	Configure the Runtime configuration.

### Disable Report Function for Other Applications

**Other Applications in the Same Domain** You must make sure that the job status information is not consumed by any other applications that run in the same application domain as the Status Messenger Project.

There are two ways to disable the report function for an application:

- Disable the report function in the `repositorymanager.xml` file of the StreamServer application.
- Configure the custom startup argument `-statusevent` to disable the report function. This is done in Design Center.

You must disable the report function for every StreamServer application in the application domain, except for the StreamServer application that runs the Status Messenger Project.

The `repositorymanager.xml` file for a specific StreamServer application is located in the working directory of the application, for example:

`C:\ManagementGateway\1.0\root\applications\AdminCourse\Development`

To disable the report function in the repositorymanager.xml file of a StreamServe application

1. Open the repositorymanager.xml file.
2. Find the following line:

```
<readystatusevent schedule="T II ** MH ** 1" update="true"  
report="true" />
```

3. Update this line to:

```
<readystatusevent schedule="T II ** MH ** 1" update="true"  
report="false" />
```

**Configuring the -statusevent Startup Argument in the Project**

This is done by configuring the custom startup argument `-statusevent 0` in Projects that are run in the same application domain as the Status Messenger Project.

You configure custom startup arguments in Design Center, in the Configure Platform Export dialog box.

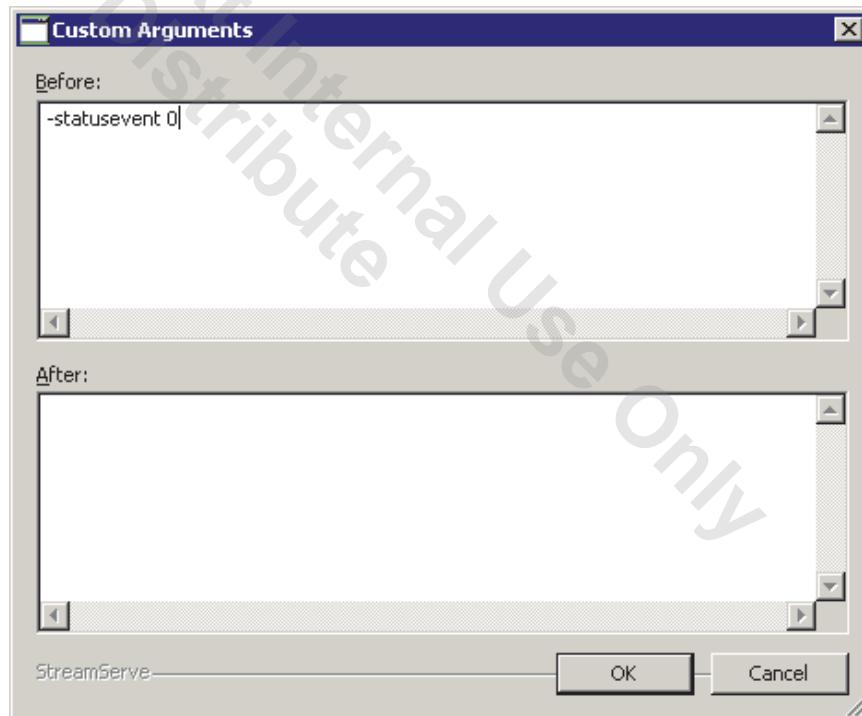


Figure 6-34:  
Platform Export Dialog Box

**Enabling Notifications** To use Status Messenger you must enable notifications in the generic Platform layer.

Enabling notifications affects the performance of the StreamServer.

When you enable the notifications, you must specify when to report job completion.

This determines when the Status Messenger input connector retrieves the notifications that are generated during a job and which notifications are included in the status report.

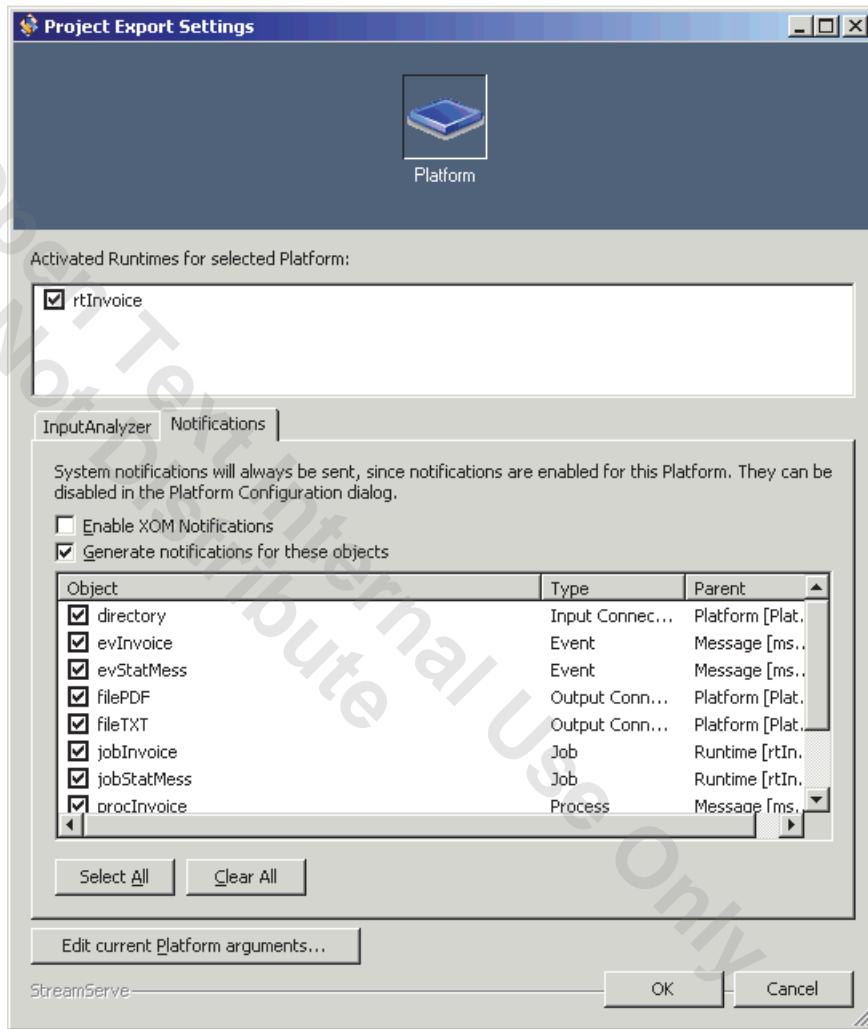
**Settings** The table below describes the report status settings:

Setting	Description
When input is received	Job completion is reported when input is received. Use this option if you only want notifications until the job is placed in the input queue included status report.
When output is queued	Job completion is reported when the job is queued in the output queue. Use this option if you only want notifications until the job is placed in the output queue included in the status report.
When delivered from the output queue	Job completion is reported when the job is delivered from the output queue. Use this option if you want all notifications for the job included in the status report (i.e. until the output connector delivers the job from the output queue).

### Generating Notifications for Project Components

Notifications are generated for the components of the Project (Processes, Events, etc) you want to monitor.

You specify the components to generate notifications for in the Project Export settings in the generic Platform layer.



**Figure 6-35:**  
Project Export Settings

### Retrieving Notifications

The Status Messenger input connector retrieves the notifications to include in the status report.

In the input connector, you specify which notifications you want to retrieve. You can specify:

- The severity level of the notifications (e.g. warning, running).
- The Project components the notifications relate to (e.g Events, Processes).

You also specify the name of the MessageIN Event that extracts the notification data in the Status Messenger input connector.

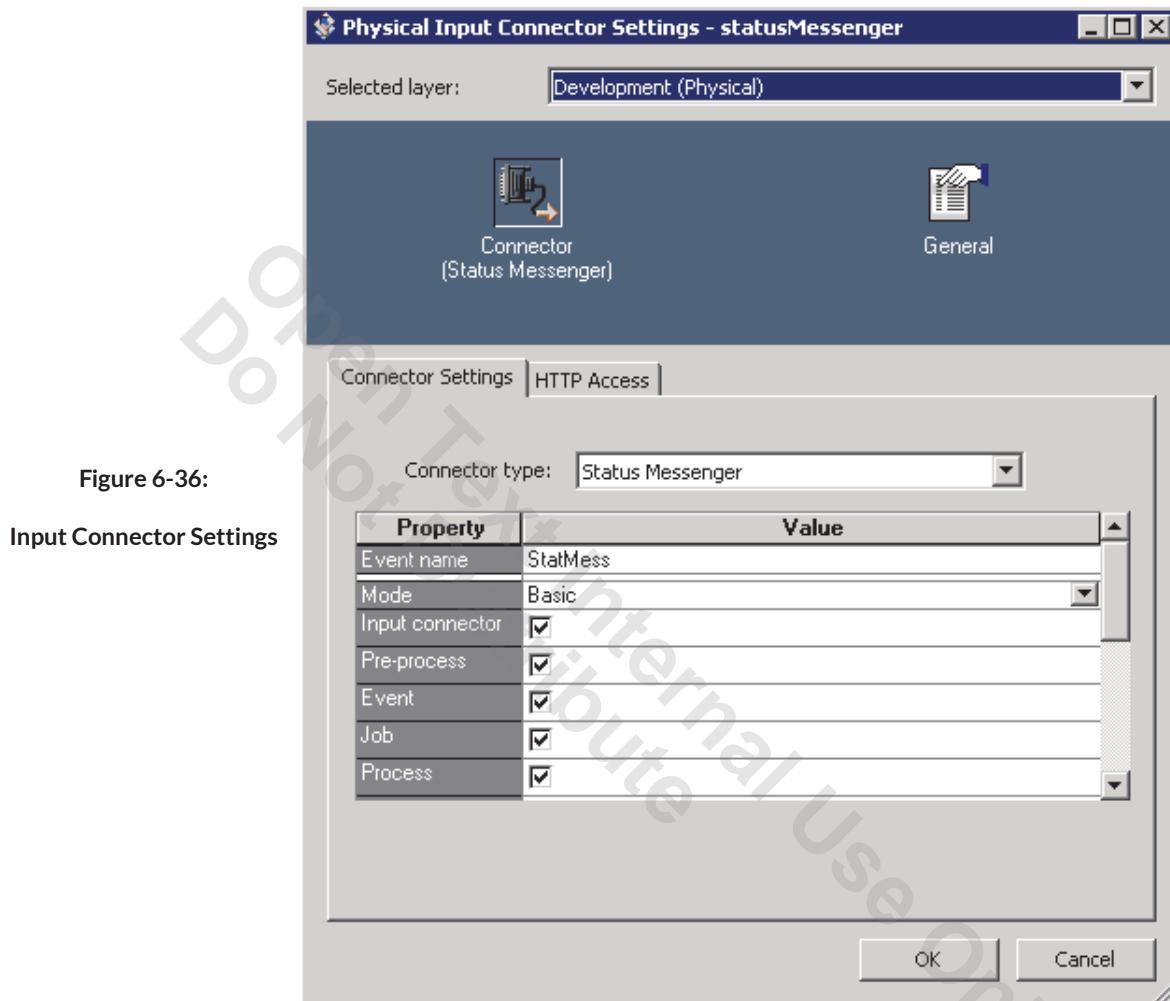


Figure 6-36:

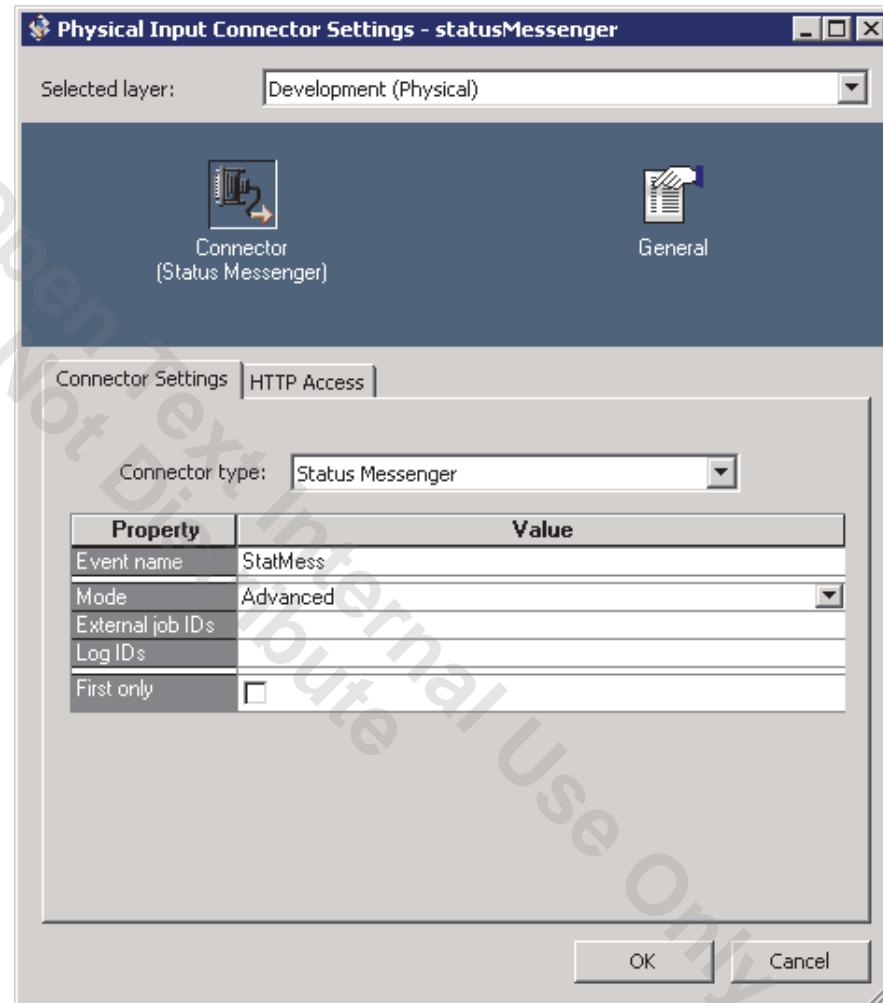
#### Input Connector Settings

**Retrieving  
Notification for  
Specific Log Messages  
and Jobs**

You use the **Advanced** Mode in the Status Messenger input connector to create status reports for specific log messages or jobs.

**Input Connector Settings**

**Figure 6-37:**

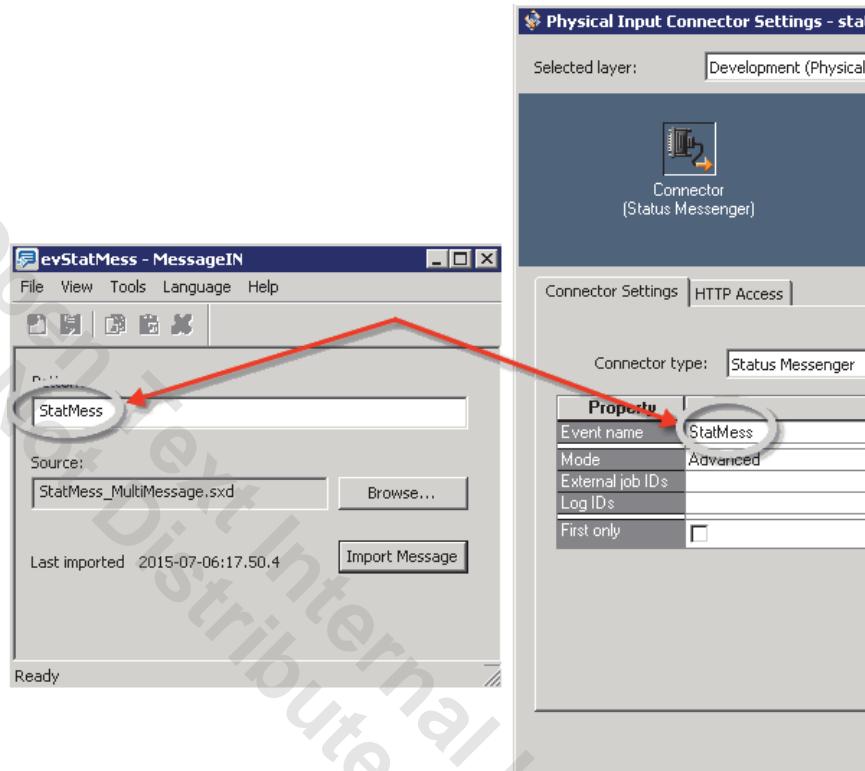


You can find a list of the log messages and log message IDs in the `logmsg.txt` file. This file is located in the StreamServe Prerequisites installation directory:

```
<StreamServe  
installation>\Applications\StreamServer\5.5.0\Common\data
```

## Configuring the Message for the Status Report

**Figure 6-38:**  
Status Report Message Configuration



**Configuring the MessageIN Event** You configure a Message with a MessageIN Event to extract the notification data and create the status report.

To create the MessageIN Event you need a predefined SXD file. This file is called `StatMess_MultiMessage.sxd`

It is located in the Design Center installation directory:

`<StreamServe installation>\Applications\StreamServer\5.5.0\Tools\System\data\sxd`

You must add this SXD file to the Project resource set.

When you create the MessageIN event, the Pattern must match the Event name in the Status Messenger Input Connector.

You must also import the `StatMess_MultiMessage.sxd` file into the MessageIN Event.

**Configuring a Process to Format the Status Report** You can configure any Process to format the status report.

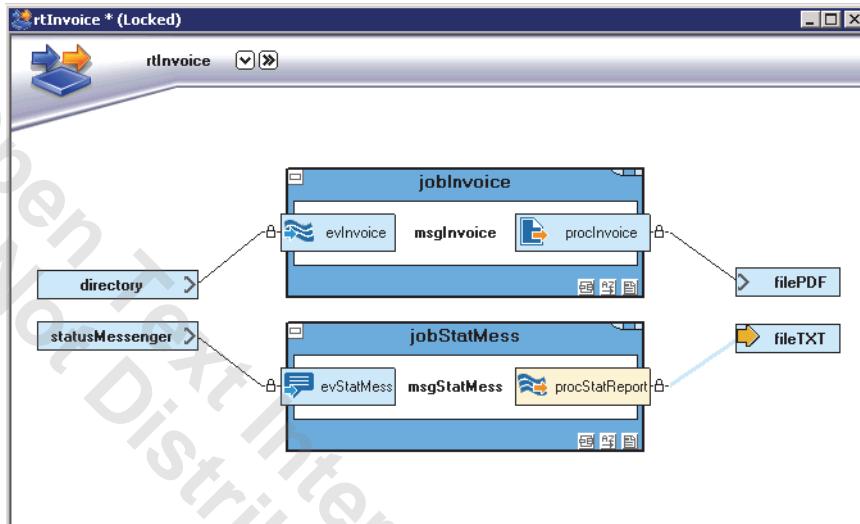
### Configuring the Runtime Configuration

In the Runtime configuration, you create a new Runtime job for the status report.

You connect the Status Messenger input connector to the MessageIN Event.

You connect the Process to an appropriate output connector.

**Figure 6-39:**  
Input Connector Settings



### Creating More Than One Status Report for the Same Job

To create more than one status report for the same job, you configure different Status Messenger input connectors to retrieve to different notifications. Then you create a separate Message for each Status Messenger input connector.

## Exercise 6 - Status Messenger Project Preparation

In the exercises in this module, you will start with a “clean” Project. Before you start with the exercises, you must do the following:

- Create a new Project directory called AdminCourse\_SM.
- Create an export directory.
- Unpack the AdminCourse\_SM package to the AdminCourse\_SM Project directory.
- Rename the Project to AdminCourse\_SM.
- Set demo mode.
- Set the export path.



### Project Preparation

1. In Windows Explorer, browse to C:\STRS\_PROJECTS.
2. Create a new directory, and rename it to AdminCourse\_SM.
3. In Windows Explorer, browse to C:\STRS\_PROJECTS\AdminCourse\_SM.
4. Create a new directory, and rename it to Export.
5. Browse to the C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles\dminCourse\_SM.dcpackage file.
6. Double-click the AdminCourse\_SM.dcpackage file.

The Unpack Project dialog box opens.

7. Browse to, and select, C:\STRS\_PROJECTS\AdminCourse\_SM.
8. Click OK.

The Project Settings dialog box opens.

9. Change Project name to AdminCourse\_SM and click OK.
10. In the Project Settings window, click OK.
11. Select File > Save Project to save the changes made to the Project.
12. In the Project browser, double-click the Platform > Development node.

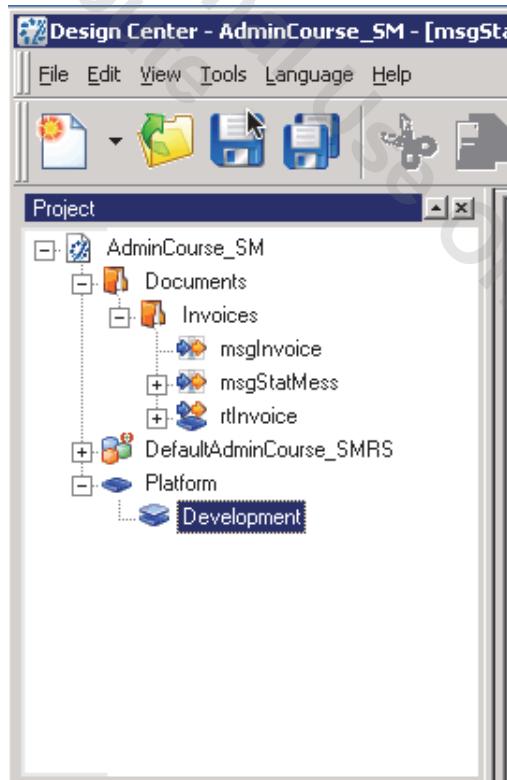
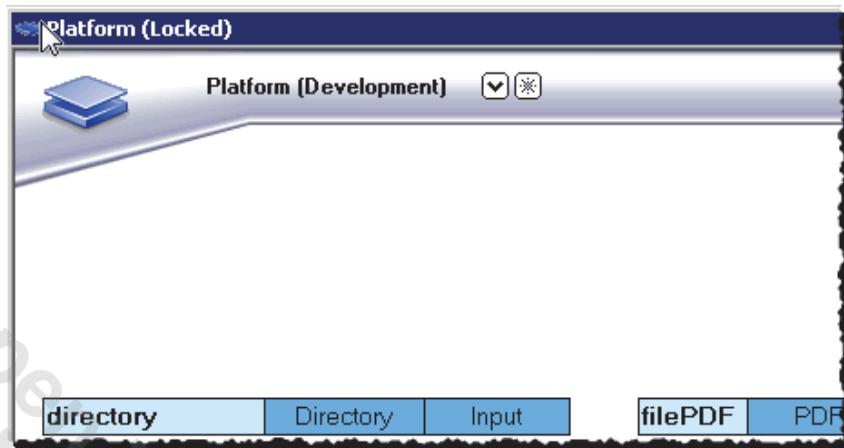


Figure 6-40:

Development Platform  
Node

The Development layer is activated in the Platform view.

Figure 6-41:  
Development Platform View



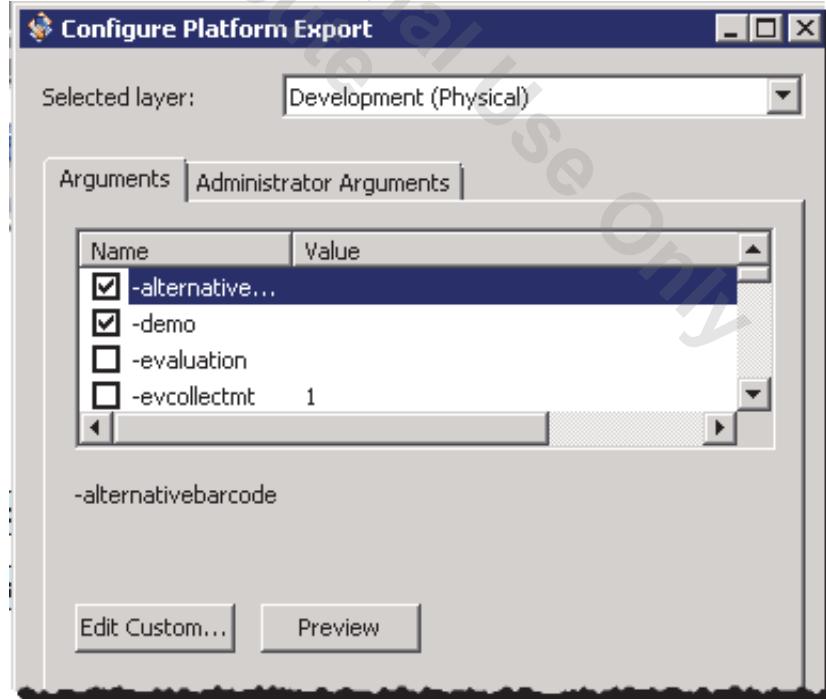
13. Right-click the **Platform** view and select **Configure Export**.

The Configure Platform Export dialog box opens.

14. Select the **Arguments** tab.

15. Select **-demo** and click **OK**.

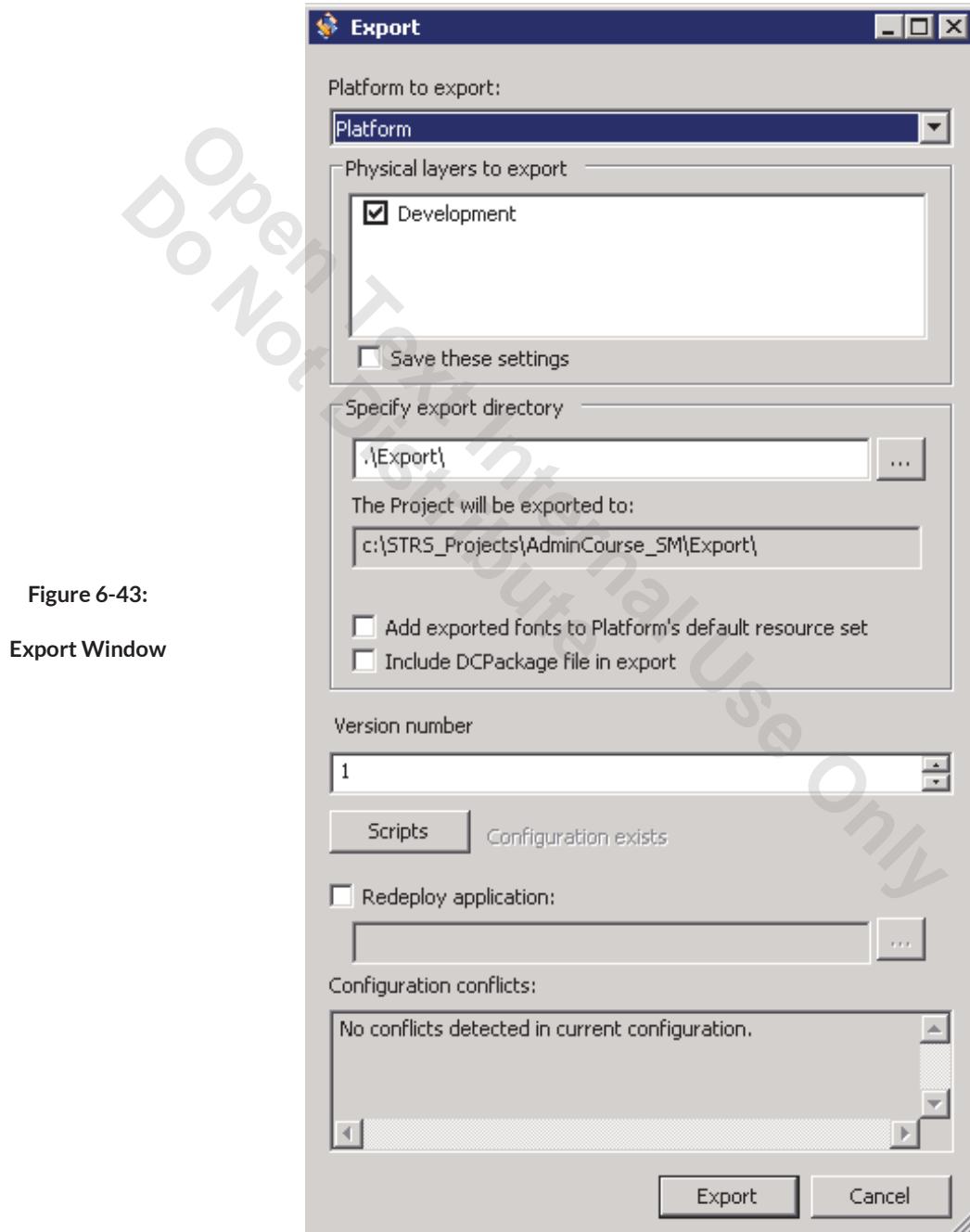
Figure 6-42:  
Setting Demo Mode



16. In Design Center, select **Tools > Export**.

The Export dialog box opens.

17. In *Specify export directory*, browse to and select **C:\STRS\_PROJECTS\AdminCourse\_SM\Export**.
18. Leave the other settings as they are, and click **Export**.



19. In the Export Report window click **Close**.

The Project is exported.

20. Select **File > Save Project** to save the changes made to the Project.

## Exercise 7 – Configure Status Messenger

In this exercise, you will configure Status Messenger to monitor jobs sent to the AdminCourse\_SM StreamServer application. The following steps are required to configure Status Messenger in the AdminCourse\_SM Project:

- Enable notifications in the generic Platform layer and specify when to report job completion.
- Specify which components of the AdminCourse\_SM Project you want to monitor in the Project Export Setting dialog box.
- Create a Status Messenger input connector to retrieve the notifications you want to include in the status report.
- Import the SXD file for the Status Messenger MessageIN Event into the Project resource set.
- Create a Message with a MessageIN Event.
- Configure the MessageIN Event for Status Messenger.



### **Enable Notifications in the Platform**

1. In the Project browser in Design Center, double-click the **Platform** node.

The generic layer is activated in the Platform view.

2. Right-click in the **Platform** view and select **Configure Platform**.

The Configure Platform dialog box opens.

3. Ensure **Use notifications** is selected.

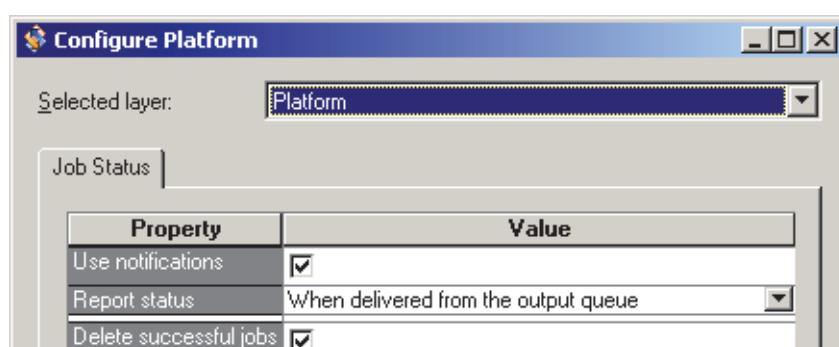


Figure 6-44:

Configure Platform

4. From the Report status drop-down list, select **When delivered from the output queue**.
5. Click **OK**.



#### **Specify the Project Components to Monitor**

1. In the Project browser, right-click the **AdminCourse\_SM** node and select **Project Export Settings**.

The Project Export Settings dialog box opens.

2. Select the **Notifications** tab.
3. Select the **directory**, **evInvoice**, **filePDF**, **jobInvoice** and **procInvoice** objects.

Figure 6-45:  
Selected Objects

Object	Type	Parent
<input checked="" type="checkbox"/> directory	Input Connec...	Platform [Plat...
<input checked="" type="checkbox"/> evInvoice	Event	Message [ms...
<input checked="" type="checkbox"/> filePDF	Output Conn...	Platform [Plat...
<input checked="" type="checkbox"/> jobInvoice	Job	Runtime [rtIn...
<input checked="" type="checkbox"/> procInvoice	Process	Message [ms...

4. Click **OK**.



#### **Create the Status Messenger Input Connector**

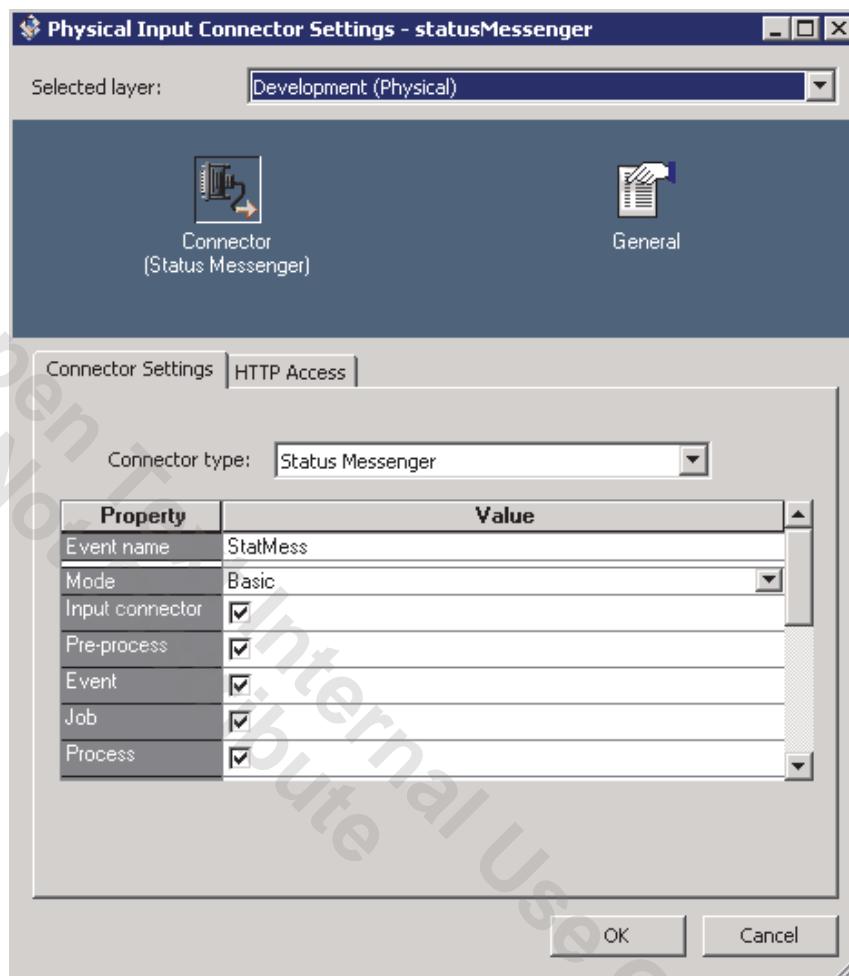
1. In the Platform view, right-click the **Platform** layer and select **New Input Connector**.
2. Name the input connector **statusMessenger**.
3. Double-click the **statusMessenger** input connector.

The Input Connector dialog box opens.

4. From the Queue drop-down list, select the **Input**.
5. From the Selected layer drop-down list, select **Development (Physical)** and click **Yes** to save changes to the layer.

6. From the Connector type drop-down list, select **Status Messenger**.

**Figure 6-46:**  
Configuring the Input  
Connector



7. Click **OK** to select the default settings for the Status Messenger input connector.



#### Import the Status Messenger SXD File to the Resource Set

1. In the Project browser, double-click the **DefaultAdminCourse\_SMRS** node.

The resource set is displayed in main window.

2. In the Resource set view, right-click the **DefaultAdminCourse\_SMRS** node and select **New > Folder**.

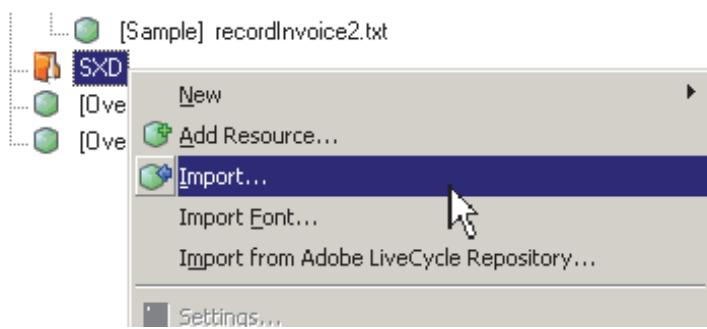
A new folder is added.

3. Rename the new folder to **SXD**.

4. Right-click the **SXD** node and select **Import**.

Figure 6-47:

Import



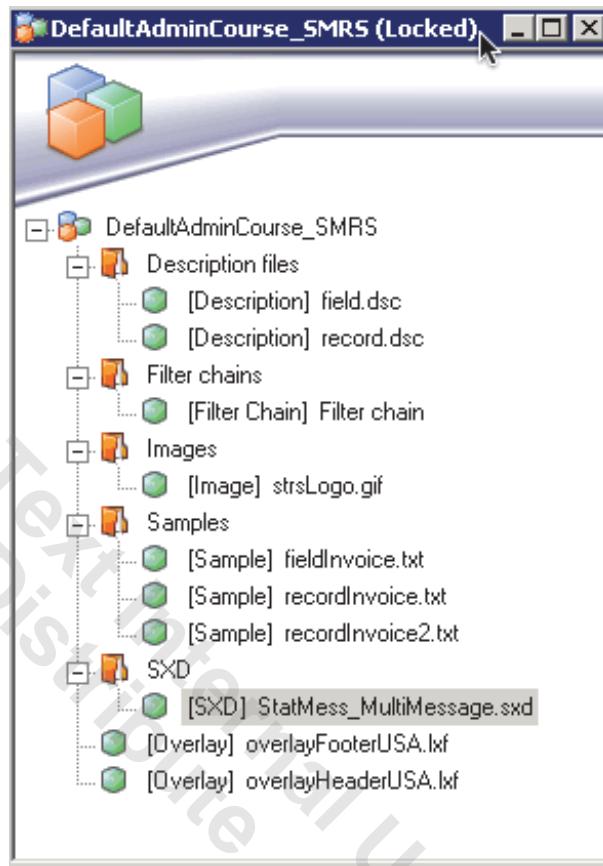
5. Browse to the **StatMess\_MultiMessage.sxd** file in the installation directory: C:\Program Files\OpenText\StreamServe\Applications\Tools\5.6.2\System\data\sxd
6. Double-click **StatMess\_MultiMessage.sxd**.

The Resource Type Settings dialog is displayed.

7. Select **SXD** and click **OK**.

The file is added to the resource set.

Figure 6-48:  
SXD Folder



#### Create a Message with a MessageIN Event

1. In the Project browser, right-click the **Invoices** node and select **New > Message**.

A new Message is added to the Project tree.

2. Name the Message **msgStatMess**.
3. Right-click the **Message** view and select **Add Event > MessageIN**.

A new MessageIN event is added to the Message.

4. Name the MessageIN event **evStatMess**.

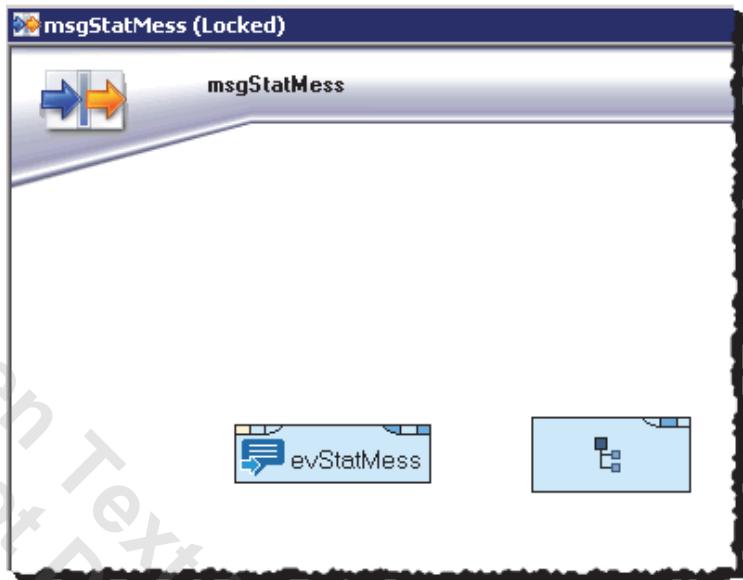


Figure 6-49:  
Event Message



#### Configure the MessageIN Event

1. In msgStatMess, double-click **evStatMess**.

The MessageIN dialog box opens.

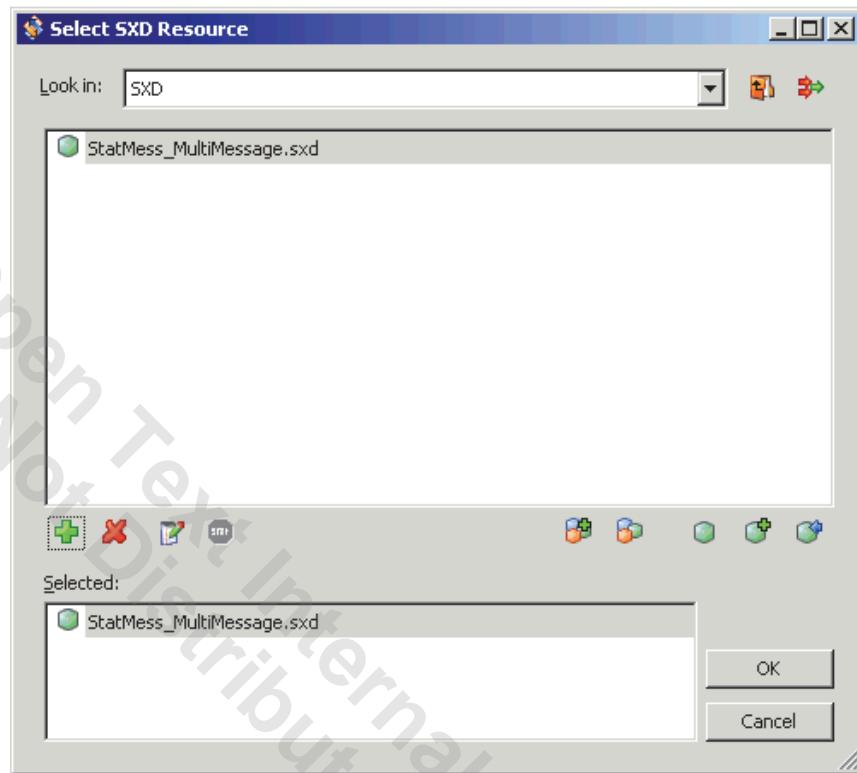
2. In the Pattern field, enter **StatMess**.
3. Click **Browse**.

The Select SXD Resource dialog box opens.

4. Navigate to **DefaultAdminCourseSMRS > SXD**, double-click **StatMess\_MultiMessage.sxd** to select it and click the **OK** button.

Figure 6-50:

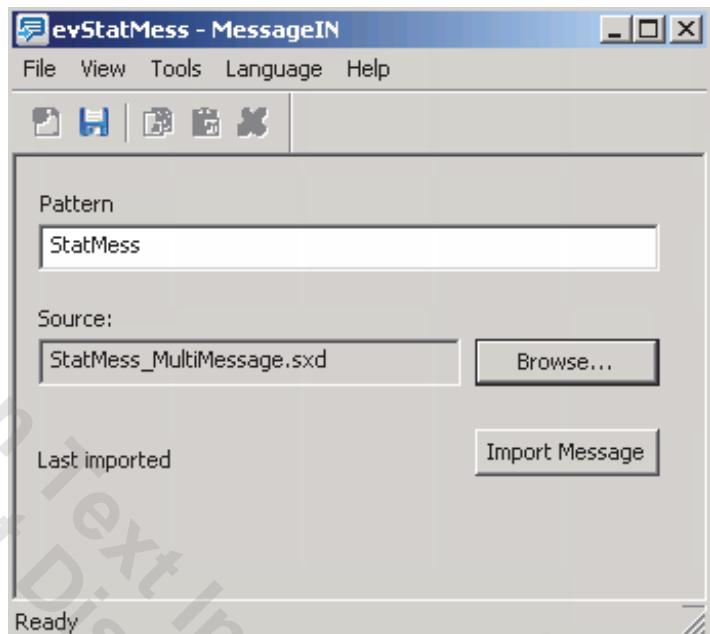
XSD Selection



The MessageIN dialog box is displayed with the StatMess\_MultiMessage.sxd displayed in the Source field.

5. In the MessageIN dialog box, click **Import Message**.

Figure 6-51:  
Import Message



The Message is imported into the MessageIN event

6. Select **File > Exit**.
7. Click **Yes** to save the changes to the Event.

## Exercise 8 – Configure a Process, Output Connector and the Runtime Configuration

In this exercise, you will configure the Project to generate the status report. The following steps are required:

- Add a StreamOUT Process that is used to format the status report.
- Configure an output connector for the status report.
- Configure the Runtime configuration.



### Create a Process to Format the Status Report

You can configure your own StreamOUT process for the status report. To save time we will use a predefined StreamOUT Process.

1. In the Project browser, double-click the **msgStatMess** node.
2. Right-click **msgStatMess** and select **Add Process > Existing Process**.

3. In the Select process window, click the Browse button to navigate to **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles** and select **procStatReport.dcprocess**.
4. Name the Process **procStatReport**.

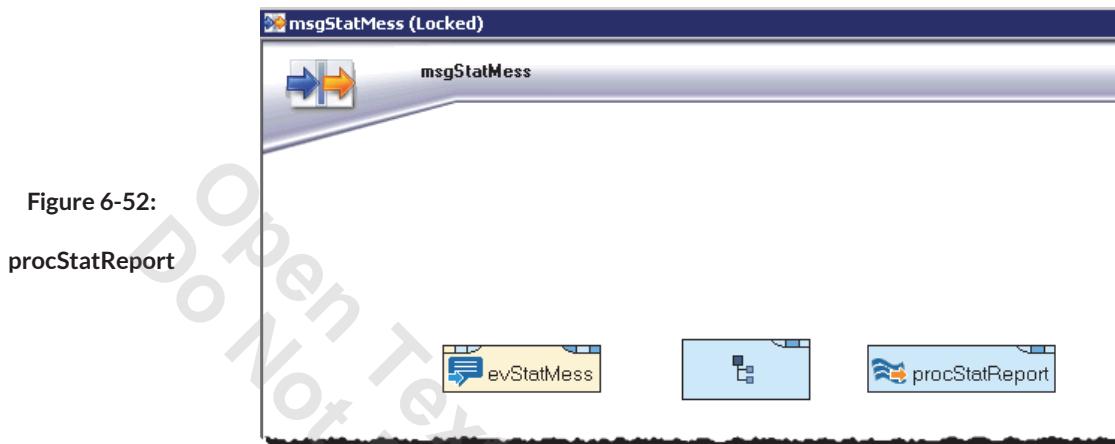


Figure 6-52:

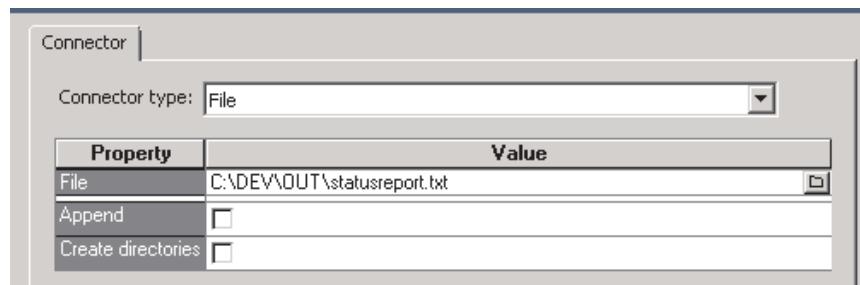
procStatReport

**Configure an Output Connector for the Status Report**

1. In the Project Browser, double-click the **Platform > Development** node.
- The Development layer is activated in the Platform view.
2. Right-click the **Development** layer and select **New Output Connector > Generic**.
  3. Name the output connector **fileTXT**.
  4. Right-click **fileTXT** and select **Settings**.
  5. From the Connector Type drop-down list, select **File**.
  6. In the File field, enter **C:\DEV\OUT\statusreport.txt**.

Figure 6-53:

StreamOUT



7. In the Select layer drop-down select **Platform**.
8. Select **Yes** to save the changes.

9. In the Queue drop-down select **Output**.

10. Click **OK**.



### Configure the Runtime Configuration

1. In the Project browser, double-click the **rtInvoice** node.

The **rtInvoice** is activated in the Runtime configuration view.

2. Right-click **rtInvoice** and select **New Job**.

3. Name the Runtime job **jobStatMess**

4. Right-click **jobStatMess** and select **Add Message**.

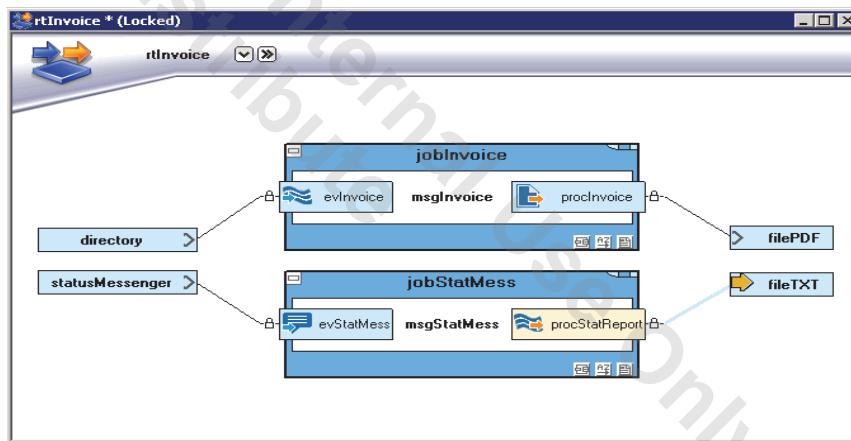
The Select Message dialog box opens.

5. Select **msgStatMess** and click **OK**.

6. Link the **statusMessenger** input connector to the **evStatMess** Event.

7. Link the **procStatReport** Process to the **fileTXT** output connector.

Figure 6-54:  
Linking Objects



## Exercise 9 – Test Your Development

In this exercise, you will test the development of Status Messenger. The following steps are required:

- Export the Project, send a job to the StreamServer application and view the output.
- Simulate an error in the Project.
- Re-export the Project, send a job to the StreamServe application and view the output in statusreport.txt.



### Export, Deploy and Run

1. Export the Project.
2. In Control Center, deploy the export file **AdminCourse\_SM-Platform.export** to a new StreamServer application that you call for example **AdminCourse\_SM**.
3. Start the **AdminCourse\_SM** application.
4. In Design Center, double-click the **DefaultAdminCourse\_SMRS** node.

The resource set is activated in the Resource set view.

5. In the Samples folder, right-click **recordInvoice.txt** and select **Extract To File**.

A file browser opens.

6. Browse to **C:\DEV\IN** and select **Save**.
7. In Windows Explorer, browse to **C:\DEV\OUT** and view the result.
8. In the Project browser in Design Center, double-click the **Development** node.

The Development layer is activated in the Platform view.

9. Right-click **FilePDF** and select **Settings**.
10. In the **File** field, change the path to **C:\DEV\OUT1\Invoice.pdf** and click **OK**.
11. Export the Project.
12. In Control Center, redeploy the Project.
13. In Design Center, double-click the **DefaultAdminCourse\_SMRS** node.

The resource set is activated in the Resource set view.

14. In the Samples folder, right-click **recordInvoice.txt** and select **Extract To File**.

A file browser opens.

15. Browse to **C:\DEV\IN** and select **Save**.
16. In Windows Explorer, browse to **C:\DEV\OUT**.

After a few seconds the statusreport.txt file is generated.

17. View the results in the file called **statusreport.txt**.

**Figure 6-55:**  
**statusreport.txt**

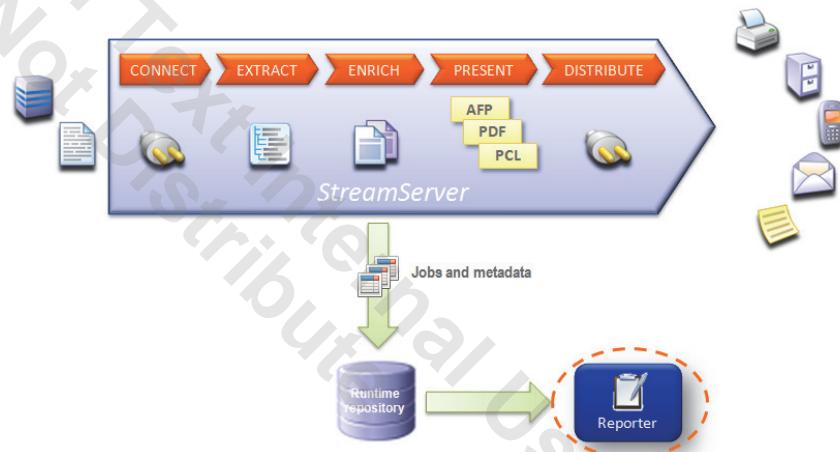
```

statusreport - Notepad
File Edit Format View Help
STATUS REPORT
Job ID: 167
ServerID: wgbg3k14\data\data
Component name: filePDF
Component type: output connector
Message details: File outconnector: unable to append/copy/move from temporary file to C:
Status: fail
Time: Mon Jan 15 16:01:00 2007

```

## Reporter Overview

**Figure 6-56:**  
**Reporter Overview**



Reporter is used for monitoring and managing all jobs processed by StreamServer applications.

You use the Reporter to:

- View job status.
- Resend failed and successful jobs from the queues.
- Delete jobs from the queues.
- View other job information, such as, the documents generated by the job and the customers associated with the jobs.

With Reporter, IT departments can:

- Provide greater visibility and control of the job environment.
- Decrease production time and testing.
- Reduce costs through enhanced workflow processes.
- Improve IT staff productivity through a more flexible design environment.

## StreamStudio Application



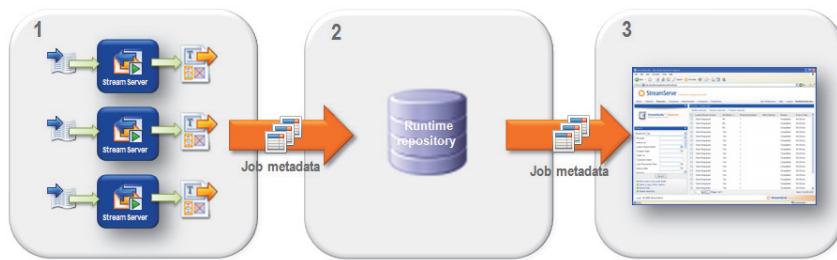
Figure 6-57:

Reporter is a StreamStudio application. You access Reporter via the StreamStudio web portal.

To run Reporter, you need to install and configure the StreamStudio web applications. For more information see the online help.

## Storing Job Metadata

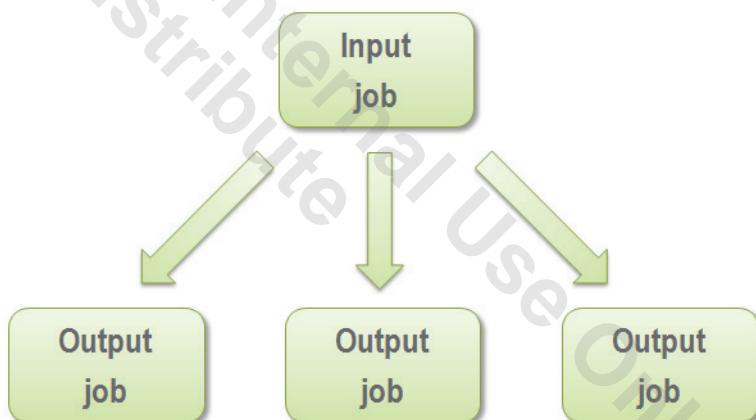
Figure 6-58:  
Store Job Metadata



1. When a StreamServer applications processes jobs, jobs and job metadata are generated.
2. The jobs and job metadata are stored in the runtime repository.
3. From Reporter, you use the job metadata to search for, view, reprocess and delete jobs.

## Input and Output Jobs

Figure 6-59:  
Input Output Jobs

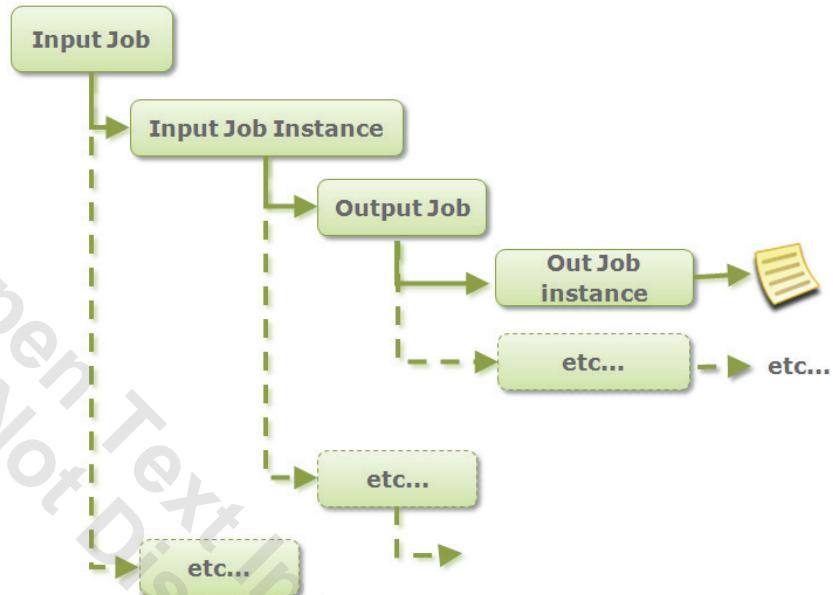


**Input Job** An input job is one stream of data that comes into the input connector for processing. For example, a file received via a directory scan connector. The input job is created when the file is stored in the input queue.

**Output Job** The output job is one output entity that the StreamServer has created to deliver to a destination. The output job is created when it is stored in the output queue.

## Input & Output Job Instances

Figure 6-60:  
Input & Output Job Instances



Every successful input file results in at least:

- One input job.
- One input job instance.
  - If the input job is resent, several input job instances are created.
- One output job per Process and per Event.
- One output job instance per document.
  - If the output job is resent several output jobs are created.

When you search in Reporter, the data you search for is retrieved from a node in the input job/output job tree.

- Job no is retrieved from the input job.
- The document type metadata is retrieved from the output job.

## Configuring Queues to Store Job Metadata

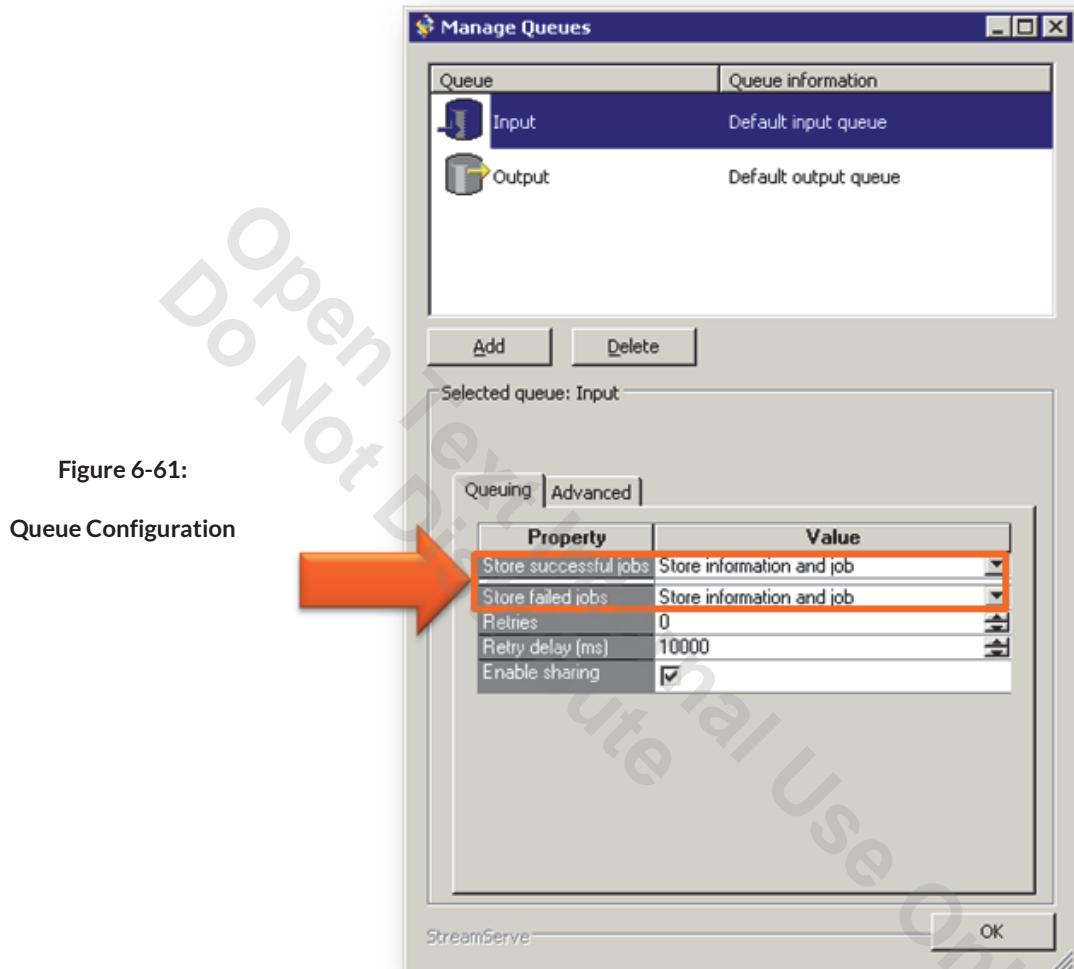


Figure 6-61:

### Queue Configuration

To use Reporter to access jobs, you must configure the input and output queues to store job information (i.e. jobs and job metadata). This is done in Design Center.

To configure the queues to store job metadata

1. In Design Center, in the Platform configuration view, right-click and select Manage queues. The Manage Queues dialog is opened.
2. Select Store information and job.

It is also possible to use the option **Store information only**, however if you use this option, you cannot use Reporter to resend or reroute jobs.

**Examples – Stored Job** The job metadata listed below can be stored when you run a job:

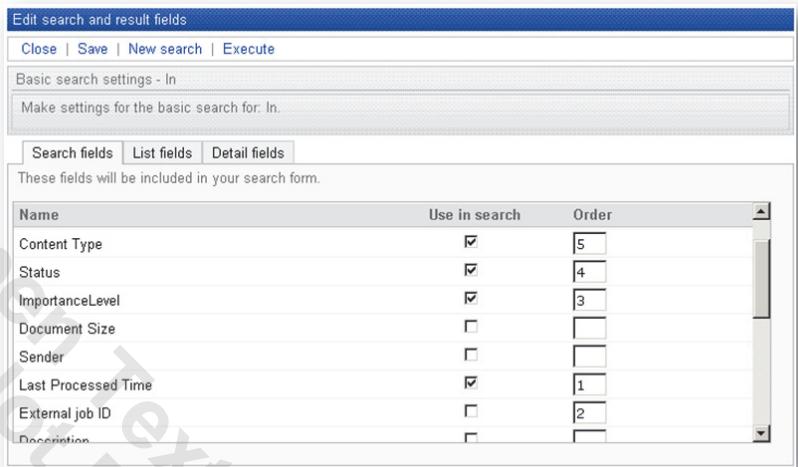
**Metadata**

- Content type
- Creation date
- Job description
- Document size
- Document type
- Error code
- Expire date
- External job ID
- Initiation time
- Job no
- Latest queue
- Latest queue event
- Last job error message
- Last time job processed
- Output job queue
- Processing state (temporary state used during processing)
- Job receiver
- Job sender
- Job status
- Number of times a job has been processed
- Times to attempt resend a job

## Accessing Jobs Using Reporter

### Preparing Searches

**Figure 6-62:**  
Preparing Searches



Before you can search for jobs in Reporter, you must prepare searches. This is done by defining:

- Which metadata to use as search criteria.
- Which metadata to display in the search results.

You must do this for both input and output jobs.

To prepare a search:

1. In the Search area, select In or Out in the Search for field.
2. Click Edit search and results fields.
3. On the Search fields tab, select metadata to use in the search.
4. On the List fields tab, select metadata to display in the search results.
5. On the Detail fields tab, select metadata to display as job details.
6. Click Save and click Close.

You can also use saved searches and quick lists to search for jobs. For more information see the Reporter online help.

## Searching for Jobs

Figure 6-63:

Job Search

The screenshot shows a 'Search' dialog box with the following fields:

- Search for: In
- Creation Date: (calendar icon)
- Job No.
- Initiation Time
- Processing State
- Last Processed Time
- Archived
- External job ID
- Times Accessed

A 'Search' button is at the bottom right.

When you search for jobs you use job metadata as search criteria.

You search for input and output jobs separately.

When you search, you can:

- Combine several search criteria.
- Use dates, variables, wildcards, intervals, etc.

## Viewing Search Results

Figure 6-64:

Preparing Searches

Results - Input jobs					
	Job No.	Initiation Time	ImportanceLevel	Status	Error Code
<input type="checkbox"/>	49	2010-09-07 07:42:53.567		Completed	No Errors
<input type="checkbox"/>	48	2010-09-07 07:42:53.543		Completed	No Errors
<input type="checkbox"/>	45	2010-09-07 07:38:21.493		Completed	No Errors
<input type="checkbox"/>	39	2010-09-07 07:28:08.680		Aborted	Processed With Errors
<input type="checkbox"/>	36	2010-08-30 11:54:07.970		Aborted	Processed With Errors
<input type="checkbox"/>	32	2010-08-30 11:53:36.330		Aborted	Processed With Errors
<input type="checkbox"/>	28	2010-08-30 11:52:02.517		Aborted	Processed With Errors

The search results are displayed as a list of jobs in the results area. You can sort the list by clicking any of the column headers. The search results area displays either input jobs or output, depending on what you searched for.

If a job has been resent, only information about the latest job processing is included in the results.

From the results view, you can:

- Resend jobs.
- Delete input and output jobs.
- Preview the document that the output job generated.
- Select an input/output job and view the job details.

To view the job details and job history, click the job to open the Input job/Output job details view.

### Input Job Details View

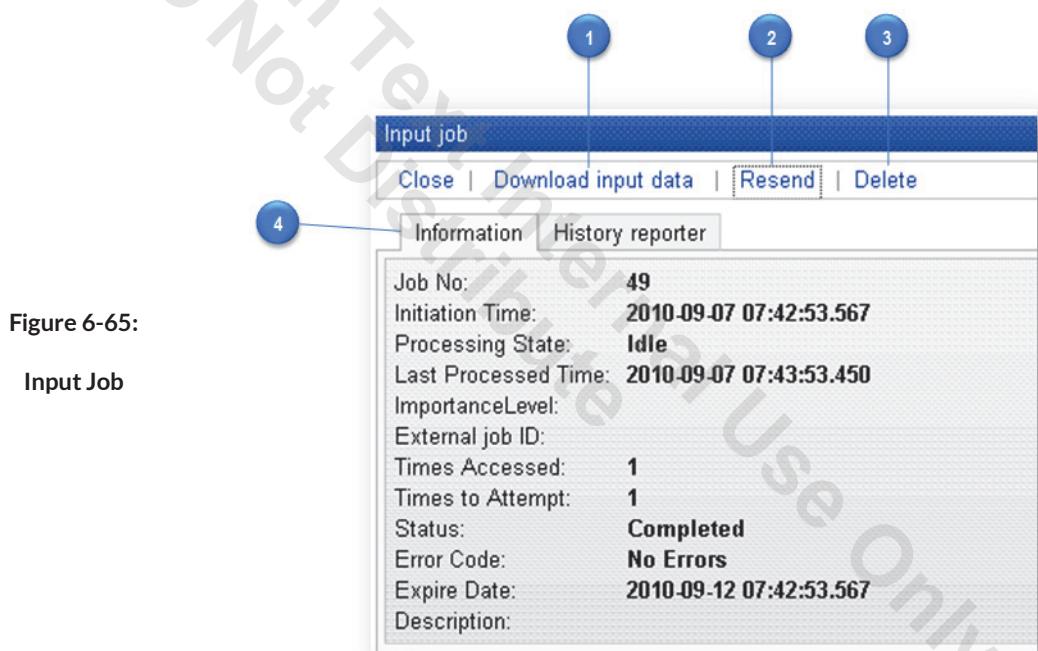


Figure 6-65:  
Input Job

Setting	Description
(1) Download input data	Displays the input data used during processing. Since the type of this data is unknown, you must always specify how the file should be viewed or where to save it. The default file name is JobData.bin.
(2) Resend	Resends the job, and reroute print and email jobs.

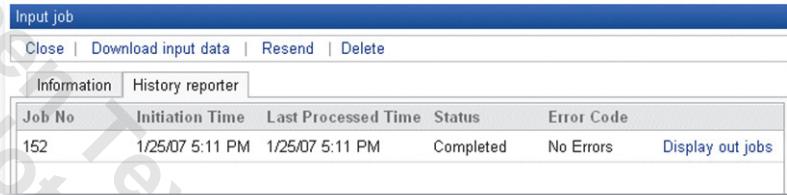
Setting	Description
(3) Delete	Marks the job for deletion.
(4) Information tab	Displays the metadata you selected on the Detail fields tab when you defined the search.

To view information about the job history, click the **History reporter** tab.

**Input Job Details – History Reporter Tab**

**Figure 6-66:**

**History Tab**



The History reporter tab displays one row for each time the input job is processed (i.e. each input job instance).

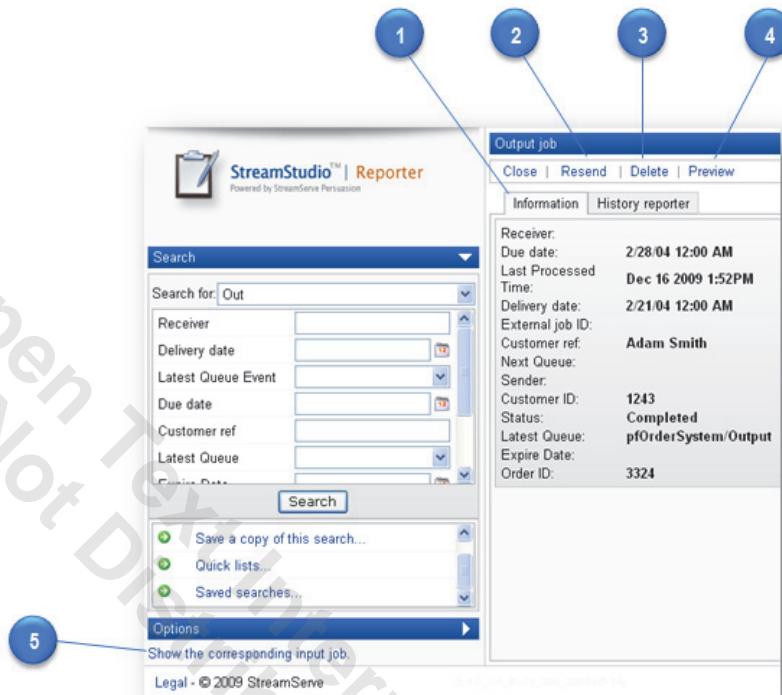
You can also access the related output jobs from the History reporter tab.

## Output Job Details

### View

Figure 6-67:

History Tab



Setting	Description
(1) Information tab	Displays the metadata you selected on the Detail fields tab when you defined the search. For more information about the metadata, see the Reporter online help.
(2) Resend	Resends the job and can also be used to reroute print and email jobs.
(3) Delete	Marks the job for deletion.
(4) Preview	Displays the document generated and stored by the output job.
(5) Show corresponding input job	Displays the related input jobs.

To view information about the job history, click the **History reporter** tab.

#### History Reporter Tab

Figure 6-68:

History Tab

Output job				
Close   Resend   Delete   Preview				
Information History reporter				
Job No	Initiation Time	Last Processed Time	Status	Error Code
108	1/21/07 12:09 PM	1/21/07 12:09 PM	Completed	No Errors

The History reporter tab displays one row for each time the job is processed.

#### Resending Jobs

Figure 6-69:

History Tab



You use Reporter to resend failed or successful jobs from the queue. A resent job is processed by the StreamServer that originally processed the job.

- How It Works**
1. When a StreamServer application processes jobs, jobs and job metadata are generated.
  2. The jobs and job metadata are stored in the runtime repository.
  3. A resend command is sent by the Reporter user.
  4. The selected job is retrieved from the runtime repository and reprocessed by the StreamServer application.
  5. New input and output job instances are created when jobs are reprocessed, and stored in the runtime repository.

## Configuring the Queues

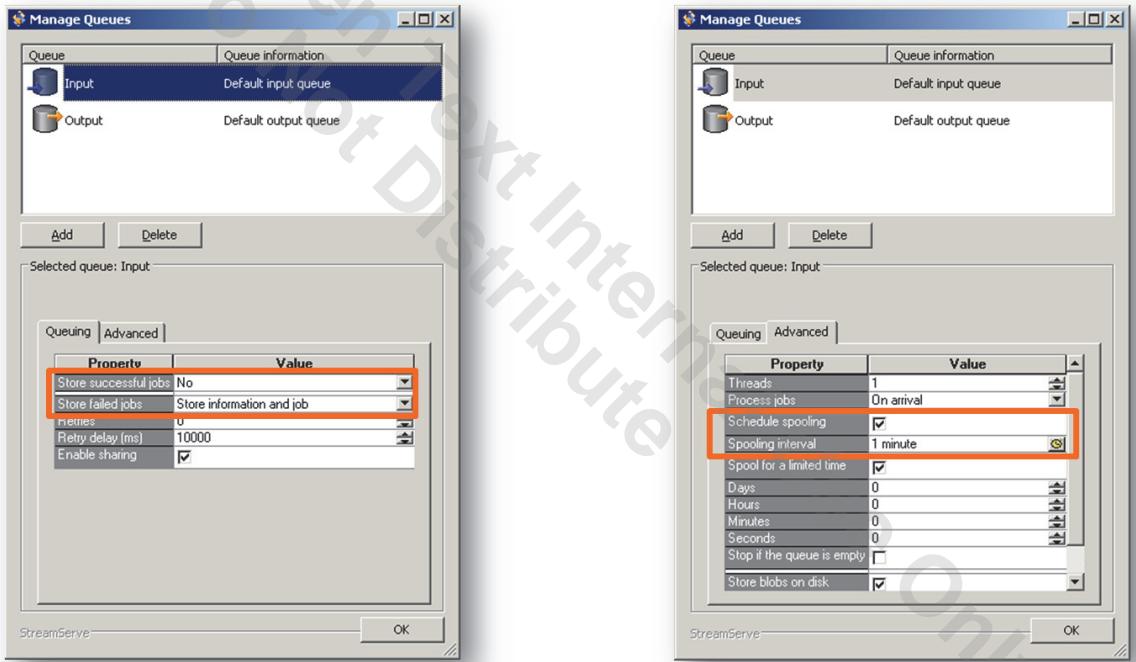


Figure 6-70: Queue Configuration

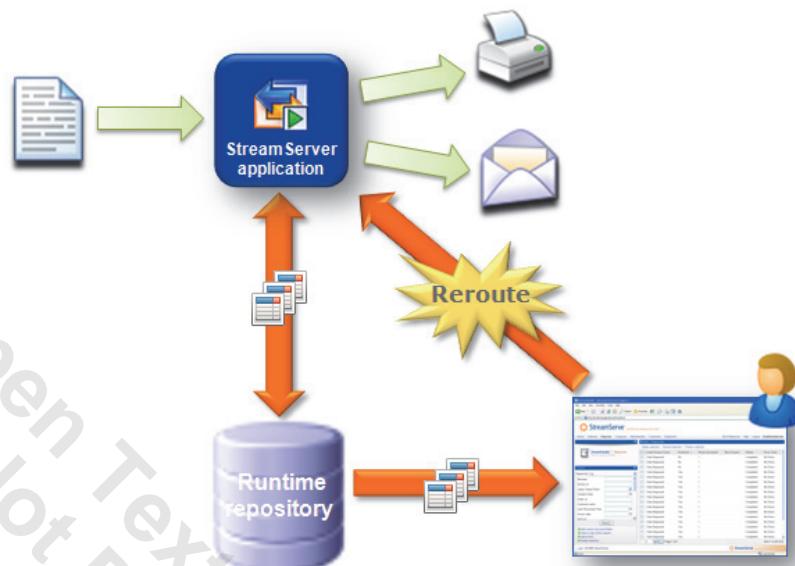
You must configure the queues to enable jobs to be resent.

To configure the input and output queues so jobs can be resent:

1. In Design Center, in the Platform configuration view, right-click and select Manage queues. The Manage Queues dialog is opened.
2. On the Queuing tab, select Store information and job for successful and failed jobs.
3. On the Advanced tab, select Schedule spooling.

## Rerouting the Output

Figure 6-71:  
Job Rerouting



You can reroute the output when you resend print jobs and email jobs. For example, if a print job failed due to a printer error, you can resend the job to another printer.

Fax jobs can only be resent to the original fax number.

If you resend print or email jobs, the parameters used the last time the job was processed are displayed in Reporter. You can reroute jobs by changing the values of these parameters. If you leave a field empty, the value from the original processing is used.

To reroute several jobs simultaneously, you must select jobs of the same type, for example, print jobs.

## Exercise 1 – Use Reporter

In this exercises you will:

- Set up a search that returns a list of all processed input jobs
- Examine the details for a specific job
- View data in input files
- Reprocess a job



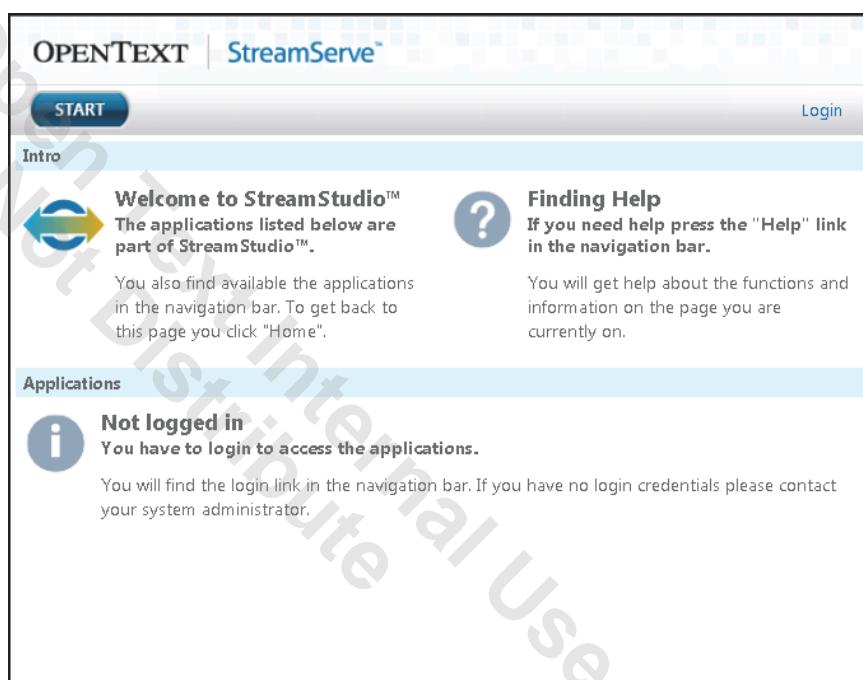
### Create a Process to Format the Status Report

1. From the Windows Task bar, click the Mozilla Firefox button.

The Mozilla Firefox web browser opens.

2. In the address text box, enter <http://localhost:8080/StreamStudio/Portal/start>.

The StreamStudio portal start window opens.

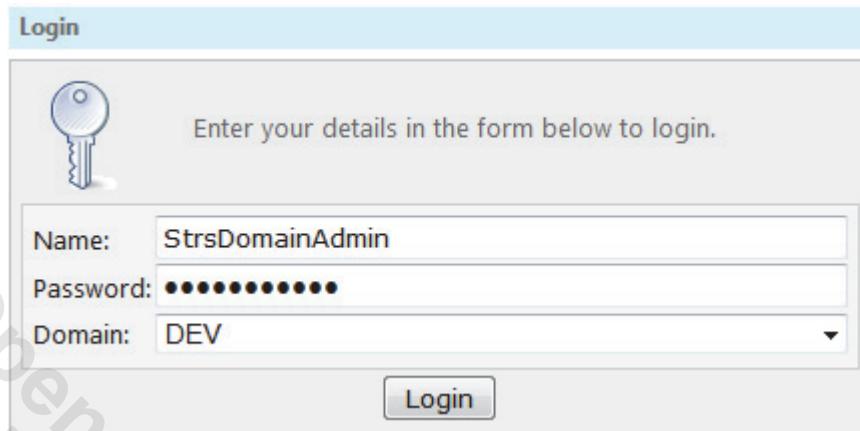


**Figure 6-72:**  
StreamStudio Window

3. Click **Login**.

The Login window opens.

Figure 6-73:  
StreamStudio Login  
Window



4. Leave the settings as they are.
5. Click **Login**.
6. Open the **Reporter** application.

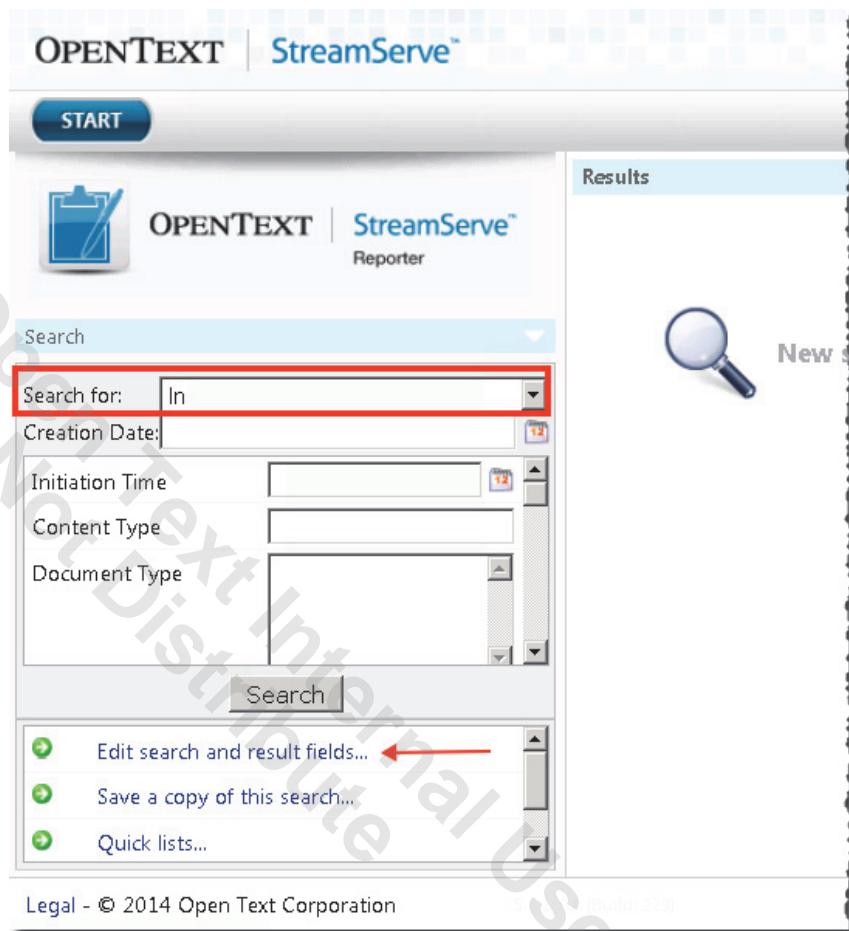
Figure 6-74:

The screenshot shows the StreamStudio Reporter application. The top navigation bar includes 'OPENTEXT | StreamServe™', 'Application Domain Administrator : DE', 'My Preferences | Logout', and a 'START' button. Below the navigation bar, there's an 'Intro' section with a welcome message and a 'Finding Help' section. The main content area has tabs for 'Reporter' and 'Applications'. Under 'Applications', there are four items: 'Composition Center', 'Administrator', 'Collector', and 'Reporter'. The 'Reporter' item is highlighted with a red border.

7. In the Search for field, select to search for **In jobs**.

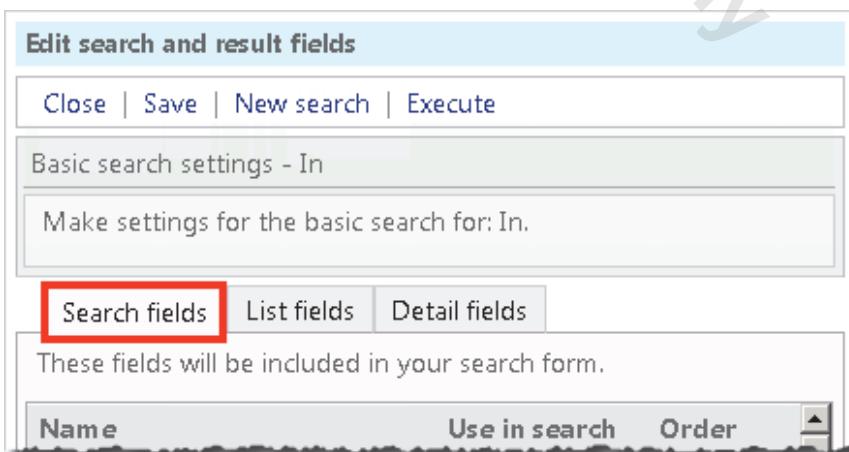
8. Click **Edit Search and result fields** to define a search.

Figure 6-75:  
Configuring In Search



9. Select the **Search fields** tab.

Figure 6-76:  
Search Fields Tab



10. Select the first 6 fields and enter 1 through 6 in the Order column and click Save.

Figure 6-77:  
Setting the Search Fields

Basic search settings - In

Make settings for the basic search for: In.

These fields will be included in your search form.

Name	Use in search	Order
General document data		
Initiation Time	<input checked="" type="checkbox"/>	1
Content Type	<input checked="" type="checkbox"/>	2
Document Type	<input checked="" type="checkbox"/>	3
FailedExpiringDateTime	<input checked="" type="checkbox"/>	4
External job ID	<input checked="" type="checkbox"/>	5
Expire Date	<input checked="" type="checkbox"/>	6

These fields will be included in the basic search for In.

11. Select the **List fields** tab and then select the first 6 fields, enter **1** through **6** in the **Order** column, select the **Sort asc** for the **Initiation Time** field and click **Save**.

Figure 6-78:  
Setting the List Fields

Basic search settings - In  
Make settings for the basic search for: In.

Search fields List fields Detail fields

These fields will be used in your list.

Name	Display	Sort asc	Sort desc	Order
General document data				
Initiation Time	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	1
FailedExpiringDateTime	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	2
External job ID	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	3
Expire Date	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	4
Times Accessed	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	5
ImportanceLevel	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	6

These fields will be included in the list.

12. Select the **List fields** tab and then select the first 6 fields, enter 1 through 6 in the **Order** column, select the **Sort asc** for the **Initiation Time** field and click **Save**.

Figure 6-79:  
Setting the List Fields

Close | Save | New search | Execute

Basic search settings - In

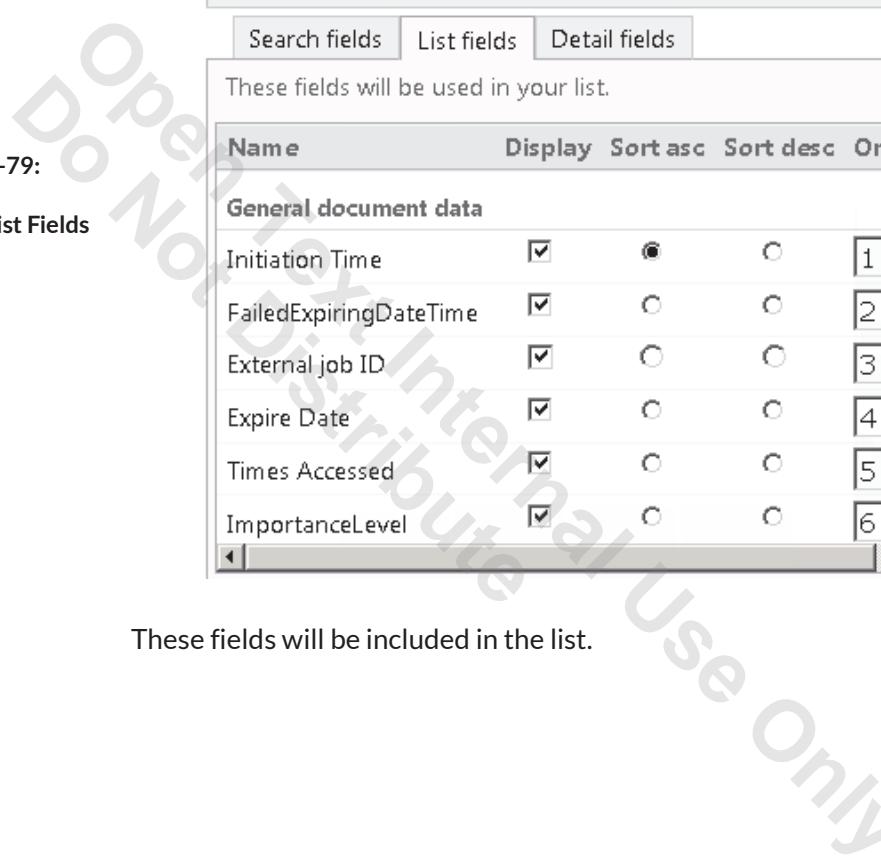
Make settings for the basic search for: In.

Search fields List fields Detail fields

These fields will be used in your list.

Name	Display	Sort asc	Sort desc	Order
General document data				
Initiation Time	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	1
FailedExpiringDateTime	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	2
External job ID	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	3
Expire Date	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	4
Times Accessed	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	5
ImportanceLevel	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	6

These fields will be included in the list.



13. Select the **Detail fields** tab and then select the first 6 fields, enter 1 through 6 in the **Order** column and click **Save**.

Figure 6-80:  
Setting the List Fields

Name	Display	Order
Initiation Time	<input checked="" type="checkbox"/>	1
FailedExpiringDateTime	<input checked="" type="checkbox"/>	2
External job ID	<input checked="" type="checkbox"/>	3
Expire Date	<input checked="" type="checkbox"/>	4
Times Accessed	<input checked="" type="checkbox"/>	5
ImportanceLevel	<input checked="" type="checkbox"/>	6

These fields will be included in the details.

14. Save the search and click **Close**.

You are taken back to the search configuration page.

15. In the **Search for** field, select to search for **Out** jobs.

16. Repeat step 9-14 for output jobs.

You are taken back to the search configuration page.

You can execute the searches with or without providing search criteria.



#### Use Reporter to See Job Details

1. In the **Search for** field make sure **Out** is selected.
2. Click the **Search** button.

In the right panel a list of jobs displays. The content of this list vary depending on the jobs that have been processed. Notice that the Results - Output jobs grid only contains those fields indicated in the previous activity.

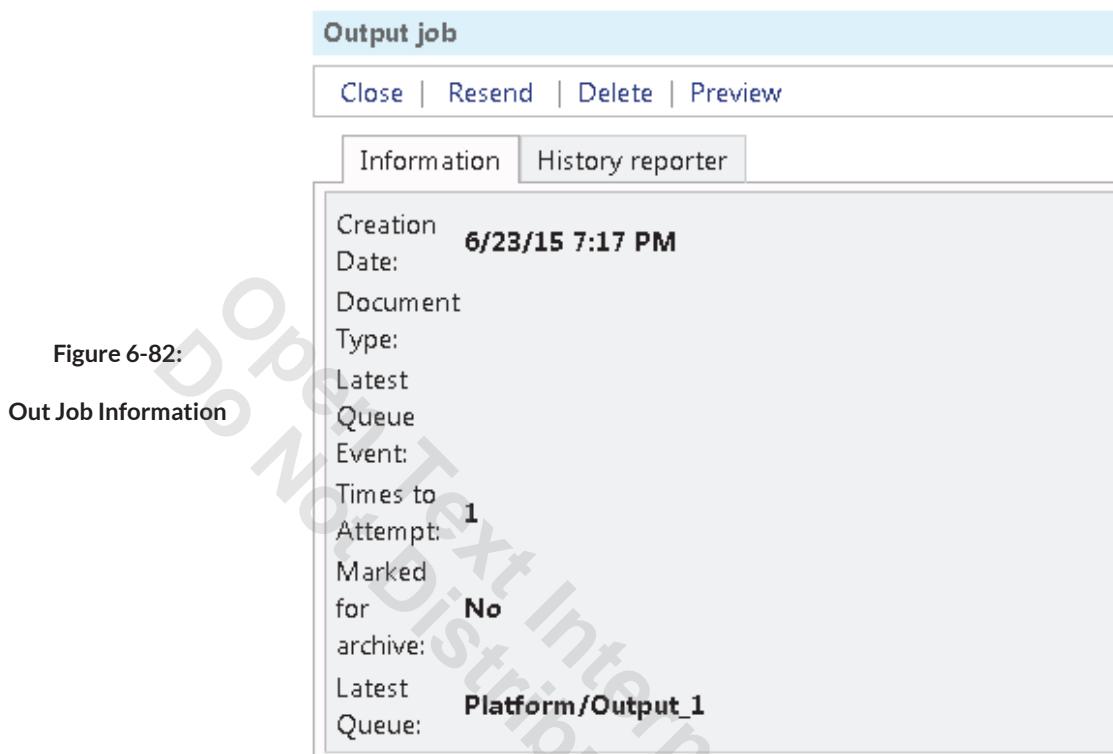
Figure 6-81:

Out Jobs

Results - Output jobs				
<a href="#">Close</a>   <a href="#">Delete selected</a>   <a href="#">Resend selected</a>   <a href="#">Preview selected</a>				
	Creation Date	Document Type	Latest Queue Event	Ti
<input type="checkbox"/>	6/9/15 4:54 PM			1
<input type="checkbox"/>	6/23/15 7:17 PM			1
<input type="checkbox"/>	6/23/15 7:17 PM			1
<input type="checkbox"/>	6/23/15 7:17 PM			1
<input type="checkbox"/>	6/23/15 8:26 PM			1
<input type="checkbox"/>	6/23/15 8:49 PM			1
<input type="checkbox"/>	6/23/15 8:49 PM			1

3. Double-click any job to see its details.

4. Examine the *Information* tab for the output job.



5. Click the *History Reporter* tab.

The screenshot shows the "Output job" interface again, but the "History reporter" tab is now selected. At the top, there are buttons for "Close", "Resend", "Delete", and "Preview". Below these are two tabs: "Information" and "History reporter" (which is selected). The main content area is a table titled "Job History" with the following columns: Job No, Initiation Time, Last Processed Time, and Status. One row is present in the table:

Job No	Initiation Time	Last Processed Time	Status
10	6/23/15 7:17 PM	6/23/15 7:17 PM	Completed

On the left side of the window, there is a vertical sidebar with the heading "History Reporter" and the text "Figure 6-83:".

6. Examine the *History Reporter* tab for the output job.

Notice whether the job was completed and if it reports any errors.

7. Click the **Preview** link.

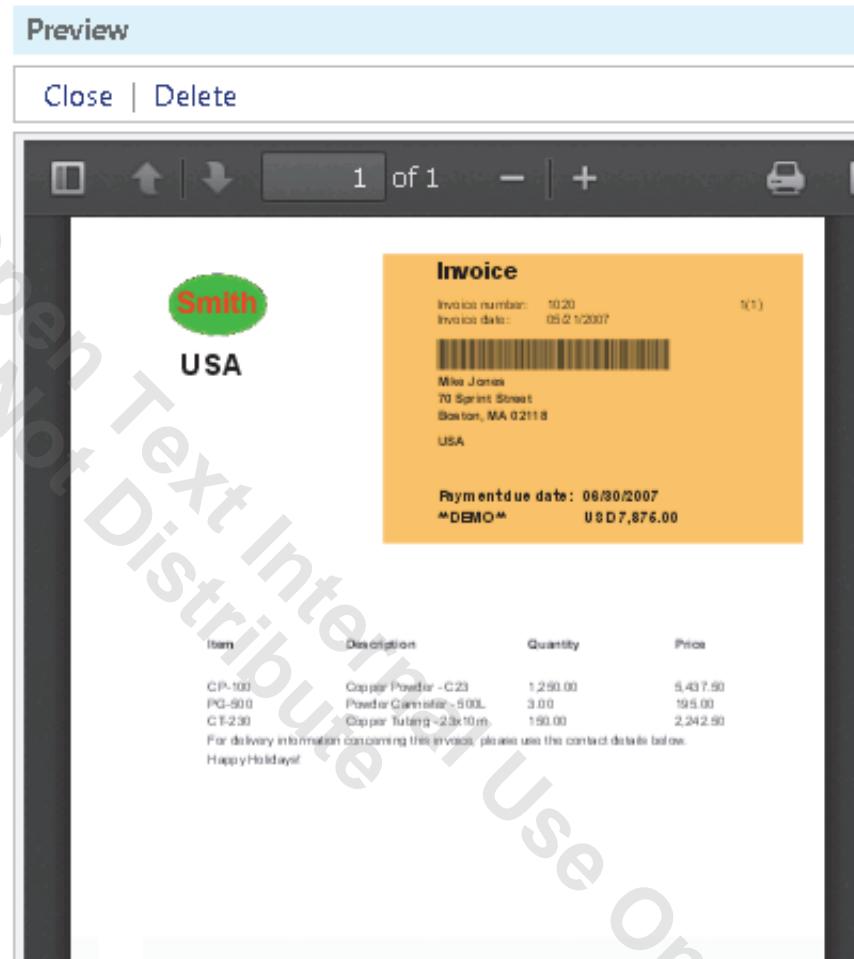


Figure 6-84:

Preview

8. Click **Close**.



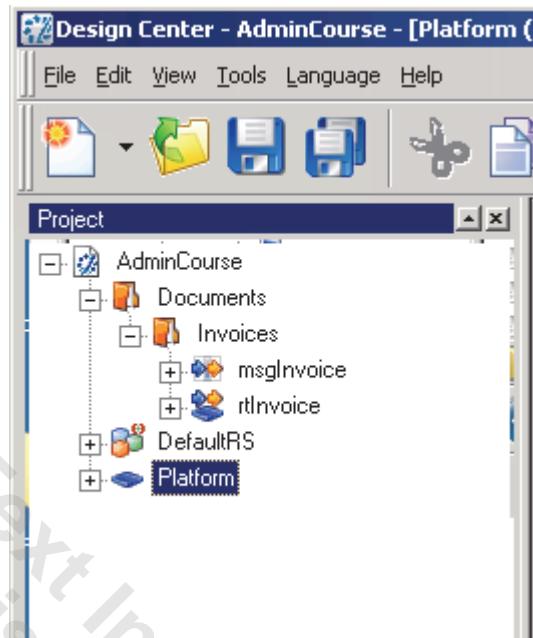
#### **View Input Data and Reprocess a Job**

1. In Design Center, open the **AdminCourse** project.

2. In the Project panel, double-click **Platform**.

Figure 6-85:

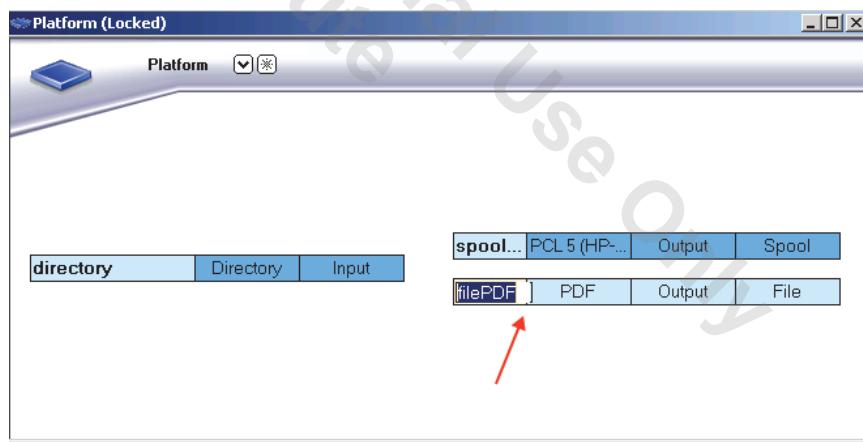
Platform



3. In the right panel, double-click the **filePDF** output connector.

Figure 6-86:

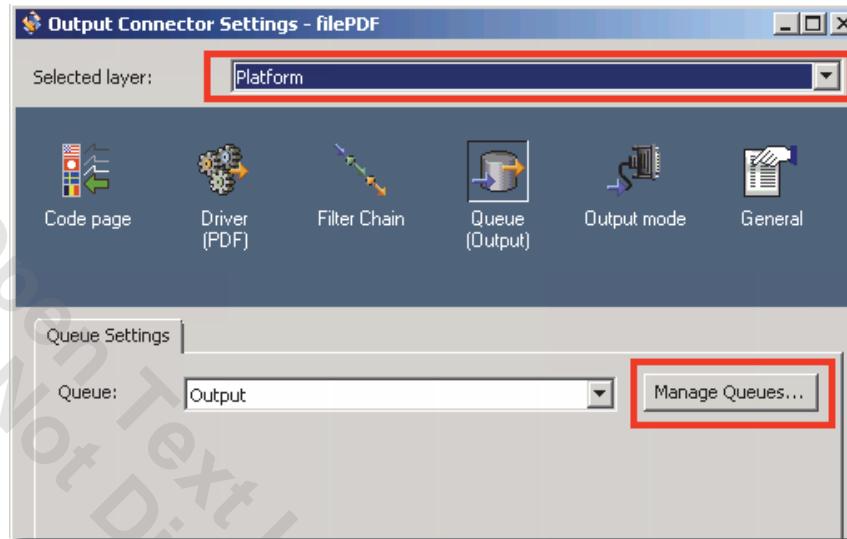
Output Connector



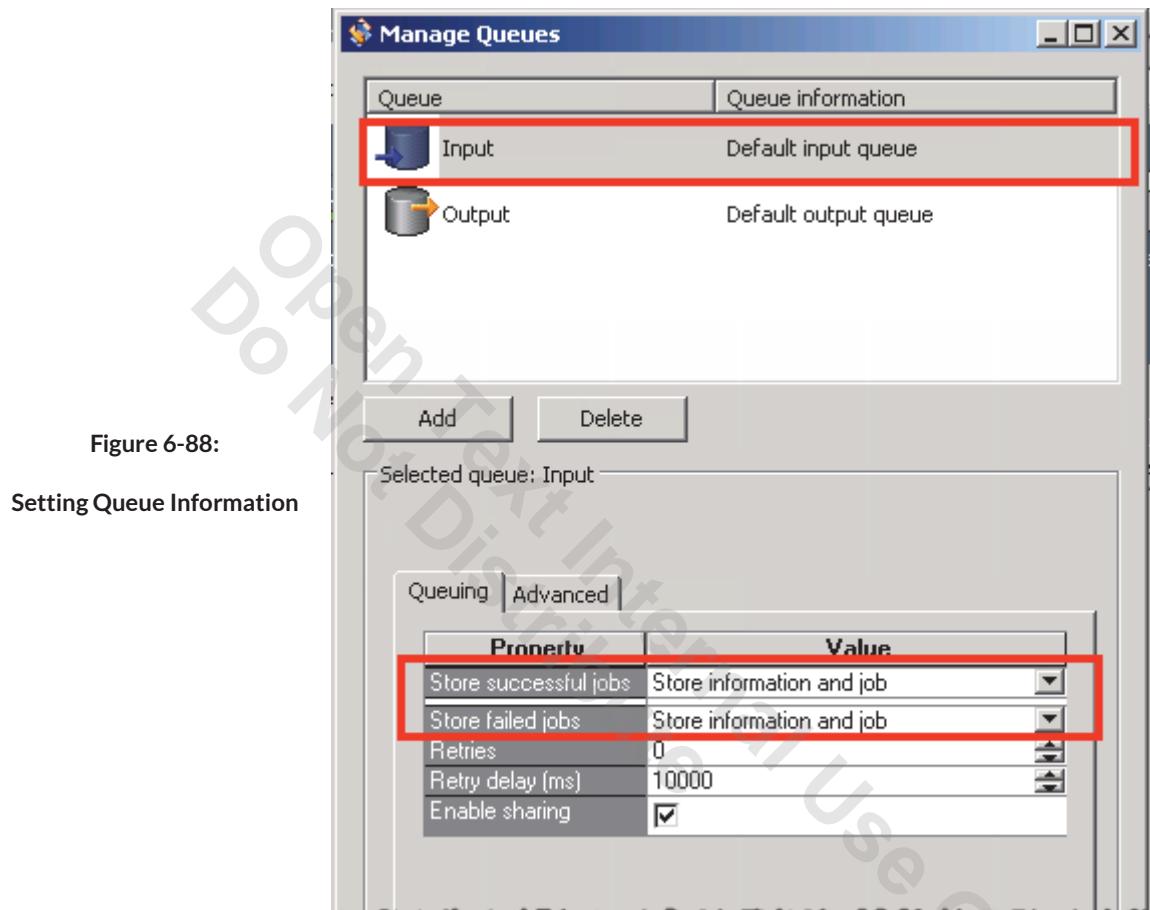
4. In the Output Connector Settings - filePDF window, make sure **Platform** is selected in the Selected layer drop-down and select the **Manage Queues** button.

Figure 6-87:

Manage Queues

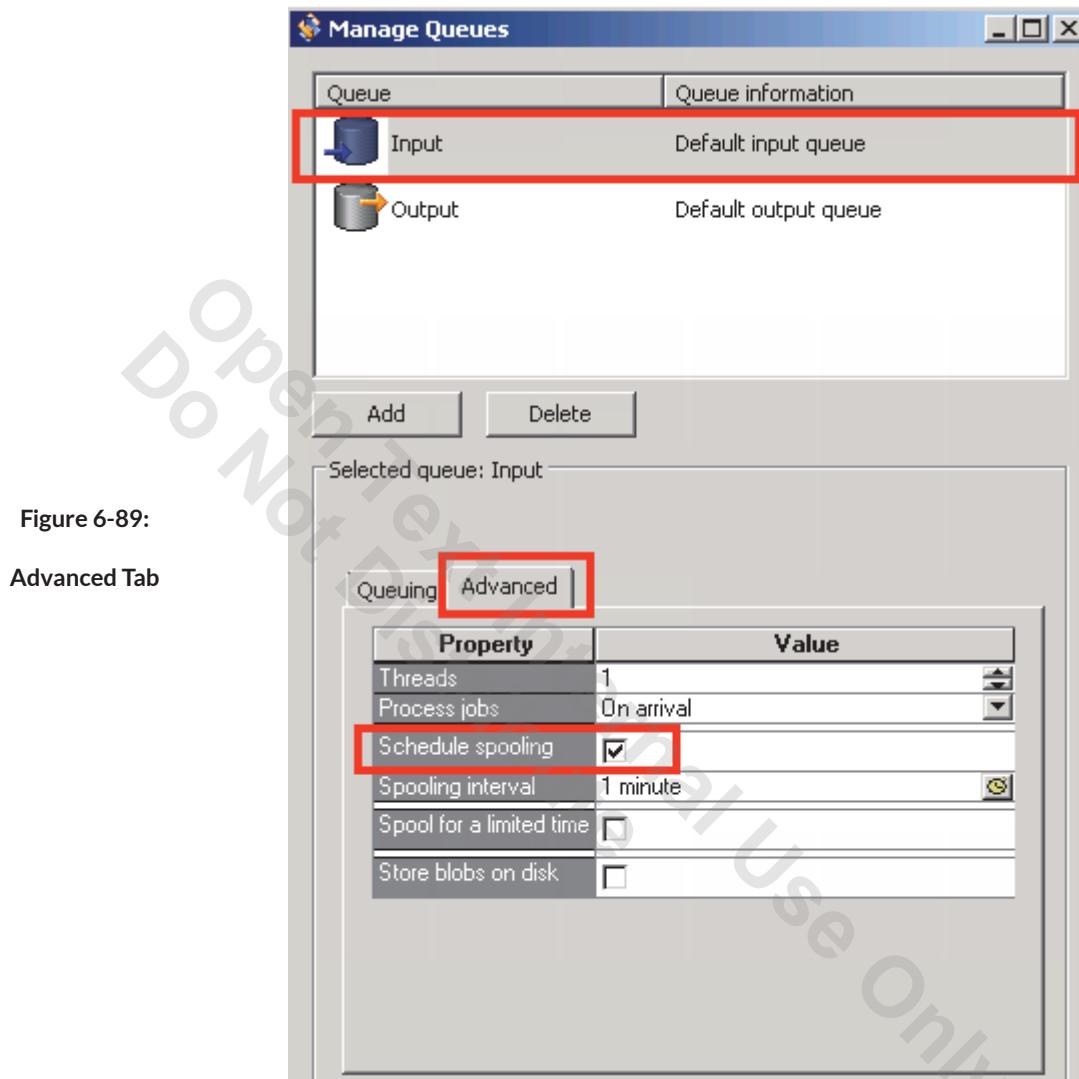


5. In the Manage Queues window, select the **Input** queue in the upper panel, then select **Information and Job** for the Store successful jobs and Store failed jobs properties.



6. Select the **Advanced** tab.

7. Check the *Schedule spooling* property.



8. In the upper panel of the Manage Queues window, select the **Output** queue, then select **Information and Job** for the **Store successful jobs** and **Store failed jobs** properties (in the **Queuing** tab).
9. Select the **Advanced** tab.
10. Check the *Schedule spooling* property.
11. Click the **OK** button.
12. Click the **OK** button.

13. Save all the changes and click the **Export** button.

Figure 6-90:  
Export Button

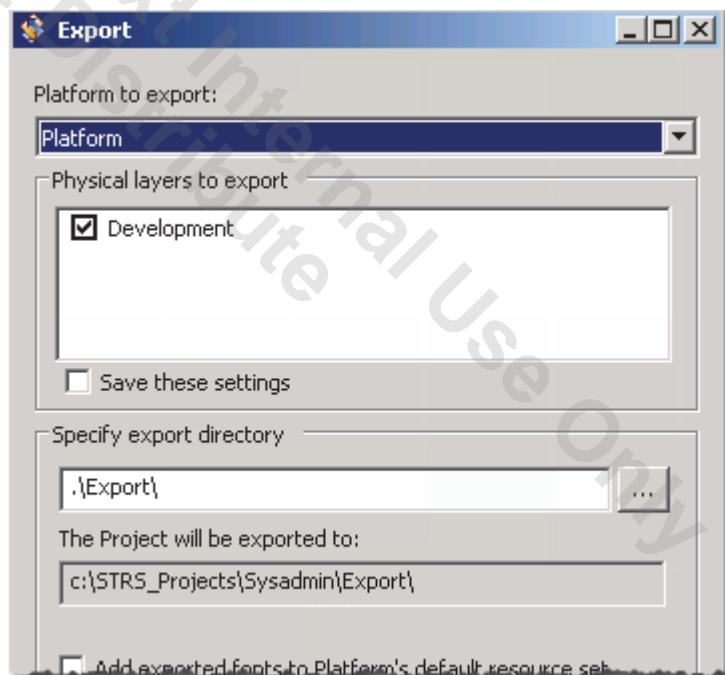


14. Make sure the following values are set respectively and click the Export button:

- Physical layers to export: checked
- Export Directory: .\Export\

15. In Control Center, Redeploy the **AdminCourse** application.

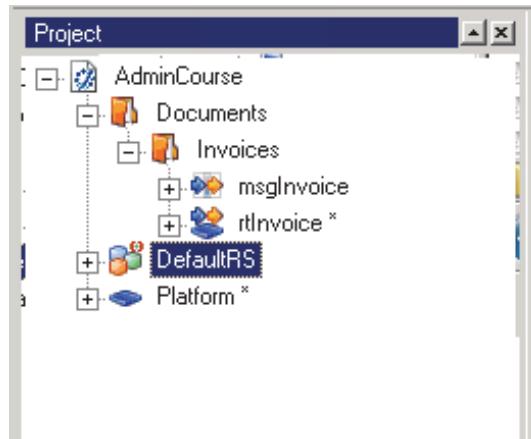
Figure 6-91:  
Export Setting



16. In Design Center double-click the **DefaultRS** node.

Figure 6-92:

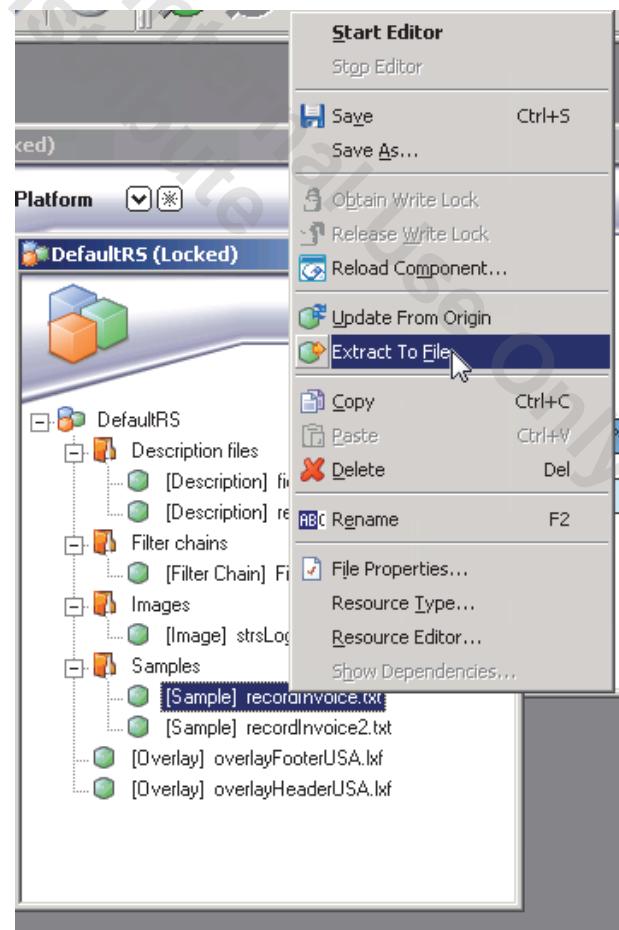
DefaultRS Node



17. In the DefaultRS window (right panel), right-click [Sample]recordInvoice.txt and select **Extract To File** in the pop-up menu.

Figure 6-93:

Extract to File



18. Navigate and save the recordInvoice.txt file in the C:\DEV\IN folder.
19. In the DefaultRS window (right panel), right-click [Sample]recordInvoice2.txt and select Extract To File in the pop-up menu.
20. Navigate and save the recordInvoice2.txt file in the C:\DEV\IN folder.
21. In Reporter, perform a search for **input** jobs.
22. Double-click any of the returned input jobs.
23. Click **Download input data** and save the file to your desktop.

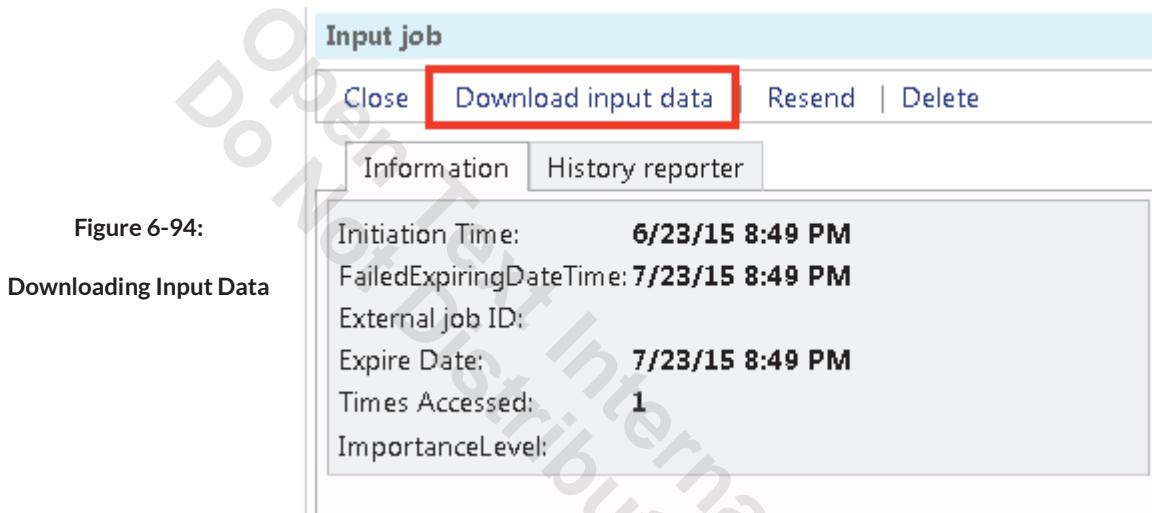


Figure 6-94:  
Downloading Input Data

Because input jobs are sometimes very big, they are not viewed inside the Reporter. To view the input job you must download it to a disk instead.

24. Open the input data file to verify that the data looks as expected.
25. Close the file.
26. Examine the details for the output job.
27. Go back into the **Reporter** and click **Display out jobs** to see all output jobs related to the selected input job.
28. Open an output job to view the contents of it and verify that it looks as expected.
29. Click **Resend** to resend the output job using the same settings as for the original output job.
30. Verify that the job was resent as expected.
31. Close the job.

Open Text Internal Use Only  
Do Not Distribute

## 7. Common Administrative Tasks

On completion of this (section/module), participants should be able to:

- Configure the Task Scheduler application
- Deploy document types
- Use the Database Administration Tool
- Managing StreamServe applications
- Apply a hotfix to the database
- Get information used to log support incidents
- Managing Gateway users
- Directory structure for StreamServe applications

### Task Scheduler Overview



Figure 7-1:

Task Scheduler

You can add a Task Scheduler application and schedule one or more tasks to be carried out by the application. For example:

- Run a batch file that triggers maintenance of the runtime repository each night at 1:00 a.m.
- Delete expired StreamServe jobs from the runtime repository every 60 minutes.

The following tasks are available in the Task Scheduler application:

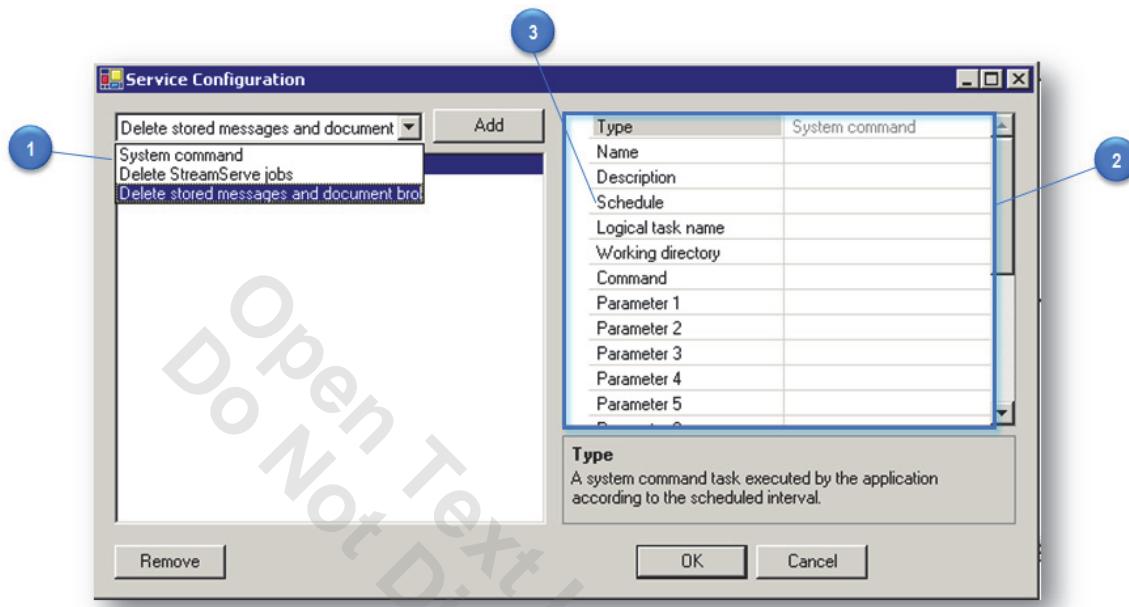
Task	Description
Run system command	A system command task.
Update job status	Updates and reports the statuses of top jobs.
Delete expired jobs	Deletes expired StreamServe jobs from the runtime repository.
Delete expired Messages	Deletes expired Messages from Message storages in the runtime repository.
Delete expired documents	Deletes expired documents from Post-processing storages in the runtime repository.
Run database maintenance	Performs database maintenance on the runtime repository.

By default, the StreamServer applications in an application domain perform the following maintenance tasks on the runtime repository:

- Update statuses of top jobs.
- Delete expired jobs.
- Delete expired Messages from Message storages.
- Delete expired documents from Post-processing storages.

If you run StreamServer applications with a high-load, you may gain performance and avoid deadlocks by letting one or several Task Scheduler applications perform these tasks instead.

## Configuring a Task Scheduler Application



**Figure 7-2: Task Scheduler**

You configure Task Schedule applications in Control Center. The Service Configuration dialog box is used to set up the tasks for a Task Scheduler application.

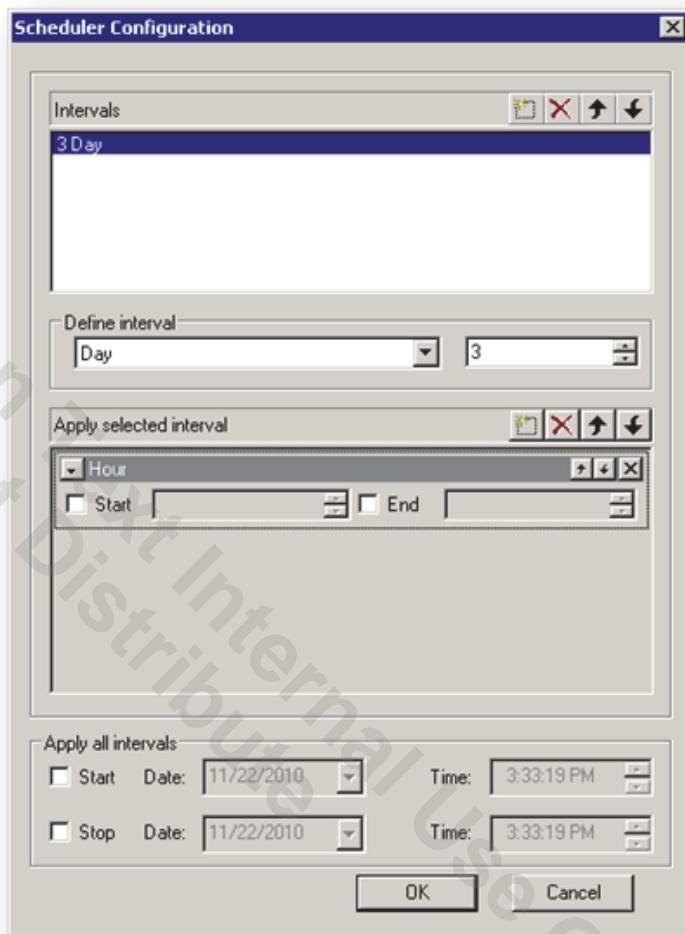
**Settings – Service Configuration Dialog Box**

Task	Description
(1) Task	Select the type of task you want to schedule. You can add several tasks to the same application.
(2) Task parameters	The parameters required depend on the type of task. For more information see the online help.
(3) Schedule	Used to specify when the task is run.

To ensure that the tasks are executed even if one Task Scheduler application goes down, you can add several Task Scheduler applications to an application domain and schedule tasks for each application.

## Configuring the Schedule

Figure 7-3:  
Scheduler Configuration



You can schedule the interval at which an application performs a task. For example, the interval at which a Task Scheduler application performs a system command task.

You can set a single interval, or create more complex schedules.

If you specify a stop time (or end time), all ongoing tasks will continue until they are finished, even if the stop time is passed.

If a new task is scheduled to start before an on-going task has completed, the application first finalizes the ongoing task before the new task is started.

The Scheduler Configuration dialog box is used to schedule intervals. For more information see the online help.

## Deploying Document Types

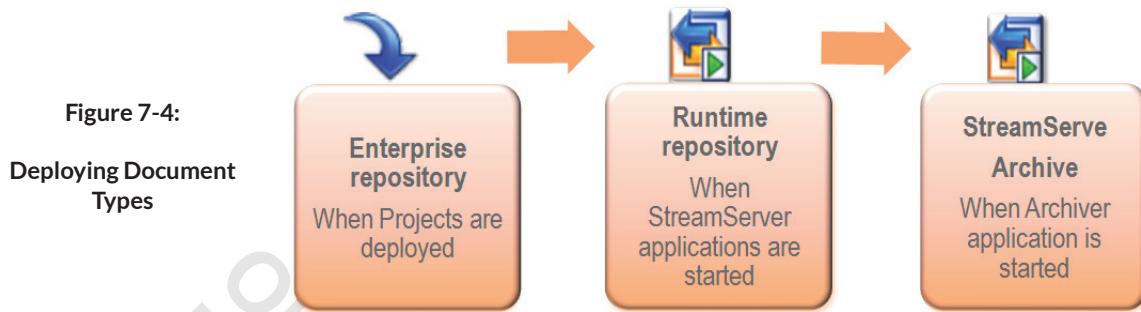


Figure 7-4:

Deploying Document Types

Document types are used to categorize documents, for example invoices, orders, etc.

For each document type, a number of metadata can be specified, such as, invoice number customer name, etc.

Metadata is added to groups, which contain a specific class of metadata. For example, a group can contain customer metadata, such as customer name and customer number.

A document type can be used in different contexts:

- Archive context
- Message context
- Post-processing context

Metadata must be enabled in each context where it will be used.

**Metadata Examples** Metadata can be used to:

- Search for documents from StreamStudio Collector.
- Write personalized marketing messages in StreamStudio Composition Center.

Document types and metadata are configured in Design Center.

The document types (including metadata) must be stored in the StreamServe Enterprise Repository, the runtime repository, and the StreamServe archive.

The document types in the StreamServe archive and runtime repository must comply with the document types in the enterprise repository.

### How Document Type Deployment Works

Repository	Description
StreamServe Enterprise Repository	When you deploy a Project to a StreamServer application in Control Center, the document types are stored in the enterprise repository.
Runtime repository	When the StreamServer application is started, the document types are copied to the runtime repository.  Note, only document types associated with the application domain that the runtime repository belongs to are stored in the repository.
StreamServe archive	If you run StreamStudio Collector, document types are stored in the StreamServe Archive when you start the Archiver application.  Only document types associated with the application domain(s) that the StreamServe archive is linked to and that include metadata intended for archiving are stored in the StreamServe archive.

## Database Administration Tool

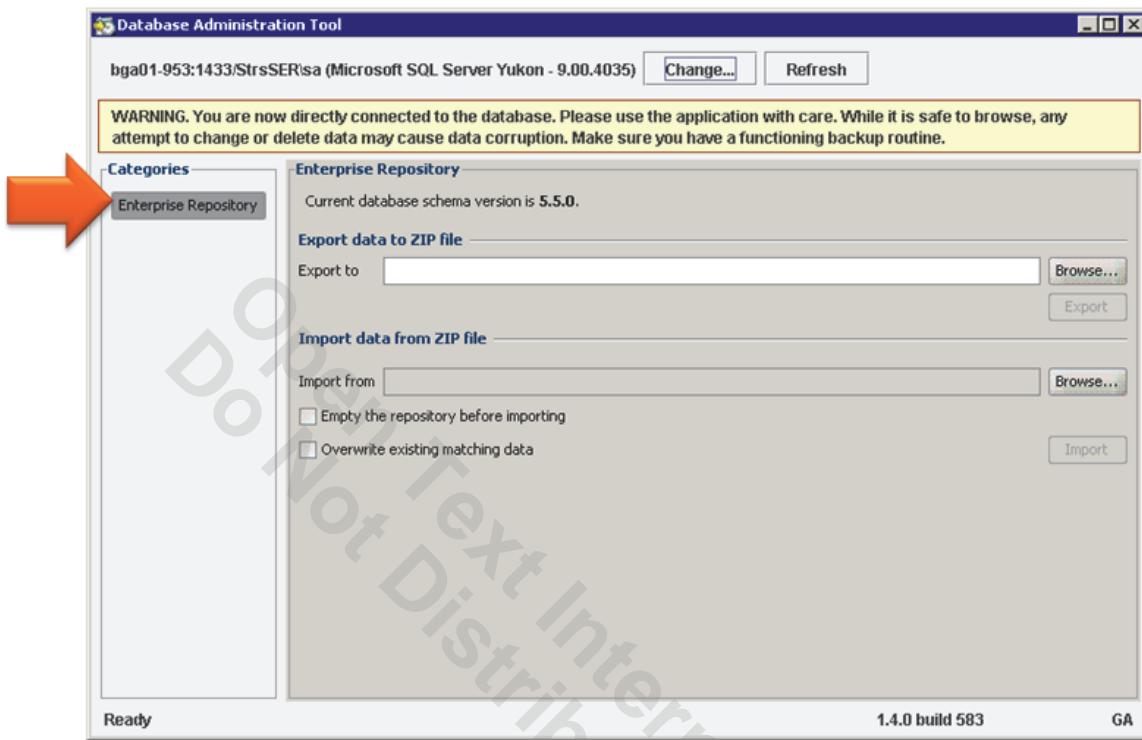


Figure 7-5: Database Administration Tool

Database Administration Tool can be used to perform administrative tasks for the following repositories:

- Runtime repository
- Enterprise repository
- Web content repository

The Database Administration Tool should only be used for tasks that cannot be performed using StreamStudio applications, Design Center, or Control Center.

All StreamServe applications (StreamServer, Archiver, service gateway, etc.) must be stopped before applying any changes to a repository.

Incorrect usage of Database Administration Tool can have serious consequences. You should not perform any actions without having the proper knowledge about the effects changes may have on StreamServe applications running in production environments, and on data used in StreamStudio applications.

The Database Administration Tool can be opened from Control Center by users who have the Administrator role.

## Database Administration Tool – Runtime Repository

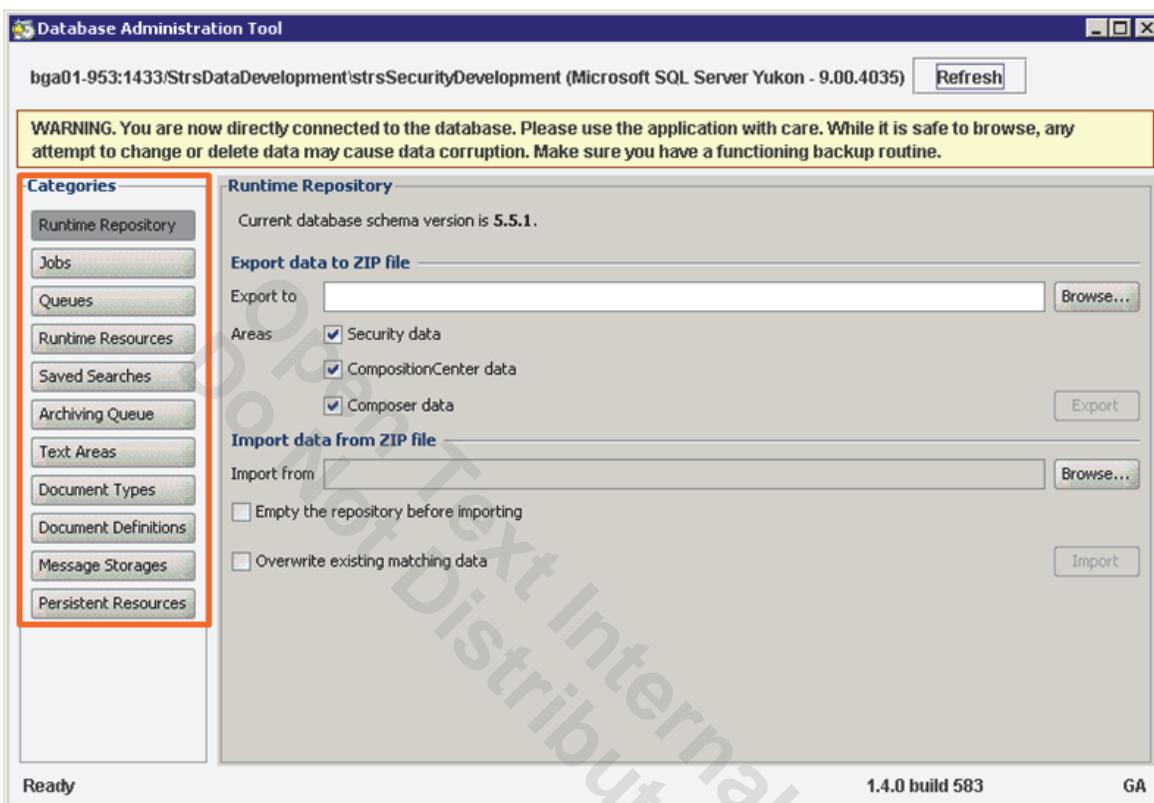


Figure 7-6: Database Administration Tool – Runtime repository

The Database Administration Tool can be used to perform the following maintenance tasks on the runtime repository:

- Migrate resources and security data (roles, permissions, etc.) from one runtime repository to another when upgrading or when migrating data between test, production, etc.
- Examine, expire, delete, and cancel jobs.
- Cancel queued jobs.
- Delete resources.
- Examine documents in a Message storage.
- Expire and delete documents in a Message storage.
- Drop Message storages.

## Exercise 1 – Create a Task Scheduler Application

In this exercise, you will create a Task Scheduler application that is used to delete expired jobs from the runtime repository. The following steps are required for this:

- Configure the expiry time for documents.
- Disable the default schedule for deleting expired jobs in the runtime repository.
- Create a Task Scheduler application with a task to delete expired jobs.
- Add the interval at which the task is executed.
- Start the Task Scheduler application and view the results.

In order to see the results of the deleting expired jobs from the runtime repository, we will set a 1 minute expiry time for successfully processed documents. This means that after a job is successfully processed, it remains in the runtime repository for 1 minute before being handled by the job deletion process.



### Configure the Expiry Time for Documents

1. Open the **AdminCourse Project** in Design Center.
2. In the Project browser, double-click the **Platform** node.

The generic layer is activated in the Platform view.

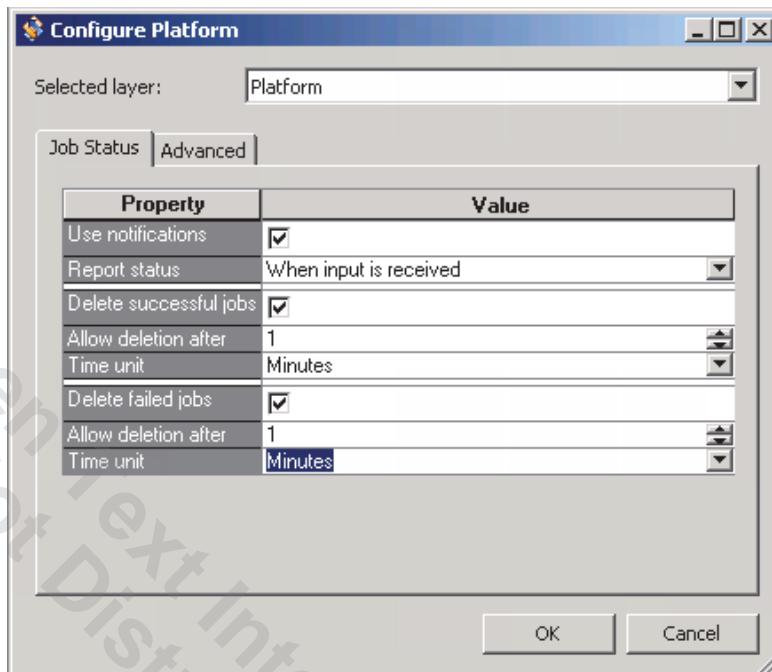
3. Right-click anywhere in the blank space of the Platform window and select **Configure Platform** in the pop-up menu.

The Configure Platform dialog box opens.

4. In both **Allow deletion after** field, select **1**.

5. In the Time units field, select Minutes.

Figure 7-7:  
Configure Platform



6. Save the Project, export and deploy the application using the newly generated export file.  
7. Start the AdminCourse application.

By default, StreamServer applications are configured to delete expired jobs every hour. As we will use Task Scheduler for this, we must disable the default settings for each of the StreamServer applications running in the application domain.



#### Disable the Default Settings to Delete Expired Jobs

1. In Windows Explorer, browse to the working directory for the AdminCourse application  
C:\ManagementGateway\5.6.2\root\applications\AdminCourse\Development
2. Using Notepad++, edit the **repositorymanager.xml** file for the AdminCourse application.

3. Delete the following section from the file.

```
<deleteevent schedule="T II * * MH * * 60" />
<!-- heartbeatevent schedule="T II * * MH * * 20"
interval="PT20M" / -->
<deletemarkevents>
<toppart use="default" schedule="T II * * MH * *
20" />
</deletemarkevents>
```

4. Save and close the file.



#### Create the Task Scheduler Application with a Delete Expired Jobs Task

1. Open the AdminCourse Project in Design Center.
2. In Control Center, right-click the application domain and select New Application.

The New Application dialog box opens.

3. Configure the application properties for the new Task Scheduler.
  - Application type: TaskScheduler5.6.2
  - Name: TS\_DeleteJobs
4. Click OK.

The Configuration dialog box opens.

5. Click anywhere in the **Tasks** line and then click the ... button in the right column.

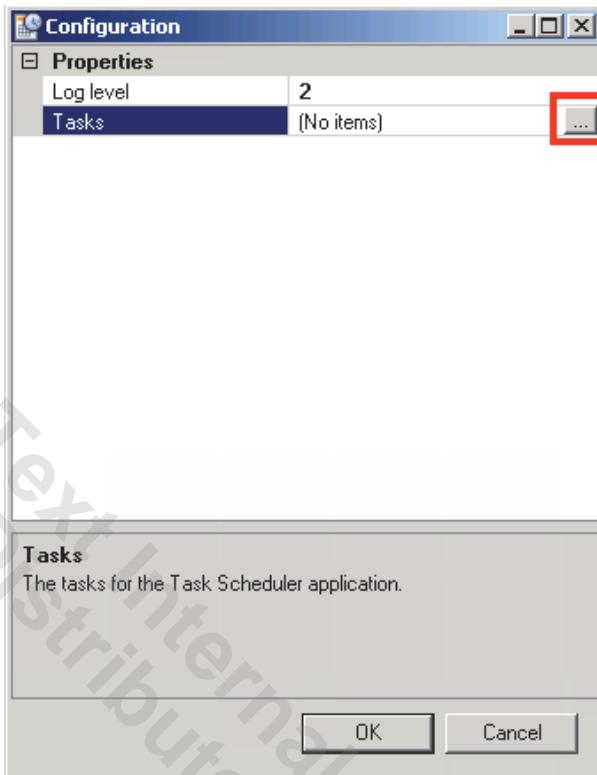


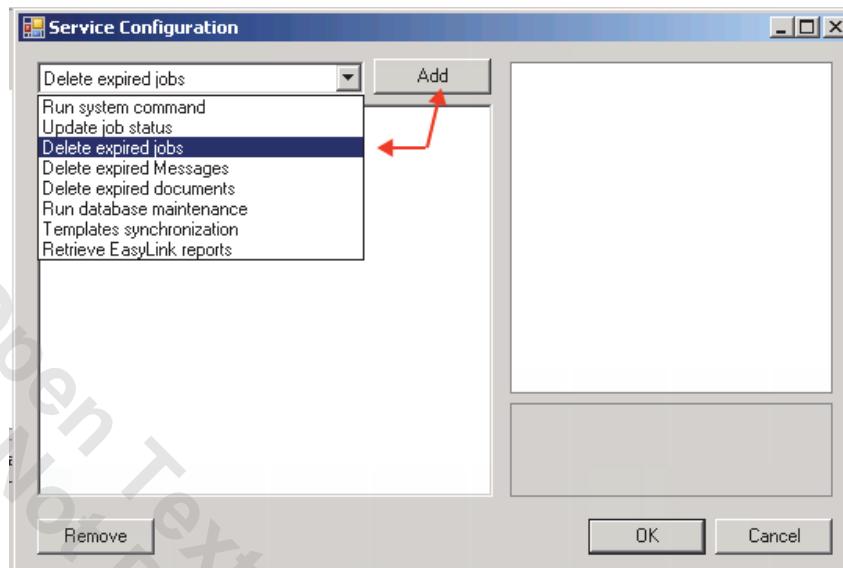
Figure 7-8:  
Configuration Window

The Service Configuration dialog box opens.

6. Select the **Delete expired jobs** task and click **Add**.

Figure 7-9:

Service Configuration Window

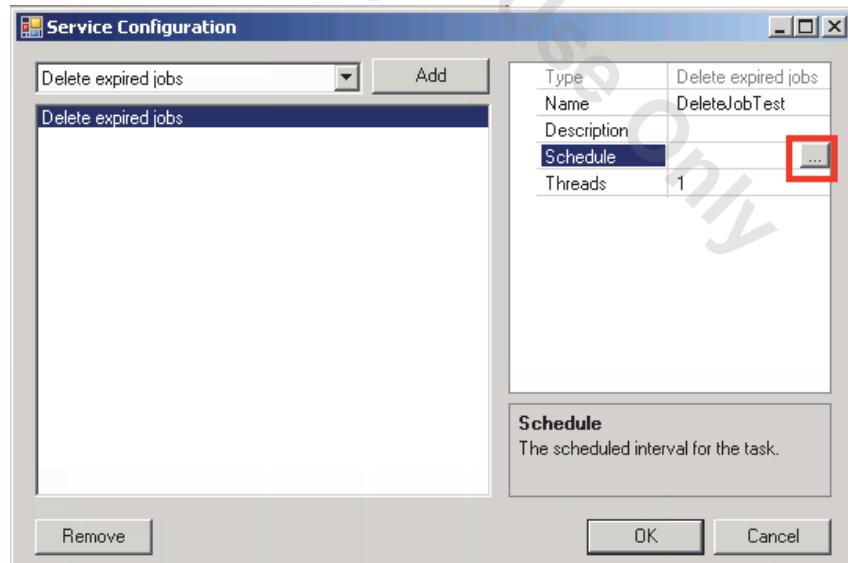


7. Configure the properties as indicated below:

- Name: **DeleteJobsTest**
- Threads: **1**

8. In the Schedule field, click...

Figure 7-10:  
Schedule Configuration



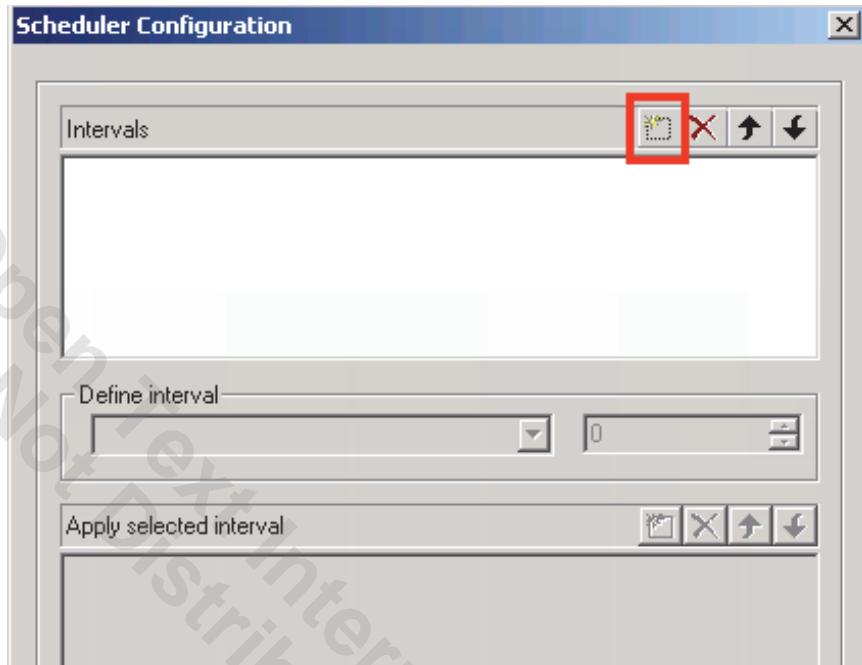
The Scheduler Configuration dialog box is opened.

 **Define the Schedule for the Delete Expired Jobs Task**

1. In the Intervals area, click the New (Insert) button.

Figure 7-11:

New Button



2. From the Define interval drop-down list, select **Minute** and **1**.

3. In the *Apply all intervals* area, select **Start** and **today's date**.

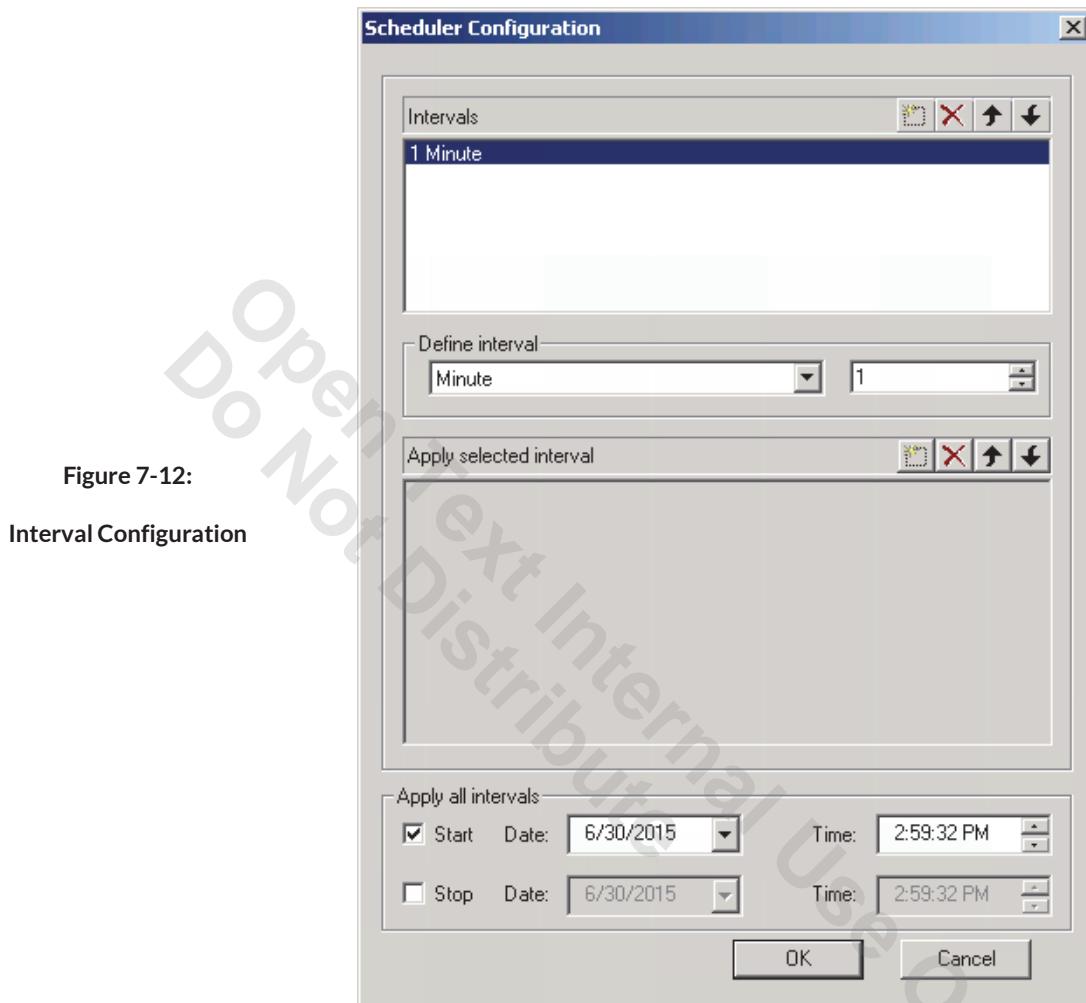


Figure 7-12:

Interval Configuration

4. Click **OK** to close the Scheduler Configuration dialog, **OK** to close the Service Configuration box and **OK** to close the Configuration dialog box.

Before running the Task Scheduler application, we will examine the jobs in the runtime repository.



#### **View the Jobs and Expiry Times in the Runtime Repository**

1. Send some jobs to the **AdminCourse** application.
2. In Microsoft SQL Server 2012 Management Studio, open the database for the runtime repository.

## 3. Run the query:

```
select * from [StrsData_DEV].[dbo].Part where
ExpiringDateTime is not null
order by CreationDateTime desc
```

## 4. View the number of entries in the results.

**Run the Task Scheduler Application and View the Results**

1. Start the **TS\_DeleteJobs** application.
2. In Microsoft SQL Server Management Studio, open the database for the runtime repository.
3. Run the query:

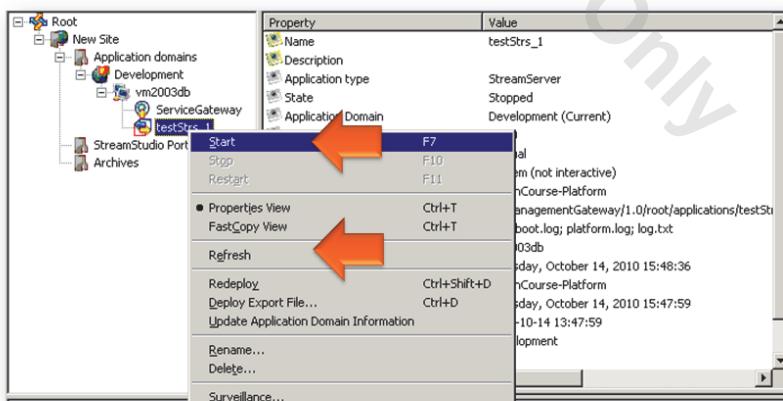
```
select * from [StrsData_DEV].[dbo].Part where
ExpiringDateTime is not null
order by CreationDateTime desc
```
4. View the number of entries in the results.

## Managing StreamServer Applications

### Starting, Stopping, and Restarting Applications

In Control center:

- Right-click to start, stop, or restart an application.
- Select **Refresh** to update the status information displayed for an application.



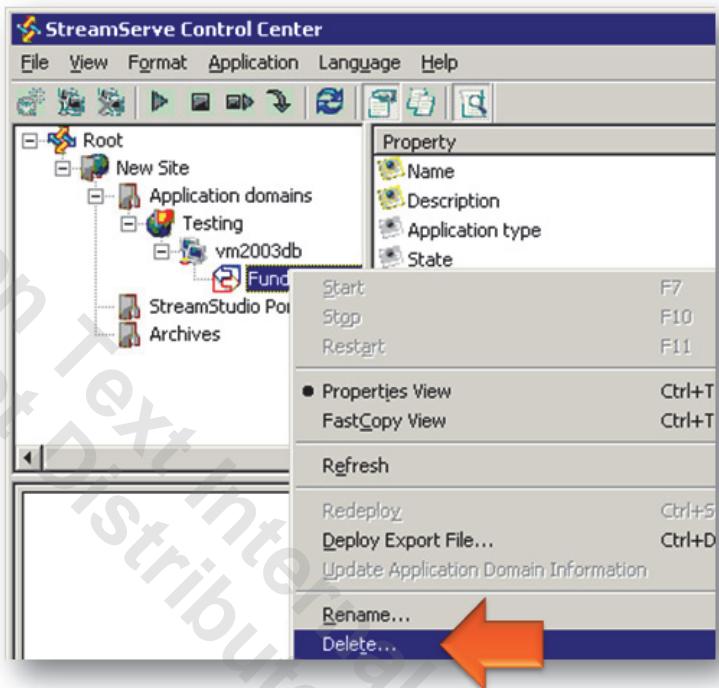
**Figure 7-13:**

### Start, Stop and Restart Applications

**Deleting Applications** You can delete StreamServe applications. This does not delete the working directory. You must manually delete this if required.

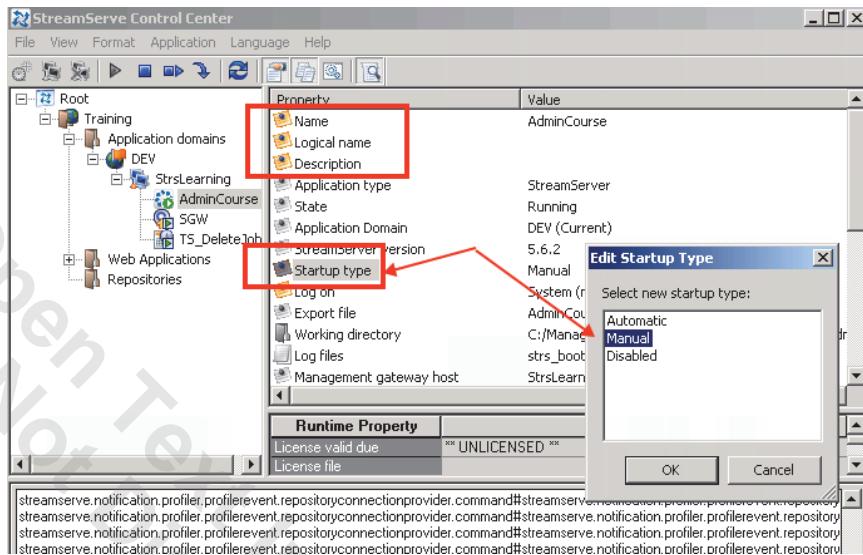
The working directory is covered in the Disk Structure module.

Figure 7-14:  
Deleting Applications



## Updating and Exporting Application Properties

**Figure 7-15:**  
Updating and Exporting Application Properties



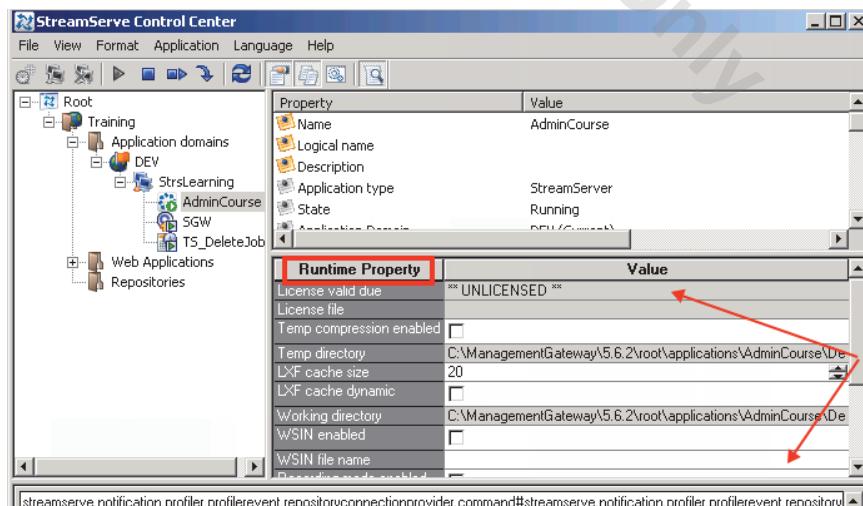
**Updating Application Properties** In the Properties view of the application, you can update the Name, Description, Startup type and Log on properties of applications.

**Exporting Properties** You can save the properties (name, description, version, etc.) for an application as a text file. This is done using the **Export List** option in the **File** Menu.

## Updating Runtime Properties

**Figure 7-16:**

**Updating Runtime Properties**



This view displays the runtime properties for the StreamServer application. For descriptions of these properties see the StreamServer online help.

You can also update the runtime properties of StreamServer applications, such as the log level of a StreamServer application, while the applications are running.

If you update the runtime properties of a StreamServer application, the application is run using the updated values until the application is restarted. When you restart the application, the configuration from the working directory takes effect.

### Adding Other StreamServer Applications

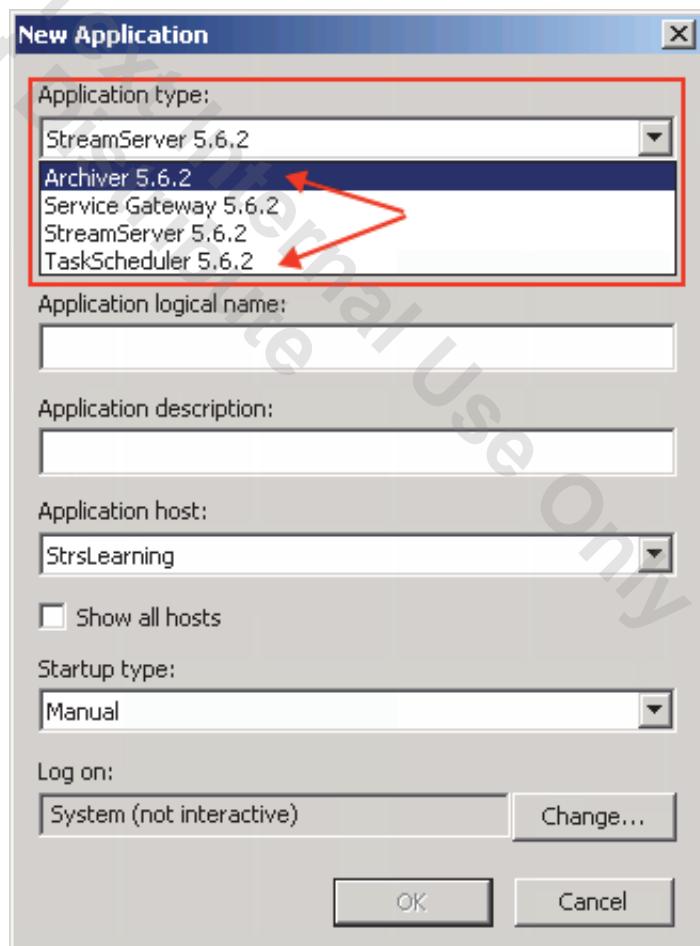


Figure 7-17:

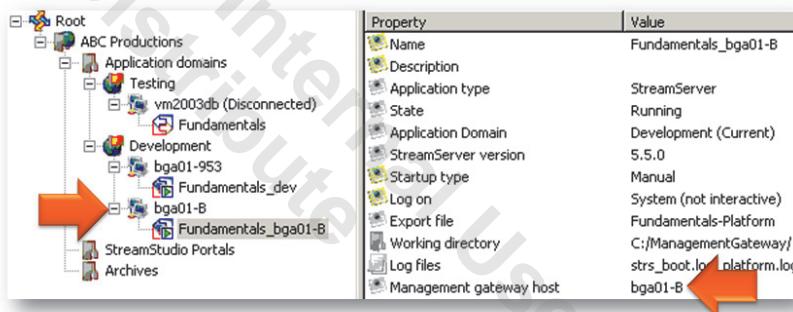
### Adding Other StreamServer Applications

**New Application Dialog Box** In addition to StreamServer applications, this dialog box is also used to add the following application types:

Application Type	Description
Archiver	This application is used with StreamStudio Collector to transfer output documents and metadata from the runtime repository to the StreamServe archive according to a specified schedule.
Service gateway	A Service Gateway is required to run StreamStudio applications.
Task Scheduler	This application schedules one or more tasks to be carried. For example, a task that runs a batch file to trigger reports based on a specified schedule.

## Administering Applications on Remote Hosts

**Figure 7-18:**  
Administering Applications on Remote Hosts



You can use Control Center to run and administer StreamServer applications on both Windows and UNIX hosts.

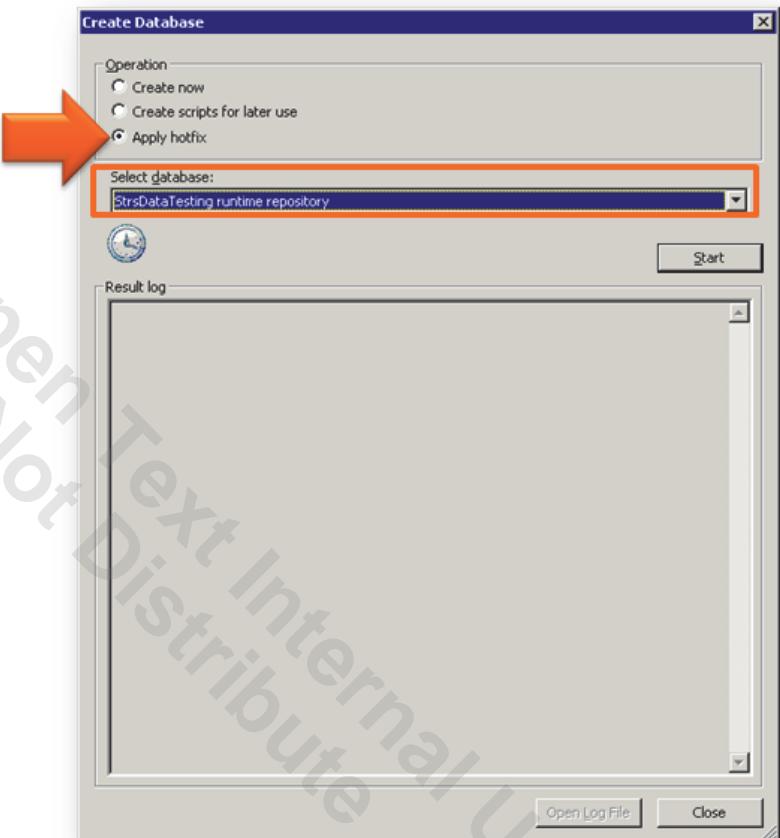
Remote hosts are displayed in the Control Center tree view after you create an application on the host.

To administer applications on a remote host, the host must be connected to the central enterprise repository.

To create an application and perform administrative tasks on the remote host, you need a user name and password to the management gateway.

## Applying a Hotfix to the Database

Figure 7-19:  
Applying a Hotfix to the  
Database



You can apply database hotfixes to the following StreamServe repositories directly from Control Center:

- StreamServe archive
- Runtime repository
- Web content repository
- Enterprise repository

Before you apply a hotfix, you can use the Available Database Hotfixes dialog to see of the current status of a repository. This can help you to decide whether a repository needs to be upgraded.

You use the Create Database dialog to apply a hotfix to a repository.

- Prerequisites**
- The login details for the database administration user are available.
  - The setup for the hotfix has been run and the database hotfix is installed in:

`<Base directory>\<Version>\root\config\database\`

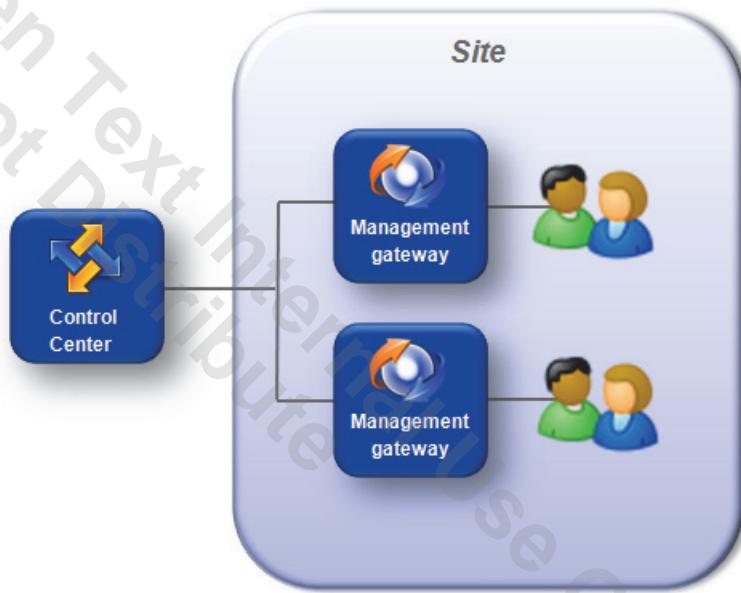
Where `<Base directory>` is the path specified for StreamServe Projects during the Framework and Control Center installation.

For example: C:\ManagementGateway

## Management Gateway Users

Figure 7-20:

Management Gateway  
Users



To log on to Control Center and manage the StreamServe applications on a computer, each user needs a login for the management gateway.

**Roles** Management gateway users can have an Administrator or a Basic role. These roles control the access rights for users in Control Center.

**Access Rights** Some examples of the privileges only available to Administrators include:

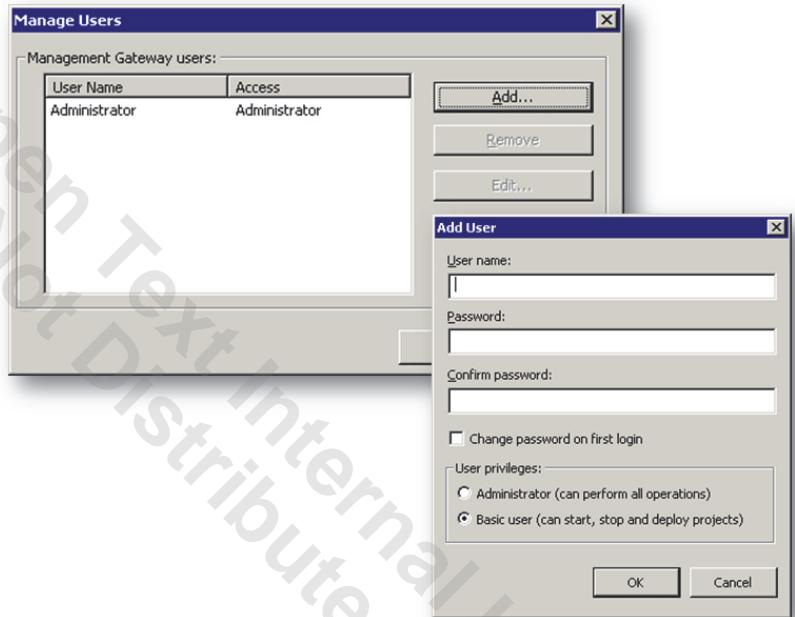
- Creating repositories and deploying hotfixes.
- Creating application domains.
- Forcing document type updates into the enterprise and runtime repositories when deploying Design Center Projects.
- Managing user accounts.
- Creating applications.

In Control Center, users are created at management gateway level.

If a Control Center user needs to manage StreamServe applications on more than one management gateway (or computer), you must create separate users on each management gateway (or computer).

## Creating Users

Figure 7-21:  
Management Gateway  
Users



**Manage Users Dialog Box** You use the Manage User dialog box to add new management gateway users, remove users, and change passwords or roles for users. This dialog box also displays all users on the management gateway.

**Add User Dialog Box** This is used to create new users. When you create a new user, you specify the role for the user.

Management gateway user names and passwords are case-sensitive.

## Exercise 2 – Find StreamServe Components and Hotfix Information

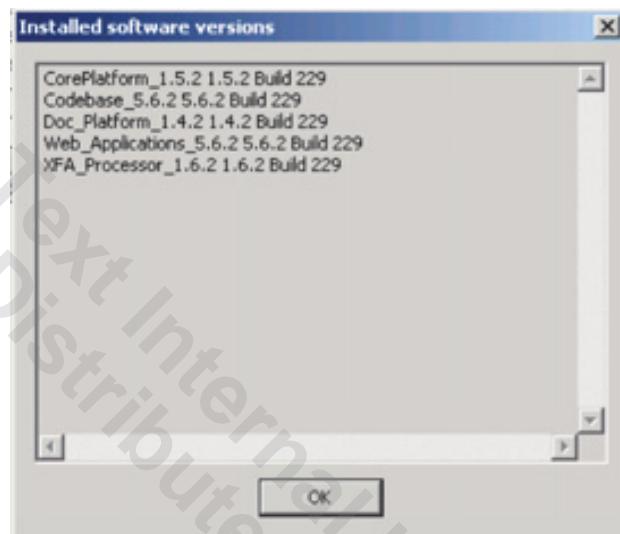
In this exercise, you will find the information about the StreamServe components installed on the computer.



### **Locate the StreamServe Components Installed on the Computer**

1. In Control Center right-click the host node and select **View Installed Versions**. The **Installed software versions** dialog is displayed.

Figure 7-22:  
Installed Software  
Versions



2. Examine the information and click **OK**.



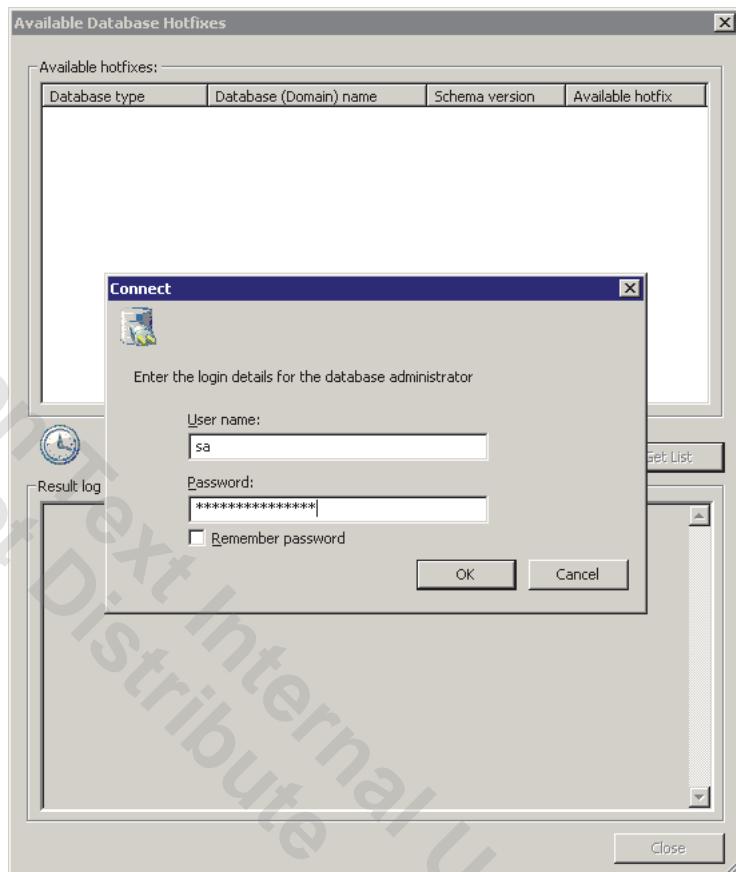
### **View a List of the Repositories at the Site that Includes Schema Versions and Hotfixes**

1. Right-click the site node and select **Available Database Hotfixes**.
2. Click **Get List**.

The Connect dialog is displayed.

3. Enter the login details for the database administration user and click **OK**.

**Figure 7-23:**  
**Schema Versions and Hotfixes**



A list of the repositories with hotfix information is displayed.

4. Review the information and click **Close** to close the dialog.

## Exercise 3 – Add a New User

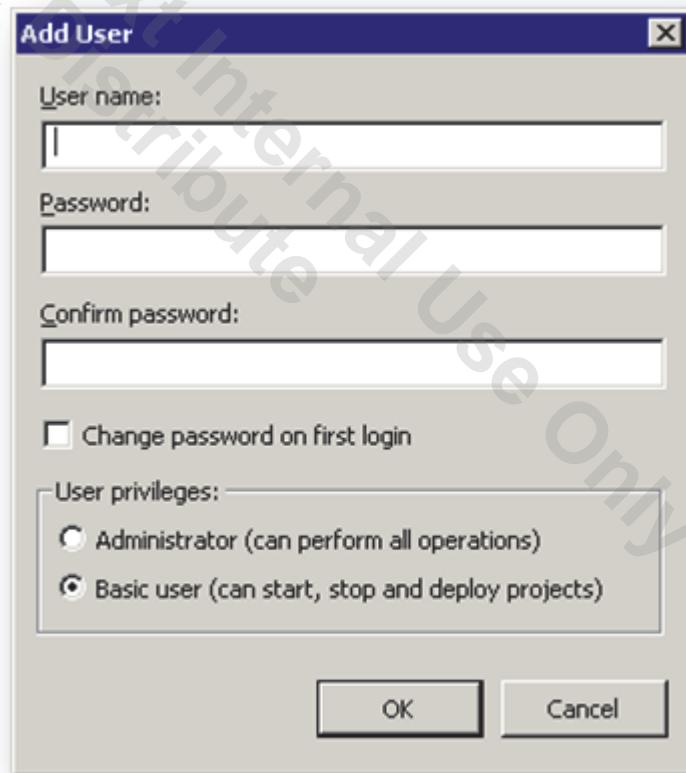
In this exercise, you will add a new management gateway user who is assigned the Basic role. The following steps are required:

- Add a new user who is assigned the Basic role.
- Exit Control Center, re-open Control Center and log on as the new user.

-  **Create a User**
1. In Control Center, right-click the host node and select **Manage Users**.  
The Manage Users dialog opens.
  2. Click the **Add** button.

Figure 7-24:

Adding a User



The Add User dialog is displayed.

3. Enter a user name and password for the user.
4. Select **Change password** on first login and the **Basic** role.
5. Click **OK** to close the Add User dialog, and **OK** again to close the Manage Users dialog.
6. In Control Center, select **File > Exit** to close Control Center.
7. Open **Control Center** again.
8. Right-click the site node and select **Connect**.

The Connect dialog opens.

9. Enter the user name and password for the new user and click **OK**.

A message is displayed informing you to change your password.

10. Enter and confirm a new password and click **OK**.

You are connected to Control Center as the new user. You can now explore the site to see which options are available to Basic users.

You now need to connect as the Administrator to continue with the next exercises.

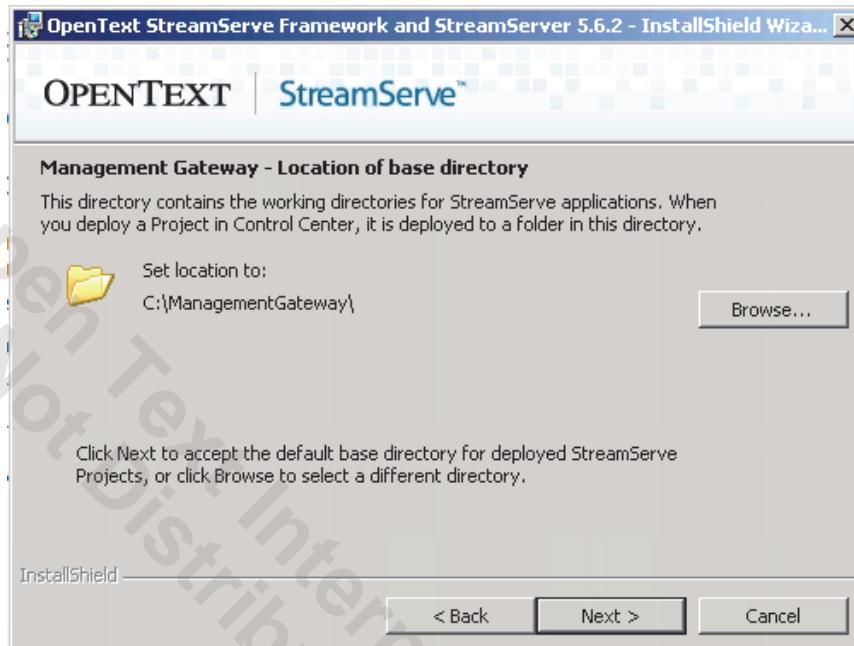
11. Right-click the site node and select **Disconnect**.
12. Connect again as the Management Gateway Administrator (**Administrator/streamserve**).

## Directory Structure for StreamServe Applications

### Base Directory for Projects

Figure 7-25:

Base Directory for Projects



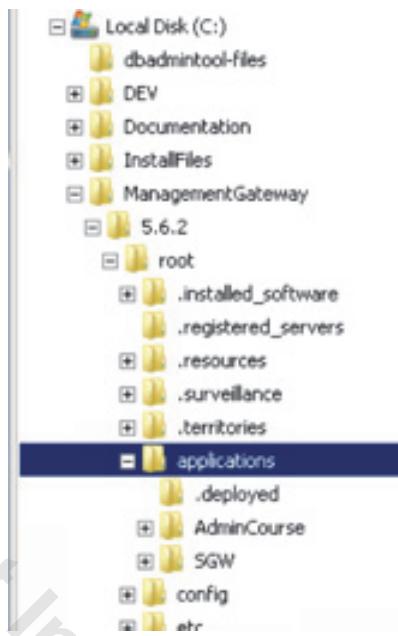
During the installation of the StreamServe software, a base directory is created for all StreamServe Projects that are deployed in Control Center.

The default path for the base directory is:

<sysdrive:>\ManagementGateway\5.6.2\root\applications\

## Working Directory

Figure 7-26:  
Working Directory



Each StreamServer application has a working directory in the base directory.

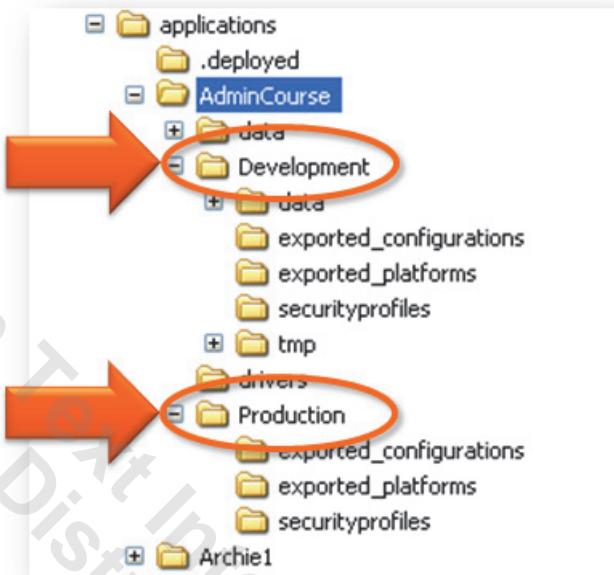
When a Project is deployed in Control Center, a subdirectory is created in the applications folder of the base directory.

The name of the subdirectory is the same as the name of the StreamServer application that is specified in Control Center.

If relative paths are specified for the connectors in the Project, the paths are relative to the working directory.

## Working Directories for Physical Platform Layers

Figure 7-27:  
Working Directories for  
Physical Platform Layers

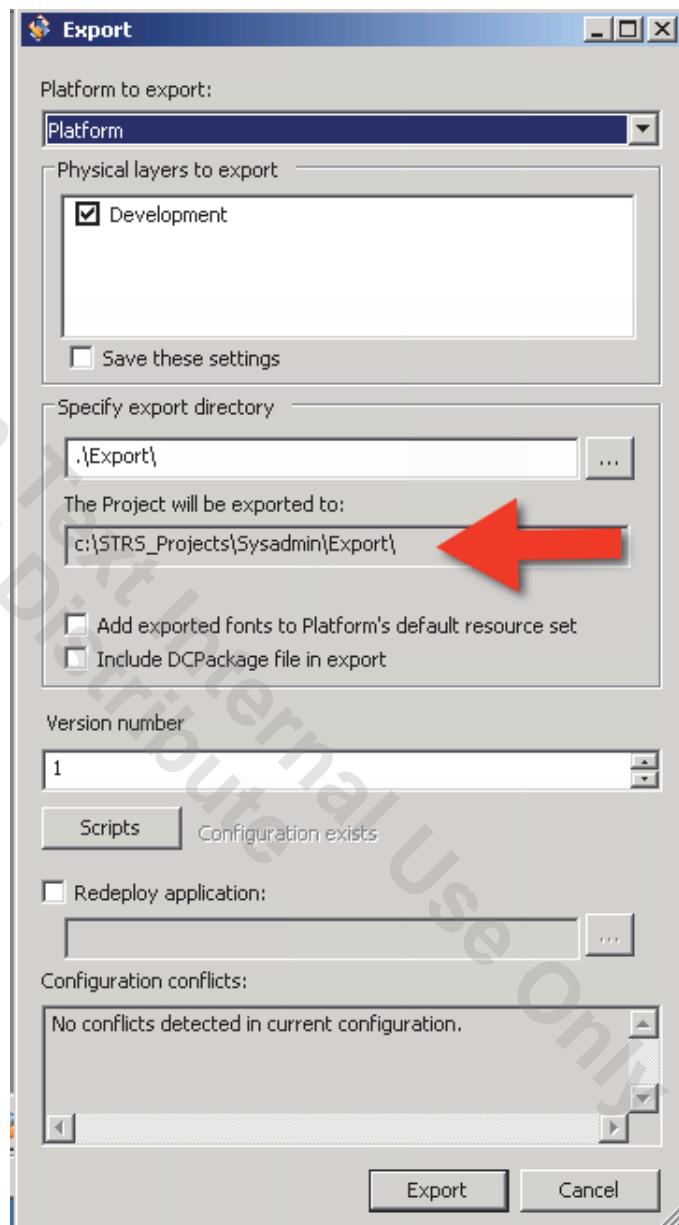


For each physical platform layer in the deployed Project, a working directory is created.

The image above shows a StreamServer application named AdminCourse that has two physical platform layers. The layers are named Development and Production, which are the working directories.

### Export Directory

Figure 7-28:  
Export Directory



The configuration files that are required to run StreamServer are stored in an export directory. The path to this directory is specified when the Project is exported from Design Center.

Open Text Internal Use Only  
Do Not Distribute

## 8. Scalability and Performance

On completion of this (section/module), participants should be able to:

- Identify the technical aspects that affect availability of StreamServe services
- Describe the trade-off between performance and fault tolerance
- Implement Profiling
- Implement procedures to improve high availability, fault tolerance and performance

### Scalability Concepts

High availability is about maximizing system availability and uptime, while masking and minimizing system downtime or component failure. It is about keeping systems up and running 24 hours a day, 7 days a week with a reasonable amount of performance, despite partial failure.

In StreamServe installations, high availability and fault tolerance is achieved by using redundant Communication Servers and clustered repositories.

#### Availability

Availability is the degree to which a service is functional and accessible to a user. A high-availability system ensures that, even in case of partial loss of function, no client will suffer loss of service.

There are a number of methods to increase service availability, for example fault tolerance and clustering. Both these methods are based on different degrees of redundancy and failover functionality that enable moving services from one machine to another in case of a failure.

#### Fault Tolerance

Fault tolerance is the extent to which a system or service will continue to operate at a given performance level, even when one or more of its components are malfunctioning.

Fault tolerant systems are designed to keep running even after a fault has occurred. A fault tolerant system might also ensure that no matter what happens, no data is lost.

Possible downtime causes can include hardware or software failure, environmental conditions, operator error, or planned maintenance and upgrades.

#### Scalability

Scalability is a measure of how well a system will work when the size of the installation increases. For example, a central server may perform adequately with ten clients, but with a thousand clients it might fail to meet performance requirements. It may even malfunction and cause downtime.

A scalable system allows you to gradually increase system capacity. This means that you can expand the system as requirements on performance and availability increase instead of buying a fully-scaled system at once.

Scalability is achieved by allowing added hardware to efficiently take up part of the system's workload. Two schemes are commonly used: Load balancing and load distribution.

**Load Balancing and Load Distribution** Load balancing means that the work load is dynamically distributed between available servers, or service providers, depending on the appreciated workload.

Load distribution means that the work load is distributed between available services in a static manner, e.g. every other input job is sent to each one of two servers for processing.

If combined with functionality that reacts to a service going down, load balancing or load distribution can help provide increased availability.

A group of servers that use load balancing or load distribution is called a server farm or a scale-out cluster. Each server is capable by itself to fulfill the functional demands of the service and does its allotted job without knowing or caring what its companions are doing.

**Clusters** A cluster, also referred to as fail-over cluster, is a group of servers and other resources that act like a single system and enable high availability and load balancing. If one server in the cluster goes down, the other servers are prepared to take over operation. To the outside world, the cluster appears to be a single highly available system.

A high availability cluster usually consists of two or more machines, a set of private network interfaces, one or more public network interfaces, and a number of shared disks. A data service can be moved from one machine to another in this configuration, enabling continued access to the service even if one machine fails.

The machines in the cluster communicate and send control messages to each other over the private network interfaces, while the public network interfaces are used for communicating with the clients of the cluster. The disks are shared by one or more machines in the cluster, so that if one machine fails, there is at least one machine that has access to the disk. When clients connect to a service running on the cluster, they connect to the logical network interfaces of the service only. In case of failure, this allows the service to be moved to another machine along with its logical network interfaces.

In order to survive any single point of failure, all parts in the cluster should be redundant, including the machines themselves.

The servers that comprise a cluster are referred to as nodes. All servers in a cluster continuously monitor one another in order to respond correctly in case of a failure. Clusters can be configured in different ways. For example, if you have a cluster with two nodes, you can have one node idle, ready to take over operation if the other node goes down. This is referred to as a Hot/Cold configuration. You can also have a Hot/Hot configuration, which means both nodes are online at the same time. Hot/Cold configurations often involve lower license costs, but it also means that hardware is not fully utilized.

## Performance Overview

By default, StreamServe software is designed to ensure well-rounded overall performance related to operating systems, databases, job throughput, response times, and memory consumption. However, depending on your specific environment and configurations, you may still experience performance issues and system bottlenecks.

**Key Points** Some key points to consider when tuning and investigating performance issues in StreamServe include:

- Find out which optimization factors your company prioritizes
- Be aware of factors in your environment that affect performance
- Take a holistic view when investigating performance issues and what may be causing them
- Configure StreamServe to use less resources than are available in your environment
- Continuously monitor and tune your environment

**Prioritization of Optimization Factors** Typically certain factors are optimized at the expense of others when tuning. For example, increasing cache sizes may improve throughput and response times, but at the cost of memory consumption. Compressing files may save disk space, but have a negative impact on performance.

Before tuning your StreamServe environment, you must find out which optimization factors your company prioritizes.

<b>Environmental Factors Affecting Performance</b>	<p>The primary determinant of performance is the configuration and design of the Design Center Project. In general, the more complex configuration, the longer the processing time.</p> <p>Besides the Project configuration, many other factors also affect performance. Some examples include:</p> <ul style="list-style-type: none"><li>• The operating environment and hardware. Including other applications running in the same environment.</li><li>• The format, type, and quantity of input data.</li><li>• The number of concurrent users.</li><li>• How the input data and number of users vary over time.</li></ul> <p>You must be aware of these performance factors when profiling and tuning the StreamServe environment.</p>
<b>Identifying Performance Issues</b>	<p>If a bottleneck exists in your environment, it can significantly affect the performance of other factors. For example, network issues can cause StreamServer applications to consume too much memory. This can make it difficult to identify the root cause of a particular performance issue.</p>
<b>Tuning Based on Available Resources</b>	<p>When tuning StreamServe, you should not configure StreamServe applications and components to use more resources than are available in your environment. For example, the number of threads configured for a StreamServer application should be less than the total number of CPU cores available.</p> <p>To avoid bottlenecks, we recommend that you configure StreamServe to use just less than the available resources. Avoiding bottlenecks is important as a bottleneck in one area can lead to a significant performance issue in another area.</p>
<b>Continuous Monitoring and Tuning</b>	<p>Tuning a StreamServe environment is an ongoing process.</p> <p>You must continuously monitor the environment to ensure that the system remains within the required performance range. You must identify problems and trends over time and tune the environment accordingly.</p>

## Profiling the StreamServe Environment

When profiling, you investigate how the StreamServe software behaves during processing. The purpose is to determine the most critical performance bottlenecks and the sections that may need optimization.

Profiling is an iterative process that should be done throughout all implementation phases for the Design Center Project. It needs to be revisited as the Project and the requests evolve.

After you reach a satisfactory level of performance in your test environment, you can save the profile logs as a baseline. Then, if you experience problems in production, you can compare the logs to investigate where problems exist.



When profiling, you should always strive to use real data. In a test environment, you should use data and volumes that resemble production data as closely as possible.

Since logging affects performance, it is recommended to minimize the logging.

---

**StreamServe Profiler Service** Most likely, your company already uses third-party profiling tools and platform specific commands to profile the environment.

As a complement, you can use the StreamServe Profiler service for the StreamServe related parts. The Profiler service enables you to monitor and measure times for the following:

- Processing events, such as collecting, preprocessing, and processing data.
- Database invocations to and from the StreamServe repositories.
- Web service requests to and from the service gateway.
- Cache service operations (mainly for ADEP Designer Processes).

## Enabling the Profiler Service

You must enable the Profiler service for each application (StreamServer, Archiver, Task Scheduler, or service gateway) that you want to profile.

The Profiler service handles all the profile data and controls how the profile output is presented. If you do not change the default settings, the Profiler service creates a file in the working directory called profiler.data, which contains the profiler output.



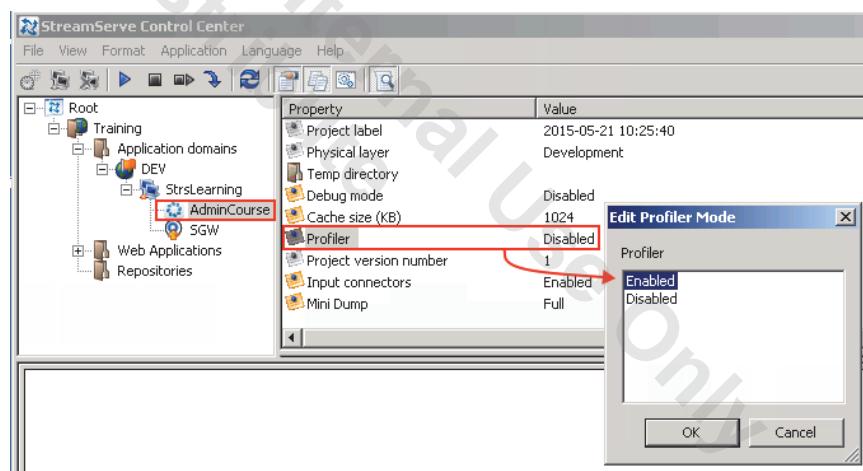
Profiling affects StreamServe processing performance.

In a production environment, you should disable the Profiler service.

To enable the Profiler service:

1. In Control Center, select the application to profile.
2. In the Properties view, enable the Profiler property.

**Figure 8-1:**  
**Enabling Profiler**



3. Restart the application.

## Configuring the Profiler Service

You can update the default configuration for the Profiler service in the profilerservice.xml file. Some examples of the changes you can make include:

- Specify a profile provider that suits an external tool that you want to use to analyze the profile output.
- Configure filters to make the output less comprehensive.

**Common or Application-Specific File** You can either:

- Update the common profilerservice.xml file, applicable for all profile-enabled applications.
- Copy the file to the working directory of a specific application and apply updates to this application only (recommended).

To configure the Profiler service:

1. Open the profilerservice.xml file.
2. Configure the profile providers.
3. Configure the filters.
4. Save and close the file.
5. Restart the application(s).

### Event Properties

Task	Description
%1	The UID (Unique IDentifier) for the event in numeric format.
%2	The UID for the event in string format.
%3	The UID for the event namespace in numeric format.
%4	The UID for the event namespace in string format.
%5	The context of the event.
%6	The ID of the thread that generated the event.
%7	The timestamp when the event was executed.
%8	The elapsed time for the event (in milliseconds).

**Profile providers** A profile provider provides an interface between the Profiler service and the tool in which you intend to analyze the profile output. By default, a profiler.data file is generated in the working directory of each profiler-enabled application.

**Properties and Delimiters** A number of properties are available for each profiled event. By default, all these properties are presented in the profile output, using commas (,) as delimiters.

In the profilerservice.xml file, you can use the messageformat attribute to limit the properties in the profile output and to specify other delimiters.

**Example of Profile Provider Configuration** In this example, a profiler.data file is generated in the working directory. In the file, a limited number of properties are presented and # is used as delimiter.

```
<provider value="http://schemas.streamserve.com/uid/component/fileprofilerprovider/1.0">
    <configuration>
        <fileprofilerprovider xmlns="http://schemas.streamserve.com/uid/component/fileprofilerprovider/1.0">
            <provider name="http://schema.streamserve.com/uid/resource/fileprofilerprovider/logfile/1.0"
                messageformat="%2#%4#%5#%6#%7#%8">
                <configuration>
                    <providers xmlns="http://schemas.streamserve.com/uid/manager/logmanager/1.0">
                        <provider name="http://schema.streamserve.com/uid/resource/fileprofilerprovider/logfile/1.0"
                            type="http://schema.streamserve.com/uid/component/filelogprovider/1.0"
                            default="no" source="profiler.data"
                            codepage="UTF-8" msginfo="0" sizelimit="10"
                            savepath=". " delete="no" asynchronous="yes"/>
                    </providers>
                </configuration>
            </provider>
        </fileprofilerprovider>
    </configuration>
</provider>
```

## Filters

By default, all measured profile events are presented in the profile output, which results in a comprehensive list. You can apply filters to limit the number of events that are passed down to the profile provider and included in the profile output.

**Namespaces** You can filter on the namespaces and their sub-events.

**Namespace Filter Syntax** In the profilerservice.xml file, you can add one or more filter elements, each containing one or more namespace filter configurations. Use the following syntax to combine the appropriate filters:

- **include** – Includes information from the namespace in the profile output. Use \* to include all namespaces, except the ones excluded.
- **exclude** – Excludes information from the namespace in the profile output. Use \* to exclude all namespaces, except the ones included.

**Example of Filter Configuration** The filter below excludes information in the profile output about detailed events relating to StoryTeller processing.

```
<filters>
    <filter type="http://schemas.streamserve.com/uid/resource/
        profilerservice/namespacefilter/1.0">
        <configuration>
            <namespacefilters xmlns=
                "http://schemas.streamserve.com/
                uid/resource/profilerservice/namespacefilter/1.0">
                <namespacefilter type="exclude">
                    streamserve.notification.storyteller.profiler.
                    profilerevent.timer.detailed</namespacefilter>
                <namespacefilter type="exclude">
                    streamserve.notification.storyteller.profiler.
                    profilerevent.value</namespacefilter>
            </namespacefilters>
        </configuration>
    </filter>
</filters>
```

## Optimizing StreamServe Processing

The complexity and configuration of the Design Center Project affects the StreamServe processing time. In general, the more complex configuration, the longer the processing time.

For example, for a StoryTeller Process, the size and complexity of the document design (fonts, images, business graphics, scripted layout rules, etc.) affect the processing time. When the document is delivered to the output destination, the format and structure of the output data, including the driver type and driver settings impact performance. Any handling of variables, metadata, post-processing, and ODBC database calls are added to the overall processing time.



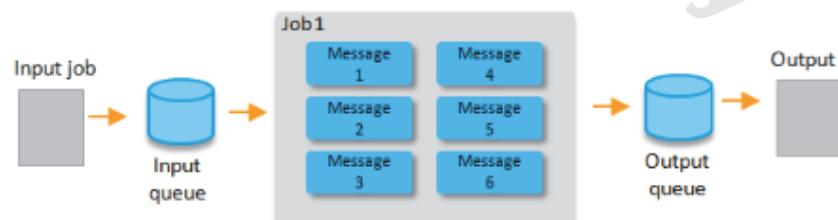
Always consult the OpenText StreamServe Online Help for information about how a certain configuration or function affects performance.

You can create clones of your site or application domains and use these clones to try out the most optimal settings for your specific environment. Using clones enables you to try out the settings in different sets of pre-configured applications domains.

## Optimizing Performance with Job Scaling

By default, StreamServer applications use a single thread to process each job. You can use job scaling to split the processing of the Messages in one job over several threads. By using job scaling, StreamServer applications can process several Messages in parallel and better utilize your available hardware.

**Figure 8-2:**  
**A Job Processed by One Thread**



**Figure 8-3:**  
A Job Processed Using Job Scaling with Three Threads



### Drawbacks of Using Job Scaling

When you use job scaling, the StreamServer application splits the Messages in each input job into separate jobs, the number of which depends on how you configure job scaling. If you want to combine the output into a single file, you can use Document Broker Plus and create a query to combine the relevant documents.

### Recommendations

- The number of threads you configure for job scaling should be feasible for the available hardware.
- In scenarios where queues are used, job scaling uses the same thread pool as the input queue. When you configure the number of threads available for job scaling, you must consider both the hardware available and the number of threads available for the input queue. If you use a single StreamServer application, we recommend the number of threads available for the input queue is always greater than the number of threads used for job scaling. For information about configuring the threads for the input queues, see Design Center documentation.

## Optimizing XMLIN Events Creating a Thread Pool for XMLIN Events

If you use XMLIN to process large XML documents, you may improve performance by configuring the `-evecollectmt` startup argument. When configured, a thread pool is created that XML Events use for field pattern matching and value extraction. This speeds up the creation of fields as many parts of a document can be processed at the same time. This argument only has an effect if Document or Message collection mode is used, otherwise regular sequential field matching is used.

**Startup Argument** `-evecollectmt <num>`

where `<num>` is a number of threads that will be used for parallel processing of matching patterns for Events.

You can reduce the memory for XML Events by using the `-xmlwhitespacenodes` and `-xmlunusedattributes` startup arguments.

<b>xmlwhitespacenodes</b>	When you set up this startup argument with the ignore option, the XMLIN parser ignores nodes containing only white spaces, which reduces the size of the node tree and the memory used by the Event. For information about the limitations and syntax of this startup argument, see Startup arguments documentation in OpenText StreamServe Online Help.
<b>xmlunusedattributes</b>	When you set up this startup argument with the ignore option, the XMLIN parser ignores attributes that are not included in XPath patterns or expressions and excludes these attributes in the DOM tree. For information about the limitations and syntax of this startup argument, see Startup arguments in OpenText StreamServe Online Help.
<b>XPath performance Considerations for XMLIN Events</b>	<p>The XMLIN tool is optimized to match events and fields according to patterns that match element or attribute names. Extensive use of string comparisons in match pattern predicates, for example by using /data/fields/field[@name='Field1'], can have a negative impact on performance in some cases. The reason for this is that an XMLIN Event iterates through all XML nodes in document order, and for each node in the document it handles, the Event goes through every possible match in the list of field matches. If an Event configuration has many patterns similar to the example above, it will lead to a lot of string comparisons.</p> <p>In some cases, the number of string comparisons can be reduced by using a block and matching the name and value separately as fields in the block, for example / data/fields/field/@name and /data/fields/field. If this is not possible, you could also consider solutions such as transforming the data before it reaches the XMLIN Event or making changes in the system that generates the data.</p>

## Tuning the Threads for the Input and Output Queues

You can configure the number of threads used by the input and output queues, which allows a StreamServer application to process several jobs in parallel.

By default, the input queue is configured to use one thread and the output queue is configured to use a maximum of 10 threads. On the output queue, each thread writes one output job to the output queue.

The default settings are optimized for StreamServer applications which receive input jobs that generate many output jobs.

- Recommendations**
- The number of threads for the input queue should be less than or equal to the number of CPU cores available.
  - The number of threads for the output queue should be more than or equal to number of threads for the input queue. This enables the StreamServer application to process jobs in parallel throughout all job phases.
  - If you run several StreamServer applications on the same machine, you should split the threads appropriately. For example, if there are 16 CPU cores available to run two StreamServer applications, eight threads could be allocated to each StreamServer application - four for the input queues and four for the output queues.
  - If an average input job creates too much load on the output queue or if several highly loaded output queues are handled simultaneously, you might increase database performance and job throughput by reducing the default number of asynchronous requests.
  - If most of the input jobs generate a single output job, we recommend to disable asynchronous queuing. By disabling asynchronous queuing, the output jobs are processed and stored in the output queue one after another. Note, that this does not guarantee the order in which documents are delivered.

**Configuring the  
Threads for the Input  
and Output Queues**

- The number of threads for the input queues is configured in the Manage Queues dialog box in Design Center. For more information see the Design Center documentation.
- The number of threads for the output queue is configured using the `-asynchronousqueue` startup argument. You disable asynchronous queuing by using the `-disableasynchronousqueuing` argument.

## Enabling Asynchronous Input Directory Scanning

By default, a Directory input connector scans the input folder synchronously. The connector runs in single-threaded mode and retrieves the files one by one from the input folder.

You can disable synchronized scanning and instead scan the input folder asynchronously. The input connector then runs in multi-threaded mode and retrieves several files in parallel.



Running asynchronously means that several files are pushed into the input queue in parallel, which increases the load on the queue.

The runtime repository must be able to handle the extra load. If not, asynchronous scanning may result in resource contention and lower job throughput.

### Recommendations and Tips

- If you expect many small input jobs, you may increase the job throughput if you run the input connector asynchronously.
- Running one input connector asynchronously is often more efficient than running multiple input connectors synchronously.
- When not to use asynchronous directory scanning
- If you expect large batches of input jobs, delivered to the input folder one by one, asynchronous scanning will not enhance performance.
- If you use Microsoft SQL Server and use more than one Directory input connector in the application domain, asynchronous scanning may not enhance performance. For example, if you share queues between several StreamServer applications and the hardware or the database server cannot handle the extra load of the asynchronous scanning.

## Storing Queued Documents on Disk

Documents in input and output jobs are by default stored as blobs (Binary Large Objects) in the queues in the runtime repository.

For each queue, you can choose to store the blobs as files on disk instead. The runtime repository then includes references to the blobs instead of the actual data.

- Recommendations**
- If you expect large blobs, you might increase job throughput and decrease resource consumption by storing the blobs on disk. The cost of the extra disk space must be weighed against the resource consumption of the StreamServe software and the database software.
  - If you store blobs on disk, we recommend to back up the blobs.

- When Not to Store Blobs on Disk**
- If several StreamServer applications share queues.
  - If you want to use StreamStudio Reporter to resend or delete jobs.
  - If you want to use StreamStudio Collector.

## Disabling Sorting for the Queues

By default items in the queues are sorted to control the order in which the items are retrieved from the queue. You can disable sorting of the queues, which almost always provides performance improvements.

You use the `-disablesortedqueuing` startup argument to disable sorting.

## Tuning the Service Queue

If the Design Center Project includes service-enabled Messages or Service Request input connectors, a service queue is used as an input queue. By default, this is a shared queue for which scheduled spooling is enabled.

Both the StreamServer application and the service gateway access the service queue.

### Tuning the Service Queue for the StreamServer Application

Each service request can be seen as a separate job stored in the service queue. The StreamServer application polls the service queue and allocates one thread to each job. By default, the StreamServer application polls for new jobs every second and can allocate a maximum of five threads, allowing five jobs to be processed in parallel.

You can edit the polling interval and the number of threads used by the application in the repositorymanager.xml file for the StreamServer application.

**Recommendations** If you expect a large number of concurrent service requests, you might:

- Enhance formatting performance by increasing the number of threads and editing the default spooling interval. Each thread consumes system resources.
- Avoid performance bottlenecks related to the runtime repository by setting up several StreamServer applications that poll the same service queue.

To tune the service queue for the StreamServer application:

1. Open the repositorymanager.xml file for the application.
2. Edit the following elements:

```
<queues>
  <servicequeue>
    <queueservice> ... </queueservice>
    <resourceservice> ... </resourceservice>
    <schedule>T II * * S * * 1</schedule>
    <threadpool min="5" max="5" />
  </servicequeue>
</queues>
```



You must use the same value for the minimum and the maximum number of threads in the thread pool.

For example, `<threadpool min="10" max="10" />`

- 
3. Save and close the file.
  4. Restart the StreamServer application.

## Tuning the Service Queue for the Service Gateway

The service gateway polls the service queue for completed jobs.

The time interval at which the service gateway polls the service queue (by default, 500 ms) is specified in the repositorymanager.xml configuration file for the service gateway.

- Recommendations**
- If the processing time for the StreamServer application is less than 500 ms, you may enhance performance by decreasing the time interval at which the service gateway polls the service queue.
  - If you expect a large number of concurrent service requests, you may avoid performance bottlenecks related to the runtime repository by:
    - Increasing the time interval at which the service gateway polls the service queue.
    - Decreasing the maximum number of threads to be used. For example, to 5 threads.

To tune the polling interval and the number of threads:

1. Open the repositorymanager.xml file for the service gateway.
2. Edit the time interval (ms) at which the service gateway application polls the service queue in the following element:

```
<!-- Remote service invocation. -->
<service ...> ...
  <configuration type="...">
    <executeRemoteService xmlns="...">
      <checkResultFrequency>500</
      checkResultFrequency>
      <onTimeOut> ... </onTimeOut>
      <accessServiceName> ... </accessServiceName>
    </executeRemoteService>
  </configuration>
</service>
```

3. Add the following element to the <executeRemoteService ...> element to specify how many concurrent threads to be used when checking the job status:

```
<concurrentStatusThreads><Number_Of_Threads></
  concurrentStatusThreads>
```

Where <Number\_Of\_Threads> is the number of threads to be used.  
For example, 5

4. Save and close the file.
5. Restart the service gateway.

## Optimizing StreamServe Repositories

Database performance is largely dependent on the tuning and maintenance of the StreamServe repositories. For example, by indexing columns in the runtime repository and in the StreamServe archive, you may improve the speed of the data retrieval operations. By optimizing the way in which expired top jobs are deleted from the runtime repository, you may improve the delete performance.

## Tuning the Database Connection Pools

A StreamServe application uses a database connection to communicate with the database server. The application takes an available connection from a connection pool, executes the database commands, and returns the connection to the pool.

By default, each application allocates a number of connections. We recommend to tune these default connections to fit the actual conditions.

For example, you may increase throughput by increasing the number of database connections in the connection pool for a StreamServer application. While doing so, you must be aware of the consequences. For example, allocating too many connections may have negative impact on the database performance, because you would have potential waits due to contention.

Any tuning of the database connection pools affects performance.

## Database Connection Pools - StreamServer

A StreamServer application uses a connection pool to access the runtime repository. Archiver applications use connection pools to access the StreamServe archive, OpenText Enterprise Library Services(ELS), OpenText Archiver Server, and external archiving systems connected to SAP.

By default, the maximum number of connections in the connection pools for the runtime repository and StreamServe archive are set to 25, allowing temporary growth if needed. You can edit the number of connections in the repositorymanager.xml file for each application.

The number of opened connections for ELS, an external archive connected to SAP, or Archive Server is limited by the number of threads used by Archiver application. The default value is 10, you can change the number of threads in the archiver.config.xml in the <threads> elements.

**Recommendations for  
repositorymanager.xml**

- The maximum number of connections in the pool is high enough so that using temporary connections is avoided. Using temporary connections occasionally does not always affect performance, however opening and closing temporary connections does consume resources.
- The maximum number of connections should be 10 or more.
- If multiple threads are configured for the queues in the Design Center Project, you may increase throughput by increasing the number of database connections for the StreamServer application.



Allocating too many connections may have negative impact on the database performance, because you would have potential waits due to contention.

- If a background event is not used (for example, if you run several jobs and configure the event for the database instead), you can remove the background event from the repositorymanager.xml file and decrease the number of database connections.

To tune the connection pool in repositorymanager.xml:

1. Open the repositorymanager.xml file for the application.
2. Edit the following element:

```
<objectpools>
  <objectpool type=...>
    <configuration>
      <objectpoolex xmlns=...>
        <general min="0" max="25"
          allowtemporaryobjects="true" />
      </objectpoolex>
    </configuration>
  </objectpool>
</objectpools>
```

3. Save and close the file.

## Database Connection Pool – Management Gateway

Control Center connects to StreamServe Enterprise Repository via the management gateway. The management gateway uses a connection pool to access the enterprise repository.

By default, the maximum number of connections in the pool is set to 25, allowing temporary growth if needed. You can edit the number of connections in the mgw-repositorymanager.xml file.

- Recommendations**
- We recommend to deny temporary growth (that is, change the allowtemporaryobjects="true" attribute to "false").
  - If the number of Control Center users that access the management gateway simultaneously (deploying Design Center Projects, starting and stopping applications, etc.) is low, you can decrease the number of database connections.
  - The maximum number of connections should be 5 or more.

To tune the connection pool in mgw-repositorymanager.xml:

1. Open the mgw-repositorymanager.xml file.
2. Edit the following element:

```
<objectpools>
  <objectpool type=...>
    <configuration>
      <objectpoolex xmlns=...>
        <general min="0" max="25"
                  allowtemporaryobjects="true" />
      </objectpoolex>
    </configuration>
  </objectpool>
</objectpools>
```

3. Save and close the file.

## Database Connection Pools – Service Gateway

The service gateway handles web services requests from clients to the StreamServe repositories. For example, from the StreamStudio portal to the runtime repository and the StreamServe archive, and from the SSSP application, to the runtime repository and the Document Broker repository.

You can edit the number of database connections for the service gateway in the following files:

- repositorymanager.xml for the service gateway.
- securitymanager.xml.

**Connection Pool in repositorymanager.xml** By default, the maximum number of database connections in the pool is set to 25, allowing temporary growth if needed.

- Recommendations**
- We recommend to deny temporary growth (that is, change the allowtemporaryobjects="true" attribute to "false").
  - The maximum number of connections should be 10 or more.

To tune the connection pool in repositorymanager.xml:

- Open the repositorymanager.xml file for the service gateway.
- Edit the following element:

```
<objectpools>
  <objectpool type=...>
    <configuration>
      <objectpoolex xmlns=...>
        <general min="0" max="25"
          allowtemporaryobjects="true" />
      </objectpoolex>
    </configuration>
  </objectpool>
</objectpools>
```

- Save and close the file.
- Restart the service gateway.

## Connection Pool in securitymanager.xml

The service gateway can access the runtime repository in order to validate StreamStudio settings, for example StreamStudio user roles. When accessing the runtime repository for such reasons, the service gateway uses a dedicated connection pool.

By default, the maximum number of connections in the pool is set to 20, allowing temporary growth if needed. You can edit the number of connections in the securitymanager.xml file.

- Recommendations**
- We recommend to deny temporary growth (that is, change the `allowtemporaryobjects="true"` attribute to `"false"`).
  - If the Design Center Project includes script functions for checking users, or if the Project is configured to validate senders and/or receivers, you can tune the connection pool in accordance with the number of concurrent StreamStudio users. For example, if you expect few users, you can decrease the number of database connections.
  - The maximum number of connections should be 10 or more.

To tune the connection pool in securitymanager.xml:

1. Open the securitymanager.xml file.
2. Edit the following element:

```
<objectpools>
    <objectpool type="http://
schemas.streamserve.com/
uid/component/objectpoollex/1.0"
name="http://
schemas.streamserve.com/uid/resource/
connectionpool/repository">
        <configuration>
            <objectpoollex xmlns=...>
                <general min="0" max="20"
allowtemporaryobjects="true" />
            </objectpoollex>
        </configuration>
    </objectpool> ...
</objectpools>
```

3. Save and close the file.
4. Restart the service gateway.

## Resetting (Flushing) a Database Connection Pool

Once a database connection has established, the connection stays live and tied to the connection pool as long as the application is running. When you restart the application, the connection pool is reset to its initial state (to the minimum number of connections specified).

You can schedule resetting of the connection pool, and thereby enable the pool to be reset without restarting the application. You schedule resetting of the connection pools separately for each application.

- Recommendations**
- If you experience that too many database connections are tied to a connection pool, performance might be enhanced if the pool is reset.
  - We recommend that you schedule resetting of the pool to a time period when the job throughput is low. For example, after scheduled batch jobs.
  - Resetting of the pool must be scheduled in conjunction with the maximum number of connections in the pool and the temporary growth.

To schedule resetting of a connection pool:

1. Open the file in which you tune the database connection pool.
2. Update the objectpool and objectpoolex attributes as described below.
3. Add the flushschedule="*<Schedule>*" attribute as described in the example below and configure the time schedule:

```
<objectpools>
    <objectpool type="http://schemas.streamserve.com/
uid/component/objectpoolex/2.0"
name="http://schemas.streamserve.com/uid/
resource/
connectionpool/repositories">

    <configuration>
        <objectpoolex xmlns="http://schemas.streamserve.com/
uid/component/
objectpoolex/2.0">
            <general min="0" max="25"
allowtemporaryobjects="true"
flushschedule="T II * * H * * 2"/>
        </objectpoolex>
    </configuration>
</objectpool>
</objectpools>
```

4. Save and close the file.
5. Restart the application.

## Increasing the Size of the Memory Buffer

During processing, a StreamServer application produces a number of temporary files. For example, files with blob data, stream caches, and settings for output jobs.

As long as the size of a single temporary file is less than the size of the memory buffer, the StreamServer application keeps the file in memory. If the file size exceeds the buffer size, the file is stored to disk.

By default, the size of the memory buffer (bufferKBSIZE) is 150 kB. The buffer pool is set to 1000. A buffer from the pool remains allocated after the temporary file is released and can be reused by other files. If the buffer pool is exhausted, temporary buffers, which are not reusable, are created.

**Prerequisites and Recommendations**

- If the sizes of the template files are larger than the buffer size, you may enhance performance by increasing the size of the memory buffer.
- The buffer size should be equal to (or slightly larger than) the size of the majority of the template files, rather than the maximum size of a single file.
- The buffer size times the buffer pool must not exceed the available memory.



The memory consumption of the buffers is added to the ordinary memory consumption of the StreamServer application execution process.

**Decreasing Memory Usage per Process**

To decrease the memory usage (provided there is enough memory available within the system), you may consider:

- Splitting up the execution process on several StreamServer applications.
- Decreasing the size of the Events in the input data.

To increase the size of the temporary buffer:

1. Open the iomanager.xml file for the StreamServer application.
2. Edit the following elements:

```
<manager type=...>
  <configuration>
    <iomanager xmlns=...>
      <temporarystream bufferKBSIZE="150">
        <objectpool type=...>
          <configuration>
            <objectpool xmlns=...>
              <general min="1" max="1000"
allowTemporaryObjects="true" />
            </objectpool>
          </configuration>
        </objectpool>
      </temporarystream>
    </iomanager>
  </configuration>
</manager>
```

3. Save and close the file.
4. Restart the StreamServer application.

## Optimizing StreamServe Web Applications

The StreamServe web applications include:

- StreamStudio web portal, which contains the StreamStudio web applications, such as Composition Center, Collector, and Reporter.
- SSSP application.
- Ad Hoc Correspondence and Correspondence Reviewer, which make their web service requests via the SSSP application.
- Document Broker Query (DBQ) Tool, which uses SSSP application to collect documents from a Document Broker repository.

The StreamServe web applications run on Java application servers. StreamStudio and the SSSP application access the related repositories using web services, which are hosted by the service gateway.

**Tuning the Thread Pool for the Service Gateway** The number of threads used by the service gateway for web service requests is configured in the threadmanager.xml configuration file for the gateway. Using several threads enables parallel processing of the service requests.

**Recommendations** If you expect a large number of concurrent service requests, you might enhance response times by increasing the number of threads.



Each thread consumes system resources.

---

To tune the thread pool:

1. Open the threadmanager.xml file for the service gateway.
2. Increase the number of threads in the following element:

```
<threadmanager ...>
    <dispatchqueues>
        <dispatchqueue ...> ... <dispatchqueue>
        <dispatchqueue type="http:
            /schemas.streamserve.com/uid/
            component/dispatchqueue/1.0"
            name="http://schemas.
            streamserve.com/uid/resource/
            iodispatchqueue/1.0">
            <configuration>
                <threadpool min="20" max="20" />
            </configuration>
        </dispatchqueue>
    </dispatchqueues>
</threadmanager>
```

3. Save and close the file.
4. Restart the service gateway.

**Tuning the Keep Alive Timeout for the Service Gateway** The service gateway can reuse one connection for several web service requests, meaning that a new connection does not have to be established for each request. This is especially important when running the SSSP application.

A connection is held open until the keep alive timeout (by default, 1 second) is reached. If there is no new inbound traffic during this time, the connection is closed down.

Prerequisites and recommendations:

- The clients accessing the service gateway must support keep alive.
- If the expected time periods between the service requests are larger than 1 second you should increase the keep alive timeout.



A timeout consumes the thread that keeps the connection open.

- If you expect a large number of concurrent service requests, you might benefit from disabling the keep alive timeout if one of the following applies:

- The clients are not handling keep alive correctly, that is are not reusing connections for subsequent requests.
- A large amount of different, individual, concurrent clients are used.
- You experience thread starvation.

To increase the keep alive timeout for the service gateway:

1. Open the semper.xml file.
2. Increase the timeout in the following element:

```
<soaplistener xmlns=...>
    <interface port="27180" name="*" backlog="100"
        reuseaddr="false"/>
    <timeout read="60" write="60"/>
    <keepalive max="100" timeout="1"/>
    <!-- <soapdebug .../> -->
    <!-- <trustedcommunicationchannel>...
        </trustedcommunicationchannel> -->
</soaplistener>
```

3. Save and close the file.
4. Restart the service gateway.

**Tuning the Response Timeout for the Service Gateway** At a web service request from a client, the service gateway waits a limited amount of time for the receiving service to complete processing and return a response.

If the timeout (by default, 60 seconds) is reached without any response, an exception is returned to the client and a failure is presented.

- Recommendations**
- If a StreamServer application requires more than 60 seconds (in general) to complete processing, you should increase the timeout.
  - The timeout should be set in conjunction with the retry settings configured in Design Center. You must consider the combination of output queue retries configured in the Platform, and any connection profile retries configured in the Profile configuration editor – for each output queue retry, the StreamServer application performs the specified number of connection profile retries.



An alternative to increasing the service gateway response timeout is to decrease the retry intervals configured in Design Center.

- The connection and socket timeouts on the client side should be set in conjunction with the increased timeout for the service gateway.

**Scenario** If a test email is sent from StreamStudio Composition Center, but the receiving mail server is unavailable, the StreamServer application that hosts the service retries to send the email according to the retry settings configured in Design Center.

If the service gateway timeout is reached before all retries are exhausted, a failure is presented in Composition Center. However, the StreamServer application might still succeed in sending the email in one of the remaining retries. In this scenario, you would benefit from either increasing the service gateway timeout or decreasing the number of retries in the connection profile settings.

To increase the response timeout for the service gateway:

1. Open the wsconfigurationservice.xml file.

2. Increase the timeout, which is specified in milliseconds, in the following element:

```
<data>
  <global>
    <parameters> ...
      <parameter name="remoteInvokeTimeOut"
                 value="60000"/>
    </parameters>
  </global>
  <local>...</local>
</data>
```

3. Save and close the file.
4. Restart the service gateway.

## Tuning Web Service Properties

You might enhance performance by tuning the web service properties for the StreamStudio portal and the SSSP application. The web service properties are defined in ws.properties files.

**Default Location of ws.properties – StreamStudio Portal** For example, for Apache Tomcat:

<TOMCAT\_HOME>\webapps\<Portal name>\WEB-INF\spring\properties

**Default Location of ws.properties – SSSP Application** For example, for Apache Tomcat:

<TOMCAT\_HOME>\webapps\sssp\WEB-INF\spring\properties

**Using an External Directory for Configuration Files** If you use a directory outside the portal root for web application configuration files, you must update the ws.properties files in this directory.

The path to the external directory is defined by a system environment variable (STRS\_WEB\_CONFIG\_DIR). For more information, see the Control Center documentation in OpenText StreamServe Online Help.

<b>Tuning Socket and Connection Timeouts</b>	<p>You can change the duration before a timeout occurs when the StreamStudio portal or the SSSP application connects to the service gateway. If the connection times out, an exception is returned to the client and a failure is presented.</p>
	<p>The socket timeout (ws.socket.timeout) limits the time to establish a connection to the service gateway. It also limits the time spent on reading the response from the service gateway. By default, this is 60 seconds.</p>
	<p>The connection timeout (ws.connection.timeout) is how long the connection is kept open, waiting for the service gateway to send more data. By default, this is 60 seconds.</p>
<b>Recommendations</b>	<p>Timeouts reduce the loss of processing resources consumed by idle connections. You should adjust the timeouts in conjunction with your environment (network capacity, bandwidth, etc.) and the expected size of the responses.</p>
	<p>To edit the connection and socket timeouts:</p>
	<ol style="list-style-type: none"><li>1. Open the ws.properties file in a text editor.</li><li>2. In the following properties, enter the new timeouts in milliseconds:</li></ol>
	<pre>#Property file for Web Service settings ws.connection.timeout=60000 ws.socket.timeout=60000</pre>
	<ol style="list-style-type: none"><li>3. Save and close the file.</li></ol>
<b>Load-Balancing Service Gateways</b>	<p>In Control Center, you can specify a primary and a secondary service gateway.</p>
	<p>By default, the secondary service gateway is only used if the primary service gateway is unavailable. If required, you can load balance the service gateways instead. The StreamStudio portal or the SSSP application then connects to the service gateway that was used on the last occasion.</p>
<b>Recommendations</b>	<p>If you add a secondary service gateway because the load on a single service gateway is too extensive, we recommend to load balance the gateways.</p>
	<p>To enable load balancing:</p>
	<ol style="list-style-type: none"><li>1. Open the ws.properties file in a text editor.</li><li>2. Change the ws.servicegateway.url.lb.roundrobin property to true.</li><li>3. Save and close the file.</li></ol>

<b>Tuning Timeouts for Failed Service Gateways</b>	<p>If the StreamStudio portal or the SSSP application fails to connect to a service gateway, this service gateway becomes unavailable for connection retries for a limited period of time.</p>
	<p>If there are two or more service gateways available, the ws.servicegateway.bad.reputation.long property specifies the time that a failed service gateway is unavailable for connection retries. By default, this is 5 minutes.</p>
	<p>If there is a single service gateway available, the ws.servicegateway.bad.reputation.short property specifies the time that a failed service gateway is unavailable for connection retries. By default, this is half a second (500 ms).</p>
	<p>To tune the timeouts for failed service gateways:</p>
	<ol style="list-style-type: none"><li>1. Open the ws.properties file in a text editor.</li><li>2. In the following properties, enter the new timeouts in milliseconds: ws.servicegateway.bad.reputation.long=300000 ws.servicegateway.bad.reputation.short=500</li><li>3. Save and close the file.</li></ol>
<b>Tuning Connections Pools for Service Gateways</b>	<p>Two connection pools are available when the StreamStudio portal or the SSSP application connects to the service gateway. If a pool is exhausted, any remaining requests are lined up in a queue.</p>
	<p>The ws.max.connections.per.host property specifies the maximum number of concurrent connections between the StreamStudio portal or the SSSP application and a single service gateway. By default, this is 10 connections.</p>
	<p>If there are two or more service gateways available, the ws.max.connections.in.total property specifies the maximum number of concurrent connections between the StreamStudio portal or the SSSP application and the service gateways. By default, this is 20 connections.</p>
	<p>To edit the service gateway connections:</p>
	<ol style="list-style-type: none"><li>1. Open the ws.properties file in a text editor.</li><li>2. Edit the following properties:</li></ol>
	<pre># Increase for more concurrent connections to SGW ws.max.connections.per.host=10 ws.max.connections.in.total=20</pre>
	<ol style="list-style-type: none"><li>3. Save and close the file.</li></ol>

## Tuning the Preview Cache for the SSSP Application

The SSSP application uses a separate preview cache to store document information from previews made in Ad Hoc Correspondence and Correspondence Reviewer.

The preview cache is configured in the ehcache.properties file using the following properties:

- `sssp.docache.maxElementsInMemory` – The maximum number of objects that can be cached. By default, this is 100 objects.
- `sssp.docache.timeToIdleSeconds` – How long these objects remain in the cache if they are not accessed. By default, this is 900 seconds.
- `sssp.docache.timeToLiveSeconds` – How long a new object remains in the cache after being created. By default, this is 900 seconds.

**Recommendations** If a large number of service requests are sent to the service gateway, you may reduce the response times by optimizing the preview cache. Increasing the maximum number of cached objects (`sssp.docache.maxElementsInMemory`) has largest impact on the response times.

Increasing the size of the preview cache consumes more memory, which may result in decreased performance.

To edit the cache for the SSSP application:

1. Open the ehcache.properties file in a text editor. The file is located in:  
`<Portal root>\sssp\WEB-INF\spring\properties`

For example, for Apache Tomcat:

`<TOMCAT_HOME>\webapps\sssp\WEB-INF\spring\properties`

2. Edit the following properties (primarily the first one):

`sssp.docache.maxElementsInMemory=100`  
`sssp.docache.timeToIdleSeconds=900`  
`sssp.docache.timeToLiveSeconds=900`

3. Save and close the file.

## Allocating Memory to StreamServe Web Applications

There must be enough memory allocated for the StreamStudio portal and the SSSP application on the JVM (Java Virtual Machine).

The amount of memory required depends on the current deployment scenario. For example:

- Deploying several StreamStudio web portals to the same Java application server requires more memory than deploying a single web portal.
- Deploying the Ad Hoc, Reviewer and SSSP applications to the same Java application server as the StreamStudio portal requires more memory than deploying these applications to a separate Java application server.

Deploying several applications to the same Java application server requires more memory, which may result in decreased performance.

- Recommendations**
- For JVM memory recommendations, see the following documentation in OpenText StreamServe Online Help:
    - StreamStudio > Recommended settings
    - Ad Hoc Correspondence and Correspondence Reviewer Administrator's Guide > Recommended settings
    - Document Broker Query Tool > Recommended settings
  - If you expect more than 200 concurrent service requests from Ad Hoc users, we recommend to install Ad Hoc Correspondence and the SSSP application on separate Java application servers.



Since the Ad Hoc application is restricted to only access data from the site where it is deployed, an HTTP web server must then be used as a front-end proxy between the Ad Hoc and the SSSP application

## Changing Java Virtual Machine Version

Applicable if Java HotSpot VM is used as Java Virtual Machine (JVM).

The better the Java Virtual Machine (JVM) performs, the better the Java application server performs.

By default, the Client version of the JVM is used by the StreamServe web applications. When running Java HotSpot VM, we recommend to use the Server version.

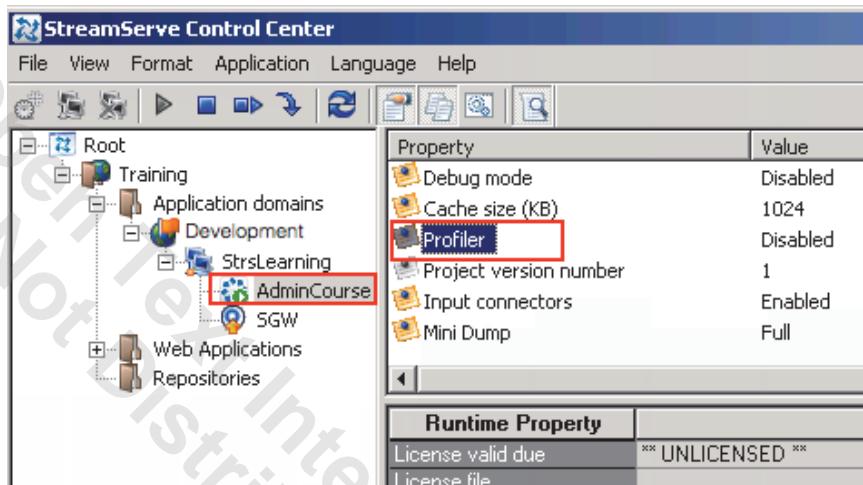
## Exercise 1 – Profiler Services



### Format the Profiler Output

1. In Control Center, click the **AdminCourse** application node.
2. In the Properties panel scroll down and double-click the **Profile** property.

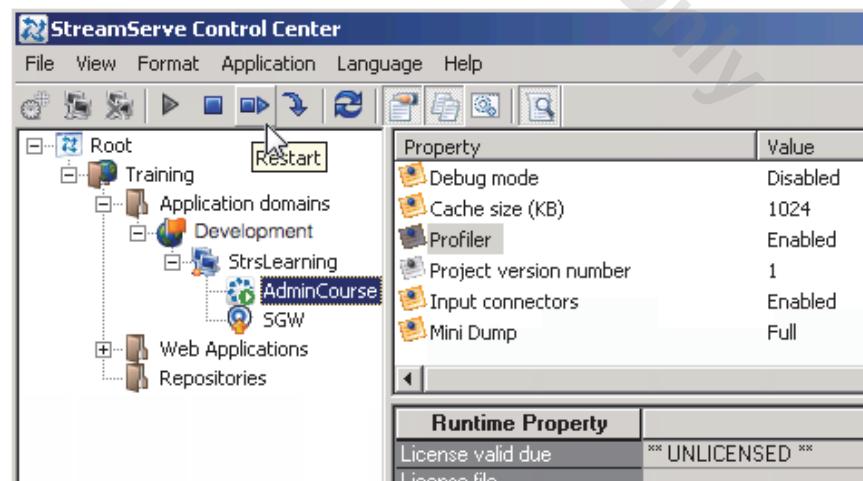
Figure 8-4:  
Enabling Profiler



The Edit Profiler Mode window opens.

3. In the Edit Profiler window select **Enabled** and click the **OK** button.
4. Making sure the **AdminCourse** application is selected, click the **Restart** button (or the **Start** button if the application is not started) in the Control Center toolbar.

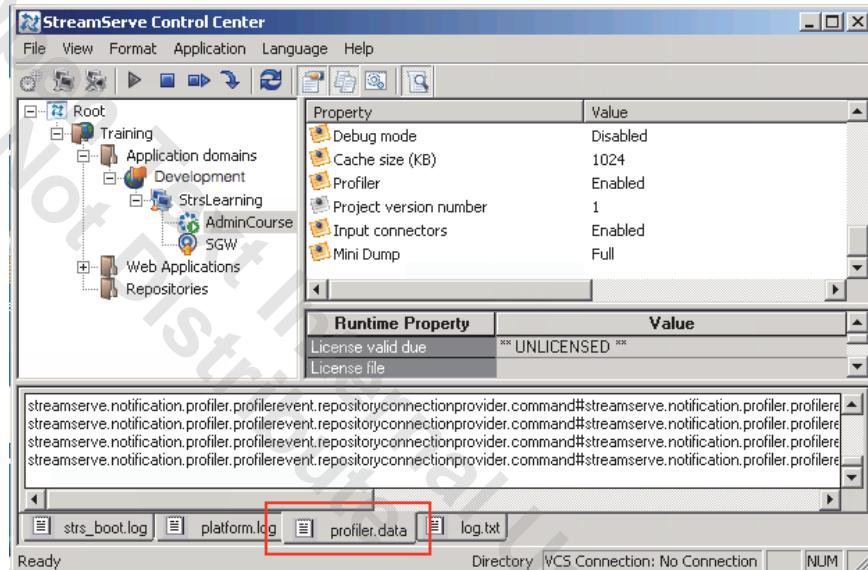
Figure 8-5:  
Restarting the  
AdminCourse Application



Upon startup a profiler.data file is added to the application working directory  
(C:\ManagementGateway\5.6.2\root\applications\AdminCourse\Development) and a new tab will be added in the logging area.

 You may need to reload Control Center for the profiler.data tab to display.

Figure 8-6:  
profiler.data



#### Set the Message Format for the Profile Provider

1. Open windows explorer and navigate to C:\Program Files\OpenText\StreamServe\Platform\Core\1.5.2\bin.
2. Make a copy of the **profilerservice.xml** naming it **profilerservice.xml.bak**.
3. Open **profilerservice.xml** with Notepad++.
4. Locate the line  
`<provider name="http://schema.streamserve.com/uid/resource/fileprofilerprovider/logfile/1.0">`
5. Add `messageformat="%#2##%#4##%#5##%#6##%#7##%#8"` at the end of the tag right before the closing ">":  
`<provider name="http://schema.streamserve.com/uid/resource/fileprofilerprovider/logfile/1.0" messageformat="%#2##%#4##%#5##%#6##%#7##%#8">`
6. Save the **profilerservice.xml** file and close Notepad++.

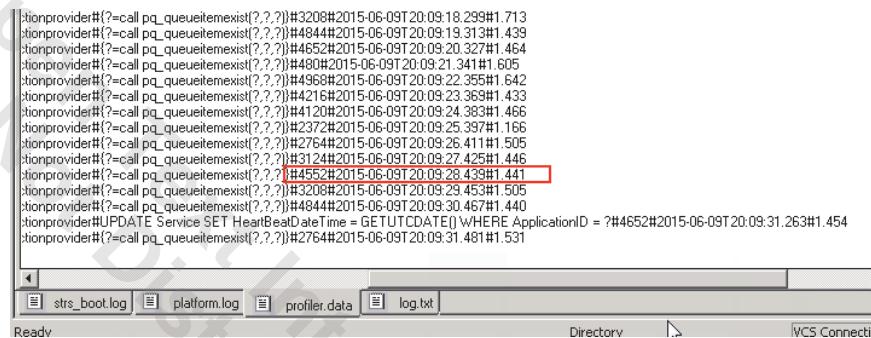
7. Stop the **AdminCourse** application by clicking the **Stop** button in the Control Center toolbar.
8. Once the **AdminCourse** application is stopped, right click within the **profiler.data** tab and select **Clear** in the pop-up menu to clear the content of the **profiler.data** tab.

The content of the **profiler.data** tab is cleared.

9. Making sure the **AdminCourse** application is selected, click the **Start** button in the Control Center toolbar.

10. In the **profiler.data** tab scroll to the right to review the content:

**Figure 8-7:**  
Enabling Profiler



A screenshot of a Windows Explorer window. The left pane shows a tree view with 'strs\_boot.log', 'platform.log', 'profiler.data' (which is highlighted in red), and 'log.txt'. The right pane displays the contents of the 'profiler.data' file. The file contains several lines of log entries, each starting with 'ctionprovider#(=?call pq\_queueitemexist(?,?))#3206#2015-06-09T20:09:18.299#1.713' and ending with '#1.439'. There are approximately 20 such lines. Below the file content, the status bar shows 'Ready', 'Directory', and 'VCS Connectic'.

```

ctionprovider#(=?call pq_queueitemexist(?,?))#3206#2015-06-09T20:09:18.299#1.713
ctionprovider#(=?call pq_queueitemexist(?,?))#4844#2015-06-09T20:09:19.313#1.439
ctionprovider#(=?call pq_queueitemexist(?,?))#4652#2015-06-09T20:09:20.327#1.464
ctionprovider#(=?call pq_queueitemexist(?,?))#480#2015-06-09T20:09:21.341#1.605
ctionprovider#(=?call pq_queueitemexist(?,?))#4968#2015-06-09T20:09:22.355#1.642
ctionprovider#(=?call pq_queueitemexist(?,?))#4216#2015-06-09T20:09:23.369#1.433
ctionprovider#(=?call pq_queueitemexist(?,?))#4120#2015-06-09T20:09:24.383#1.466
ctionprovider#(=?call pq_queueitemexist(?,?))#2372#2015-06-09T20:09:25.397#1.166
ctionprovider#(=?call pq_queueitemexist(?,?))#2764#2015-06-09T20:09:26.411#1.505
ctionprovider#(=?call pq_queueitemexist(?,?))#3124#2015-06-09T20:09:27.425#1.446
ctionprovider#(=?call pq_queueitemexist(?,?))#4452#2015-06-09T20:09:28.439#1.441
ctionprovider#(=?call pq_queueitemexist(?,?))#3208#2015-06-09T20:09:29.453#1.505
ctionprovider#(=?call pq_queueitemexist(?,?))#4844#2015-06-09T20:09:30.467#1.440
ctionprovider#(=?call pq_queueitemexist(?,?))#4652#2015-06-09T20:09:31.263#1.454
ctionprovider#(=?call pq_queueitemexist(?,?))#2764#2015-06-09T20:09:31.481#1.531

```

Notice that the messages are being displayed using the format indicated (messageformat = "%2##4##5##6##7##%8") The content of the **profiler.data** tab is cleared. See "Event Properties" on page 8-7.



#### Apply Filters to the Profile Output

1. Open windows explorer and navigate to **C:\Program Files\OpenText\StreamServer\Platform\Core\1.5.2\bin**.
2. Make a copy of the **profilerservice.xml** naming it **profilerservice.xml.bak2**.
3. Open **profilerservice.xml** with Notepad++.

4. Locate the following block (it is line 80 in the file):

```
<!-- <filters>
    <filter type="http://schemas.streamserve.com/uid/resource/
        profilerservice/namespacelfilter/1.0">
        <configuration>
            <nsmespacefilters xmlns="http://
                schemas.streamserve.com/
                    uid/resource/profilerservice/namespacelfilter/1.0">
                <nsmespacefilter
                    type="include">streamserve.notification.profiler.profilerevent.repos
                    itoryconnectionprovider</nsmespacefilter>
                    <nsmespacefilter type="exclude">*</nsmespacefilter>
                </nsmespacefilters>
            </configuration>
        </filter>
    </filters> -->
```

5. Uncomment the block by removing the "<!--" and "-->" in lines 80 and 89 respectively to activate this filter.

This filter configures the Profiler output to display only information about the repository connection provider.

6. Save the **profilerservice.xml** file and close Notepad++.
7. Stop the **AdminCourse** application by clicking the **Stop** button in the Control Center toolbar.
8. Once the **AdminCourse** application is stopped, right click within the **profiler.data** tab and select **Clear** in the pop-up menu to clear the content of the **profiler.data** tab.

The content of the **profiler.data** tab is cleared.

9. Making sure the **AdminCourse** application is selected, click the **Start** button in the Control Center toolbar.
10. In the **profiler.data** tab scroll to the right to review the content:

Notice that only the repositoryconnectionprovider events are displayed in the Profiler output.

## Exercise 2 – Analyze Profile Output

You can use a third-party tool to analyze the profile output. For example, you can import the output into a Microsoft Excel spreadsheet, or you can create a repository and analyze the output via the Database Management System.



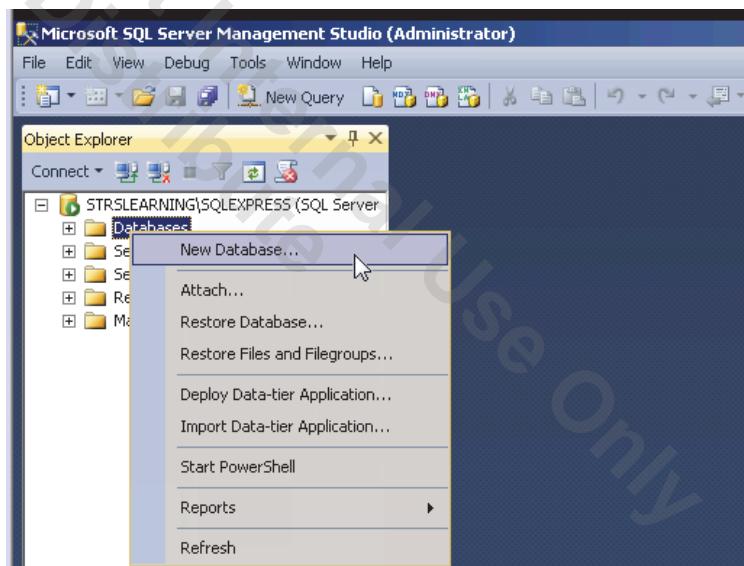
### Create a Profile Database in SQL Server

1. Navigate to **Start > All Programs > Microsoft SQL Server 2012 > SQL Server Management Studio**.

Microsoft SQL Server Management Studio opens presenting the Connect to Server window opens.

2. In the Connect to Server window click the **Connect** button.
3. Right-click the **Databases** node and select **New Database** in the pop-up menu.

Figure 8-8:  
DB Creation



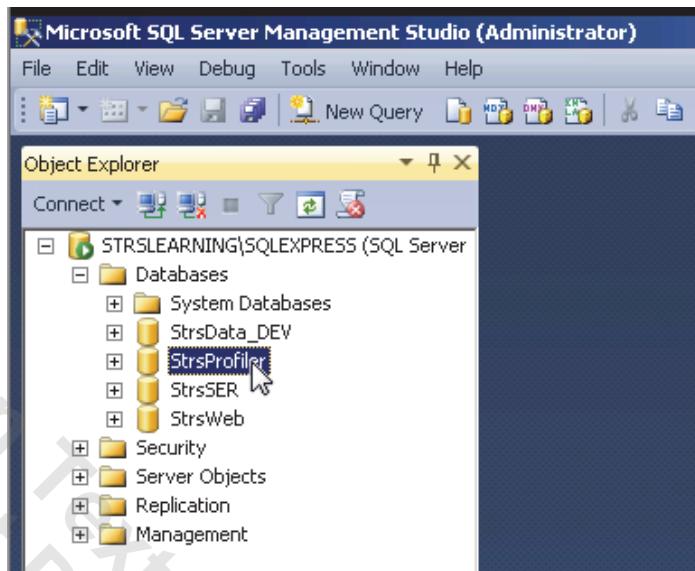
The New Database window opens.

4. In the New Database window enter **StrsProfiler** for the Database name field and click the **OK** button.

The StrsProfiler database is created.

Figure 8-9:

DB Created



5. In Microsoft SQL Server Management Studio make sure the newly created StrsProfiler database is selected and navigate to **File > Open > File**.
6. Browse to **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles\Scripts**, select **createTables.sql** and click the **Open** button.

The **createTables.sql** script opens in the right panel:

Figure 8-10:

DB Created

```

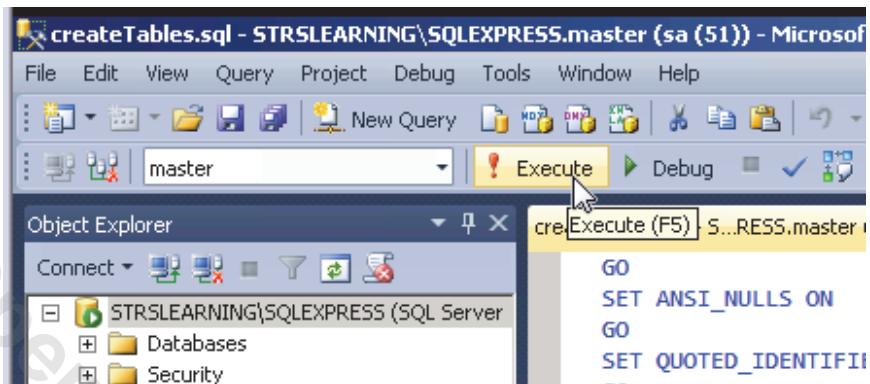
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ProfilerData](
    [EventID] [varchar](max) NULL,
    [EventNamespaceID] [varchar](max) NULL,
    [Context] [varchar](max) NULL,
    [ThreadID] [int] NULL,
    [Timestamp] [datetime] NULL,
    [ElapsedTime] [decimal](18, 3) NULL
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO

```

7. Review the script and notice the table structure.

8. In Microsoft SQL Server Management Studio toolbar click the **Execute** button to execute the script.

Figure 8-11:



Notice that no error is reported in the Messages panel (lower panel).



#### **Load the Profile Data into the Database**

1. In Control Center stop the AdminCourse application.
2. In Microsoft SQL Server Management Studio navigate to **File > Open > File**.
3. Browse to **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles\Scripts**, select **loadFromProfiler.sql** and click the **Open** button.

The loadFromProfiler.sql script opens in the right panel:

4. Review the script and notice that the script will open the profiler.data file.
5. In Microsoft SQL Server Management Studio toolbar click the **Execute** button to execute the script.

A message indicating the number of rows affected should be displayed in the Messages panel.



#### **Analyze the Profile Data for Repository Connections.**

1. In Microsoft SQL Server Management Studio navigate to **File > Open > File**.
2. Browse to **C:\Z\_CourseFiles\3-3730 56 00 StreamServe Installation and System Admin\ExerciseFiles\Scripts**, select **getData.sql** and click the **Open** button.

The getData.sql script opens in the right panel:

3. Review the script. Notice that the script will open the profiler.data file.

4. In Microsoft SQL Server Management Studio toolbar click the **Execute** button to execute the script.

The Results tab displays the grid containing the values returned by the query.

This data can now be used for analysis purposes.

Open Text Internal Use Only  
Do Not Distribute

Open Text Internal Use Only  
Do Not Distribute