OpenText Exstream Communications Server System Administration 16.6

OpenText Experience Suite

opentext™ | Professional Services
Learning Services

OpenText Communications Server Installation and System Administration v5.6.2 - First Edition: June 2015
OpenText CCE Installation and System Administration v16.0 - Second Edition: December 2016
OpenText Exstream Communications Server System Administration v16.3 - Third Edition: January 2018
OpenText Exstream Communications Server System Administration v16.6 - Fourth Edition: September 2019

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

Based on OpenText Extream 16.6

Developed by OpenText Learning Content Development and Learning Services

Course Name: OpenText Exstream Communications Server System Administration

Course Number: 3-3730

Part Number: 3-3730-166-00

Welcome

Welcome to this OpenText Exstream Communications Server System Administration course.

This course will provide an understanding of the OpenText Exstream platform and the associated system administration tasks.

The course begins with an introduction to the OpenText Exstream product, architecture and components. It continues with the identification of the required supporting stack and the corresponding configuration. Participants will then set up an OpenText Directory Services environment to integrate with OpenText Exstream.

After installation participants will cover a series of concepts that will allow them to successfully configure an OpenText Exstream Communications Server environment.

As a system administrator, you will learn techniques to monitor and tune the system.

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

Table of Contents

1. OpenText Exstream overview

Objectives 1-1
The Exstream platform 1-1
Components of the Exstream platform 1-3
- Exstream desktop applications 1-3
- Exstream web applications 1-4
- Communications Server layer 1-5
- Shared services layer 1-5
- External applications 1-6

How Design and Production works 1-7
What Exstream can do for your business 1-9
Tenancy concepts 1-9
Types of communications Exstream can produce 1-13

2. Hardware guidelines

Objectives 2-1
Hardware guidelines 2-1
- CPU 2-1
- Disk 2-2
- Memory 2-2

Hardware recommendations for development environments 2-2
Minimum hardware requirements for test environments 2-3
Hardware requirements for initial production 2-4
Hardware recommendations – expert sizing 2-4
- Initial sizing 2-4
- Configuration and Project development 2-4
- Testing and analysis 2-4

Enterprise configurations 2-5
Typical volumes 2-5

3. Exstream installation

Objectives 3-1
Installation components 3-1
- Runtime components 3-1
- Design Tools components 3-2
- Additional Exstream components 3-3

Exstream installation roadmap 3-3
- Steps before the installation 3-4
- Installation steps 3-5

Exstream bootstrap 3-6
Lab: Install the Exstream Runtime 3-7
Lab: Install the Design Tools 3-12

4. Setting up OTDS for Exstream

Objectives 4-1
Access control concepts 4-1
Each Exstream tenant has its own users 4-1
Roles control Access and permissions 4-2
Tenant OTDS 4-2
Multi-tenant users manage the overall environment 4-3
Multi-tenant OTDS 4-4
Options and recommendations 4-5
Understanding the OTDS tenancy concept 4-5
Requirements for Exstream tenants 4-5
Recommended method of setting up OTDS for Exstream 4-6
Alternative options – Exstream environments with multiple tenants 4-7
Alternative option – Exstream environments with single tenants 4-7
OTDS configuration requirements 4-8
Supported OTDS versions 4-8
Steps overview 4-8
Required OTDS backend configurations for Exstream 4-8
Required configurations 4-9
Design and Production and CAS Browser requirements 4-10
HTTPS and HTTP communication 4-10
Assigning Access roles 4-10
About assigning roles 4-10
Permissions are exclusive for each role 4-11
Role descriptions 4-11
Communications Designer and Content Author 4-12
Setting up OTDS for Exstream 4-13
Lab: Configure a multi-tenant OTDS 4-14
Lab: Create and configure a tenant 4-29

5. Setting up the Exstream environment

Objectives 5-1
About the ss_tenantadmin utility 5-1
Configuring the management gateway and creating the multi-tenant repository 5-1
Step 1 – Create the multi-tenant repository connection profile 5-2
Step 2 – Create the connection profile to the multi-tenant OTDS 5-3
Step 3 – Create the multi-tenant repository 5-4
Step 4 – Configure the management gateway to use the connection profiles 5-4
Step 5 – Start the management gateway service or process 5-5
Adding a tenant 5-5
Step 1 – Create the connection profile to the tenant repository 5-6
Step 2 – Create the connection profile to the tenant OTDS 5-7
Step 3 (optional) – Create a tenant repository 5-7
Step 4 – Create a tenant that uses the connection profiles 5-8
Lab: Configuring the management gateway and creating the multi-tenant repository 5-9
Lab: Adding a tenant 5-12

6. Application domains

Objectives 6-1
About domains 6-1

Creating application domains ... 6-2
Repositories and domains ... 6-2
Shared or separate repositories ... 6-2
Separate repositories ... 6-3
Shared repositories ... 6-3
Combinations of shared and separate repositories ... 6-4
Service gateways, the web applications, and domains ... 6-4
Lab: Creating a domain ... 6-6

7. Repositories

Repositories overview ... 7-1
Repository types ... 7-1
Repositories and domains ... 7-2
Domain-required repositories ... 7-2
Adjusting repositories ... 7-3
Indexing ... 7-3
Partitioning ... 7-3
Estimating sizes of database and transaction log ... 7-4
Database file ... 7-4
Transaction log ... 7-4
About job tracking ... 7-4
Input and output jobs ... 7-4
Top jobs ... 7-5
Creating the repositories ... 7-6
Lab: Creating the repositories ... 7-8

8. Communications Builder projects

Objectives ... 8-1
Definition of a project ... 8-1
Communications Builder ... 8-2
Communications Builder project components ... 8-3
Platform ... 8-4
Platform view ... 8-4
Platform layers ... 8-5
Generic settings ... 8-6
Physical layer settings ... 8-6
Message ... 8-7
Message View ... 8-8
Events and processes ... 8-9
Processing Engine ... 8-10
Runtime configuration ... 8-11
Runtime view ... 8-12
Resource Set ... 8-13
Managing projects ... 8-14
Lab: Project Management ... 8-15

9. Communications Server applications

About Communications Server applications ... 9-1
Adding Communications Server applications ... 9-2

New Application dialog box 9-3
Managing Communications Server applications 9-5
Starting, Stopping, and Restarting Applications 9-5
Deleting applications 9-6
Updating application properties 9-7
Adding other Communications Server applications 9-8
Administering applications on remote hosts 9-10
Directory structure for Communications Server applications 9-10
Base directory for projects 9-10
Working directory 9-10
Working directories for physical platform layers 9-11
Deploying a Communications Builder project 9-12
Deploy wizard – Select the applications 9-13
Deploy wizard – Select file 9-14
Deploy wizard – Select Physical layer 9-15
Testing Communications Server applications 9-15
Lab: Creating an Communications Server application 9-16
Lab: Creating a Service Gateway 9-23

10. Exstream web applications

Overview 10-1
Exstream web applications 10-1
Web applications communication 10-3
Web application requirements 10-4
Accessing WorkShop 10-5
Accessing Supervisor 10-6
Accessing StoryBoard 10-7
Accessing ReTouch 10-8
Accessing Writer 10-9
Accessing Rule Editor 10-10
Configuring connection properties 10-10
Service gateway properties 10-11
Service gateway secure mode 10-12
Securing management gateways 10-13
About the demonstration Exstream keystore 10-13
Setting up a secure channel for the management gateway 10-14
Securing Control Center, Communications Builder, and Describer 10-14
Management gateway properties 10-15
Lab: Deploying the Exstream web applications 10-16
Lab: Configuring the Management and Service Gateways 10-18

11. Profiling

About profiling 11-1
Exstream Profiler 11-1
Enabling Profiler 11-2
Configuring Profiler 11-2
Configuring a Profile provider 11-2
Specifying which event properties are logged 11-3
Applying event filters 11-5
Flushing Profiler data to the output file 11-6

Scheduling the Profiler counter events 11-6
Analyzing profiler output 11-6
Lab: Profiler Services 11-7
Lab: Analyze Profile Output 11-10

12. Monitoring

About Control Center logs 12-1
Log files and paths 12-1
Log messages 12-5
Log message parts 12-5
Log messages included 12-6
Log message severity levels 12-6
Log examples 12-7
Setting log levels 12-9
Platform log levels 12-9
Setting Exstream Communications Server application log level 12-10
Debugging applications 12-11
Logging to the database 12-12
Exstream notifications overview 12-12
Exstream Communications Server notification types 12-13
Preparing the Communications Server application to use notifications 12-14
Task Scheduler overview 12-16
Settings to run system command tasks 12-17
Configuring a Task Scheduler application 12-18
Configuring the Schedule 12-19
Deleting expired content 12-20
Recommended Communications Builder platform settings 12-20
Task scheduler deletion and expiry events 12-20
Recommendations for scheduling the deletion tasks 12-21
Scheduling a Task Scheduler to delete or expire content 12-21
Lab: Changing the log level at runtime 12-22
Lab: Configuring Java notifications 12-23
Lab: Creating a Task Scheduler application 12-26

13. Installing and applying Hotfixes

About installing and applying hotfixes 13-1
Installing a hotfix 13-1
Applying a hotfix to the repositories 13-2
Applying a hotfix to the tenant repository 13-3
Applying a hotfix to the repositories with Control Center 13-3
Applying a hotfix to the repositories with the hotfix-repo script 13-3
Applying a hotfix to the repositories manually 13-3
Applying a hotfix to the web applications 13-4

14. High Availability Guidelines

Objectives 14-1
Introduction 14-1
Cluster concepts 14-2
Cluster terminology 14-3

Planning the cluster 14-3
Basic environment 14-3
Complex environment 14-3
Exstream components 14-4
Load balancing and failover 14-5
Load balancing via shared queues, transactional support, and scheduled spooling 14-6
Failover via heartbeats 14-6
Failover with Communications Server applications 14-6
Failover and load balancing for service gateway applications 14-7
Failover during web authentication request 14-7
Database for Exstream repositories 14-8
Setting up a database in a cluster 14-10
Printers and print servers 14-10
Exstream web applications 14-11
Making the database instance highly available 14-11
Making the Java application server highly available 14-12

Text Conventions

This workbook uses the following conventions:

Convention	What it is Used For
Italic	Italics are used for Workshops and Exercises.
`Monospace`	Monospaced text is used to represent sample code.
Bold	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.

opentext™ | Professional Services
Learning Services

OpenText Exstream Communications Server
System Administration 16.6

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. OpenText Exstream overview

Objectives

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

- Identify the functionality scope of OpenText Exstream
- Describe the Exstream platform
- Identify the components that comprise the OpenText Exstream platform
- Define some basic tenancy-related concepts

The Exstream platform

The Exstream platform provides an integrated software solution for creating, managing, and delivering customer communications of any type, regardless of complexity, variability, or delivery channel, and enables you to eliminate many systems and point solutions by using the comprehensive omni-channel delivery capabilities of the platform. The fully integrated, robust, and flexible capabilities of Exstream let you streamline business processes with end-to-end processing of documents – from content ingestion through composition to output. Sophisticated capabilities such as campaign management, dynamic whitespace management, data-driven charting, multi-language support, and more help you acquire, retain, and grow customer relationships.

Classic Exstream software (Design and Production) is a part of the integrated Exstream platform and provides you with robust capabilities for designing, managing, and delivering customer communications using Design Manager, Designer, and Logic Designer with the Exstream production engine.

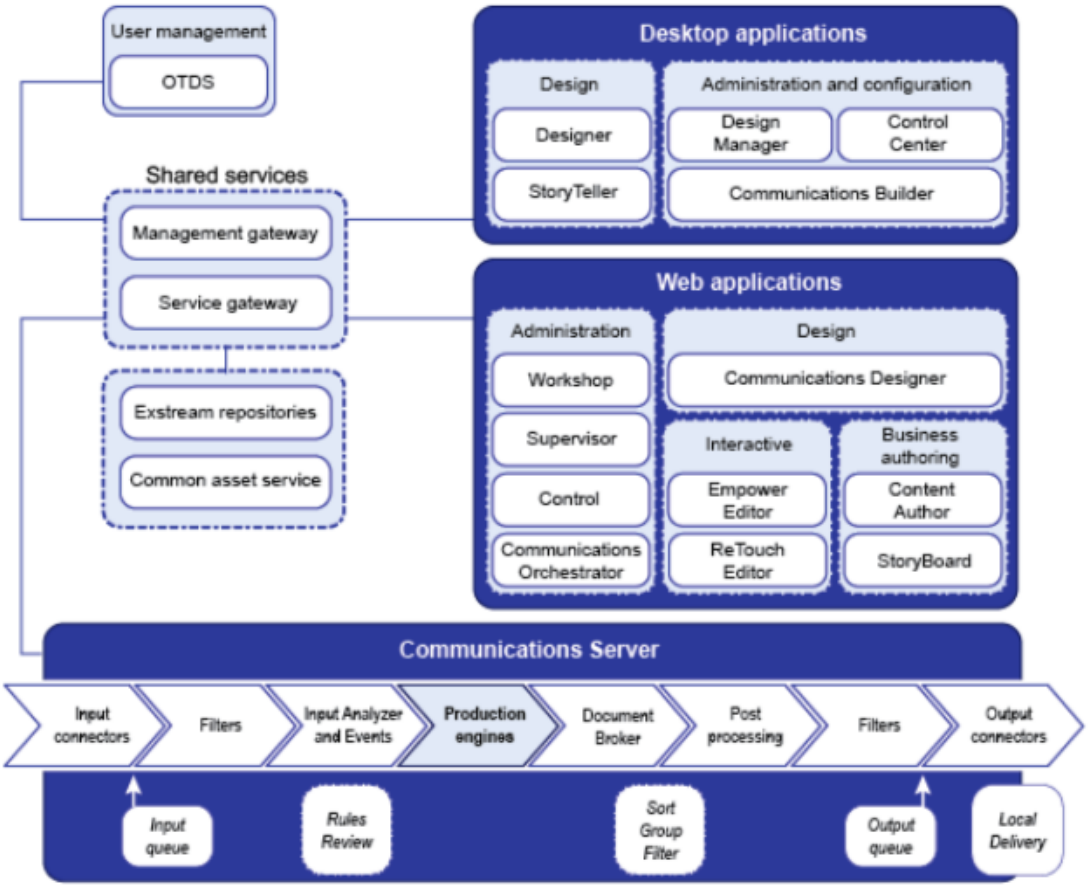


Figure 1-1: Exstream platform

Components of the Exstream platform

The Exstream platform provides you with a comprehensive solution to meet various business requirements. The way your organization installs and implements the following components of Exstream depends on your specific business requirements:

- Exstream desktop applications
- Exstream web applications
- Communications Server layer
- Shared services layer
- External applications

Exstream desktop applications

The desktop applications in the Exstream platform are used for designing customer communications and for modeling, implementing, and administering communication workflows.

Design Manager

Design Manager lets document designers create and manage the design objects that make up a Design and Production application, including design templates, data files, variables, printers, and production equipment. Additionally, system administrators in Design and Production can perform administration tasks such as managing users and design groups.

After you have set up your application in Design Manager, you can design your customer communications in Designer. You can then use Design Manager to compile your Design and Production applications and configure test and production runs for delivering communications to customers.

Designer

Designer provides the graphic design interface for designing and modifying Design and Production applications. With this tool, document designers can create and format content for pages and messages, design graphic elements, insert variables to customize documents, and put together the overall design and layout for customer communications. The device preview feature provides a convenient way to see how the customer communications look when viewed as different output formats on various devices.

StoryTeller

StoryTeller provides the graphic design interface to create the basic structure and layout of templates that can be used as building blocks for creating customer communications using the StoryTeller engine. You can access StoryTeller from within Communications Builder.

Communications Builder

Communications Builder is the main tool for modeling communication workflows in the Exstream platform. In Communications Builder, system administrators and document designers can use the connectors, filters, and queues that are available in the tool to define the collection and delivery of data and specify how Communications Server will produce customer communications. These communication workflows are stored as Communications Builder projects.

Control Center Control Center is used to deploy and administer Exstream jobs. To run a job and produce customer communications, Communications Builder projects are deployed to Communications Server applications, and these applications are then run and administered from the Control Center interface.

Exstream web applications

The Exstream web applications are intended for many different scenarios, such as campaign and resource management, interactive editing, business content authoring, and job monitoring. Each web application is available as a standalone application and can be hosted in an existing business application or as part of a workflow.

Workshop Workshop provides a graphical interface for managing and interacting with resources that are stored in the common asset service (CAS). The CAS is a central shared repository that provides access to and storage for the resources used in Exstream solutions. This includes image assets, Design and Production application package files, StoryBoard templates, Communications Builder projects, and PowerDocs templates.

Communications Designer Communications Designer lets users design communications in an intuitive web-based design environment. Users can leverage data files and output queues created in Design and Production to create customer output from communications that are designed in Communications Designer.

Content Author Content Author lets business users add content to Design and Production designs without requiring them to re-package their applications. You can use Content Author in conjunction with Workshop to create and modify themes generated from Design and Production templates, and then publish the updated content to include it directly in the next engine run.

Empower Editor The Empower Editor is a browser-based interactive editing experience for personalizing communications based on customer interactions. In Empower Editor, business users can update documents that have been created in Design and Production. These interactions include making selections from pre-defined options, changing text and images, updating variable data, adding additional documents and recipients, previewing the communications, and initiating the fulfillment process.

StoryBoard In StoryBoard, business users can enhance StoryTeller templates and personalize communications by adding text, images, and rules. Users can also use the device preview capabilities of StoryBoard to see how communications look in print and email format, and on different devices.

ReTouch Editor ReTouch is a lightweight web application that lets business users interactively edit documents generated from StoryBoard templates and also halted in reviewer.

Supervisor Supervisor is a web application that lets system administrators track and manage jobs and documents as they move through the Exstream repositories and queues during their lifecycle. The application also provides a basic statistics view where administrators can monitor job processing statistics.

Control Control provides a browser-based way for operations and system administration users to perform many common job deployment and monitoring functions. An extension to the desktop Control Center product, this thin client interface provides an easy-to-understand dashboard with status information for all applications within a domain. It also allows users to start and stop Control Center applications, as well as redeploy Communications Builder projects to existing Control Center applications configurations. Control is supported on touch mobile devices.

Communication Orchestrator Communications Orchestrator is a web application that lets users create flow models for customer communications management processes.

Communications Server layer

The Communications Server is the central layer of the Exstream platform that connects to your enterprise systems, creates customer communications, and delivers communications in print or electronic format. Communications Server generates output based on communication workflows that are designed using Communications Builder.

You can use Communications Server along with Communications Builder and Control Center to orchestrate production jobs by connecting your data sources, designs, and connectors to produce customer communications, and by defining how and when to deliver the output. The following processing engines are available in Communications Server:

Exstream engine The Exstream engine is the production engine used in Design and Production. It is a high- throughput, multichannel engine used to generate communications by processing Design and Production applications. When you use Communications Server to run Exstream engine jobs, you can take advantage of the connectors and orchestration features that are available in the Exstream platform.

StoryTeller engine Communications Server automatically invokes the StoryTeller engine to generate customer communications based on configurations made in StoryTeller, StoryBoard, and ReTouch.

Shared services layer

Communications Server applications and other Communications Server applications, such as service gateways, run on the Exstream framework, which contains a management gateway and core platform services.

The management gateway connects the Exstream desktop applications to the user management component and the Exstream repositories, as well as to other Exstream desktop applications, such as Communications Builder and Control Center.

The service gateway connects the web applications to the Exstream repositories and to the CAS.

External applications

The following external software applications are used for critical user management and monitoring functions and are installed separately from the Exstream platform:

OpenText Directory Services

OpenText Directory Services (OTDS), is an OpenText identity management system that provides access control for the Exstream platform. OTDS can synchronize with external identity providers like Microsoft Active Directory to retrieve user and group information, and map that information to OTDS access roles, providing secure access to each platform component. OTDS can be downloaded from My Support and is installed separately from the Exstream platform.

OpenText Experience Analytics

Experience Analytics is used for end-to-end tracking of communications through the document tracking framework. Experience Analytics can be downloaded from My Support and is installed separately from the Exstream platform.

The following image depicts the Exstream tools:

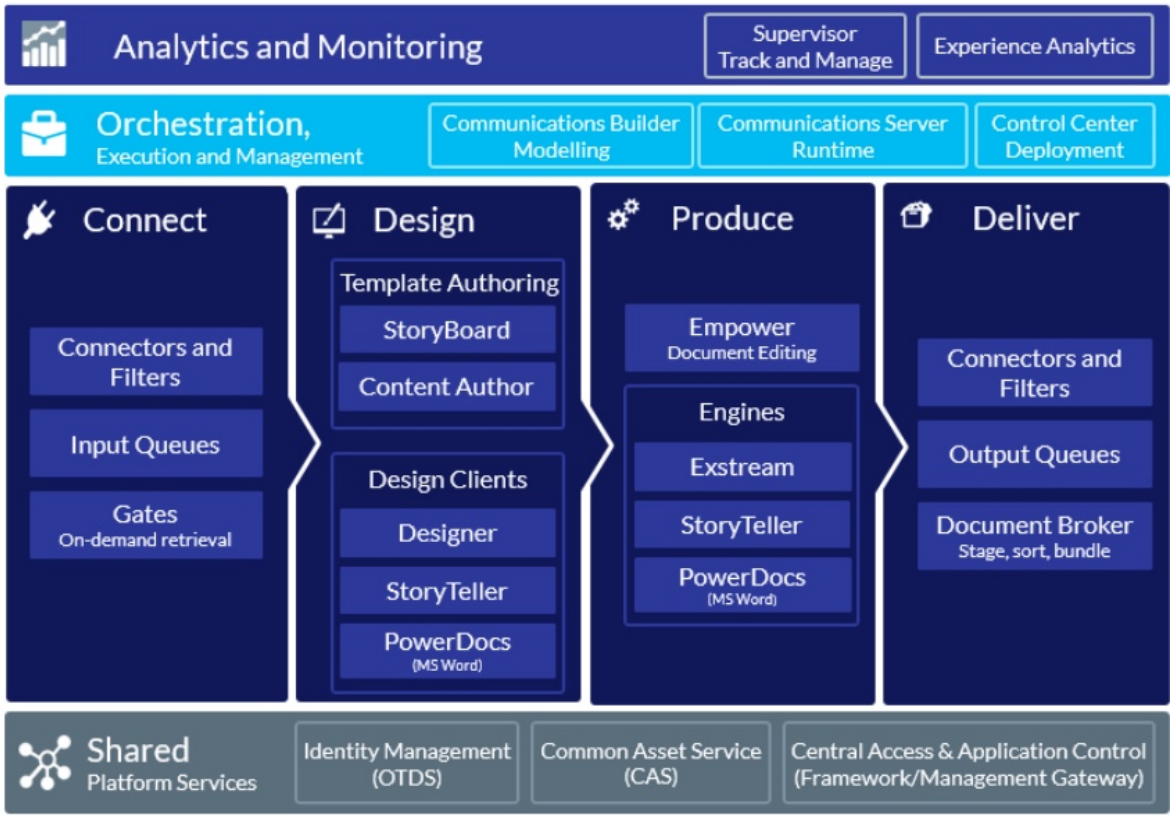


Figure 1-2: OT Exstream platform

How Design and Production works

Design and Production is a modular solution that offers robust capabilities for reducing complexity, streamlining business processes, and creating higher quality, more effective communications for delivery in high-volume, on-demand, and interactive environments.

When using Design and Production within the integrated Exstream platform, users have access to new design and content authoring, resource management, and engine orchestration capabilities that are available with the platform. The following graphic provides a high-level view of how Design and Production fits into the Exstream platform and interacts with other platform components:

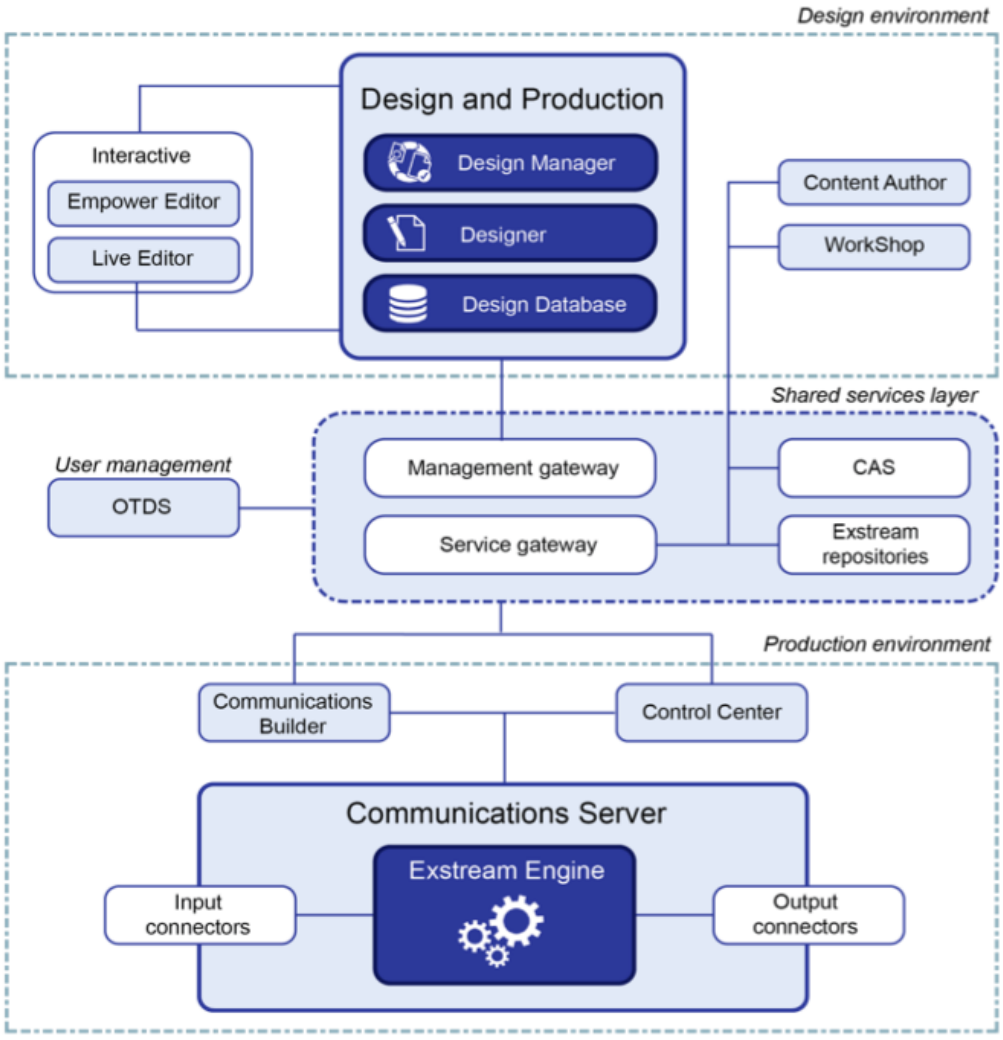


Figure 1-3: Exstream platform components

Design Manager and Designer form the desktop design environment that is used to design and set up customer communications in the form of Design and Production applications. Interactive documents (Live documents or Empower documents) are created in the same design environment as standard Design and Production applications, and leverage all of the available content integration and delivery capabilities.

Users interact with the Exstream platform through this design environment, which is connected to the design database and shared services. After Design and Production applications are designed, your files are packaged for production, and processed by the Exstream engine. The shared services layer connects the desktop design environment to the Exstream engine in the server production environment. In particular, the management gateway connects Design and Production to the OTDS user management system and the common asset service (CAS), and enables access to the Communications Server layer that is used for engine orchestration.

Various capabilities are available in this production environment for high-volume, on-demand, and interactive delivery. Sorting and bundling capabilities let you group document applications to reduce print and mail costs. Testing tools let you easily review documents to ensure changes are made and that document outputs are correctly produced. Design and Production can pull content from nearly any data source, including legacy systems. The software can receive data through web systems for creation of on-demand or interactive documents. Design and Production can then process all of the received data to re-enter and update your systems.

If you have licensed the Communications Server component in the platform in addition to Design and Production, you can use Communications Builder to model communication flows and use the connectors, filters, and queues that are available to you to configure your output delivery in Communications Builder projects. Based on these projects, you can then orchestrate Exstream production engine jobs using the Control Center to deploy the projects to Communications Server.

What Exstream can do for your business

Your business has made investments to manage and govern the vast amounts of content continually flowing in and out of your organization. And, as new technology continues to emerge, so will the content you need to manage. The key to fully differentiating your organization is the integration and deployment of systems and efficient processes to not only capture, manage, and govern content, but to get maximum value from it by creating a better and more effective experience for your customers through variable data publishing and multiple channel delivery.

Exstream is a single platform for producing documents of any type, regardless of complexity, variability, or delivery channel. You can eliminate many systems and point solutions simply by using Exstream. The fully integrated, robust, and flexible capabilities of Exstream let you streamline business processes with end-to-end processing of documents – from content ingestion through composition to output. Sophisticated capabilities such as campaign management, dynamic whitespace management, data-driven charting, multi-language support, and more, help you acquire, retain, and grow customer relationships.

Tenancy concepts

Exstream supports both single-tenant and multi-tenant environments. In single- tenant environments, a company or organization runs the Exstream components on one or more computers or hosts. The company develops and manages its own Communications Server and repositories.

In multi-tenant environments, several client organizations (or tenants) share one or more instances of the framework and management gateway. Each tenant in the Exstream environment has a unique ID and runs its own Communications Server applications. All the applications and data in the environment are separated by tenant IDs. This allows tenants to share databases or schemas or to use separate databases or schemas.

Each tenant has its own users and groups in OTDS, which controls who can access the tenant’s applications and data. When users of these groups log on to WorkShop, StoryBoard, Control Center, Describer, etc. they can only view and access the data and resources that belong to their tenant. For example, if an Exstream environment has two tenants, tenant 1 and tenant 2, then when a user from Tenant 1 logs on to Control Center, only the applications and repositories that belong to Tenant 1 are displayed.

Exstream provides the following major value propositions to help you improve and grow your business:

Increase communication effectiveness

The integrated marketing and dynamic whitespace capabilities of Exstream allow you to ensure every customer communication includes timely, relevant offers and informative messages that improve the customer experience and cross-sell additional products and services. Using Exstream, you can prioritize and incorporate only the most relevant messages and promotions into documents based on business rules, available white space, or the point of need (for example, to explain a complex billing line item). Exstream can also drive inserters to select or remove pre-printed inserts based on customer criteria or other factors you define.

In addition to relevancy, the key to improving customer satisfaction is clarity and the ability to deliver communications through preferred channels. Exstream offers a comprehensive feature set for producing visually appealing documents that simplify complex information, including data-driven charts of almost any type and support for all color modes.

From one application design, you can deliver documents to customers in their native language. Exstream even includes a spell checker for 28 languages. You can also easily deliver documents to customers through their preferred channels (for example, email only).

With Exstream, you can create high-quality, easy-to-understand, and timely communications for customers, increasing their satisfaction and reducing expensive calls to your call center.

Figure 1-4:

Effectiveness

Reduce complexity

Most organizations have significant costs tied up in multiple document creation technologies and processes that connect silos of information to customer document applications. Exstream software provides a single solution for design through delivery of any type of document across the enterprise, regardless of complexity, variability, or output channel—from fully customized high-volume statements, bills, and complex publications, to on-demand marketing and self- service web applications, to personalized correspondence and proposals produced interactively by customer-facing employees. The ability to do all of this using a single software platform allows you to integrate (or even eliminate) silos, significantly reducing costs and ensuring consistency across all customer communications.

Exstream was designed from the ground up to easily fit into any IT environment, including Service-Oriented Architectures (SOAs). Through web services and its comprehensive connector technology, Exstream directly accesses content from enterprise systems and data sources to drive the creation of personalized customer communications, eliminating the need for systems that consolidate disparate data. Exstream can directly access and process in one pass multiple data sources of almost any type, structured or unstructured.

With Exstream, you design objects and combine them together to build applications in an object- oriented fashion. All design elements are stored in a common database so they can easily be re- used across applications, significantly reducing document development and maintenance time.

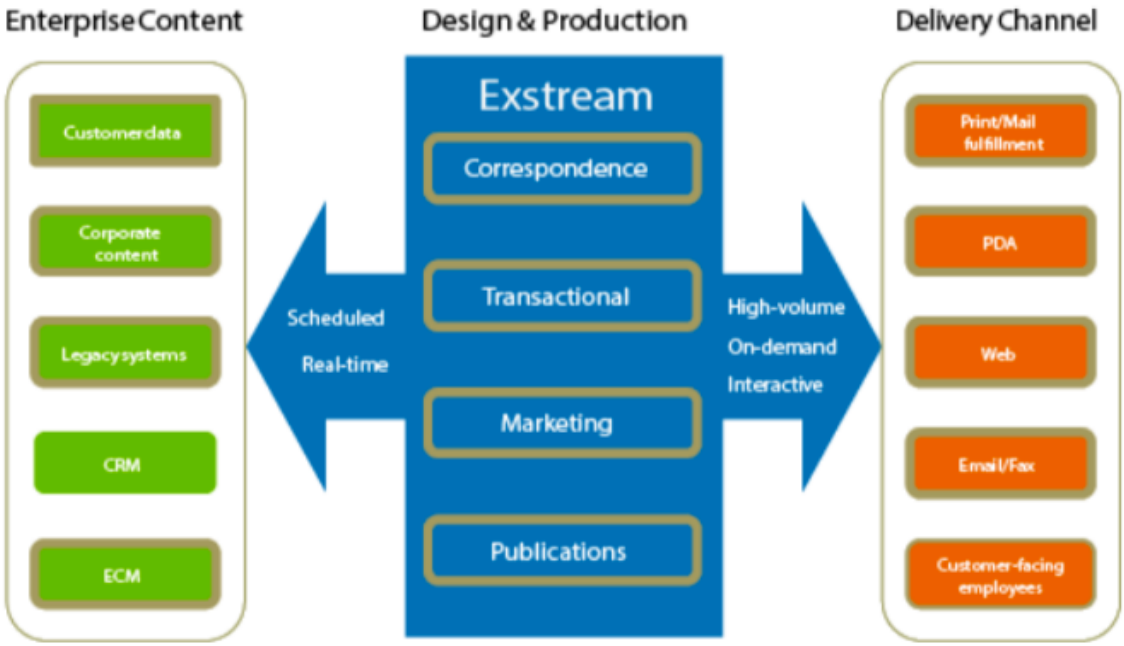


Figure 1-5: Reduce complexity

Streamline document processes

The Exstream software platform provides fully integrated capabilities for end-to-end document processing, including variable design, testing, real-time composition, advanced data and content integration, output to a variety of formats, high-volume optimization and workflow, and controlled editing of interactive documents. The robust graphical design environment of Exstream gives developers all the capabilities they need to design any kind of document, regardless of complexity, type, or variability. Browser-based design collaboration with built-in approval workflow allows marketing and line of business users to remotely create variable messages that are integrated into documents at run time, ensuring relevant and current information is communicated in a timely manner.

Exstream includes many capabilities for testing document applications, significantly improving productivity and optimizing performance. The design interface doubles as an online viewer so you can immediately prototype, review, and modify documents online, and other tools are provided for comparing output, simplifying test data, analyzing marketing campaigns, and performing regression testing.

The same high-performance Exstream engine is used to compose millions of variable documents at a time for high-volume print or mail delivery, or to compose a single document on demand when called by a Web service or custom interactive system. Exstream produces fully personalized document output up to 10 times faster than alternative solutions in customer benchmark tests.

From one template design, Exstream can natively produce more than 20 different print and electronic output formats. Rules can drive different output for different customers in a single run.

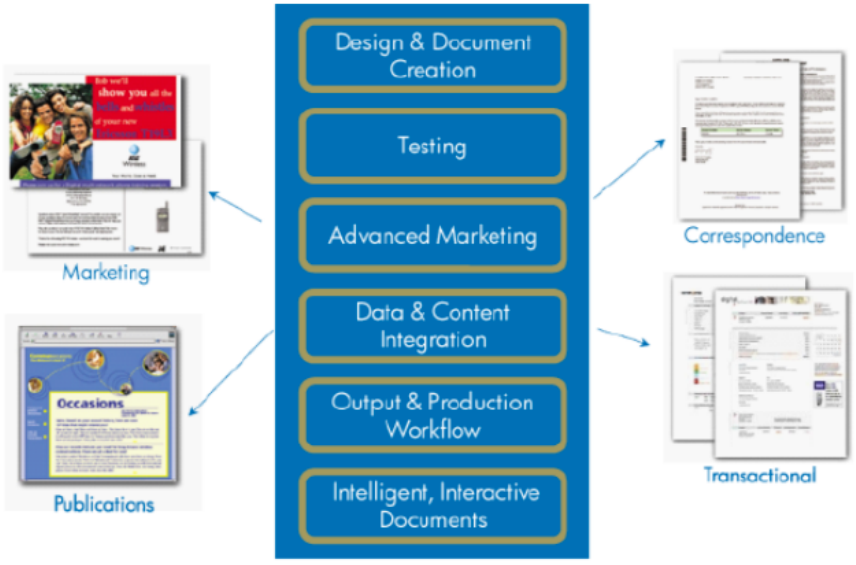


Figure 1-6:

Streamline processes

The comprehensive capabilities Exstream provides for high-volume print/ mail environments eliminates the need for post-processing programs and maximizes production efficiency by allowing you to control postage weights, prepare data for postal sort, drive inserters, and household documents to save unnecessary paper and postage costs.

Any Exstream document can be designed and deployed as an interactive document (a Live document or an Empower document) to be completed at the point of need by customer-facing employees. Edited interactive documents can be sent back to the Exstream engine to drive other processes, such as automating fulfillment, updating corporate systems, records management, and archive systems, or making copies of the edited document in other formats.

Types of communications Exstream can produce

The types of personalized customer communications you can create are virtually limitless with Exstream. It has been used as a proven solution for several types of document applications in many industries.

Correspondence Delivering accurate, consistent, and effective correspondence is one of the most important relationship-building opportunities companies have with customers. Yet, executed poorly, correspondence can damage or even end customer relationships, and can also be one of the most expensive processes companies incur. Unclear, confusing, or incorrect communications result in poor perception from the customer and require even more correspondence to correct the situation.

From personalized letters sent out to millions of customers at a time, to welcome kits generated on demand over the web, and customer service responses or sales proposals created ad hoc, Exstream provides a common platform that allows you to regain control over customer correspondence, reducing costs and ensuring communications are of the highest quality.

Correspondence applications built with Exstream can do the following:

- Lower IT costs
- Streamline processes
- Reduce errors and miscommunications
- Ensure compliance
- Support multiple languages
- Optimize high-volume output
- Enforce branding standards

Marketing When it comes to marketing applications, timeliness and relevance are of primary importance, so marketing and business users must be empowered to work independently of IT to achieve aggressive time-to-customer goals. Exstream is designed better than any other solution to meet the varying requirements of different users with different skills—from marketing and line of business managers to operations and IT—helping you create more attractive, easier to understand, relevant marketing communications that encourage buying and other desired behaviors based on individual customer data.

The role-based Windows design interface of Exstream can be configured so various users across the extended enterprise can easily and independently manage different aspects of the workflow process, such as design, versioning, message creation, business rules, and output controls.

Exstream marketing applications include personalized newsletters, email, direct mail, TransPromo communications, pitch books, and more. Campaigns can be linked and tracked for targeted, round-trip campaign management. Using Exstream, some companies have seen 30 percent improvements in customer response rates and have gone to market up to 85 percent faster.

Marketing applications built with Exstream can do the following:

- Achieve simple and seamless integration
- Cut paper and postal costs
- Boost response with color and graphics
- Automate time-consuming design steps
- Ensure brand compliance
- Utilize Web-based variable messaging
- Reward your best customers

Publications Industry-leading companies worldwide are building customer loyalty by efficiently producing timely, customized insurance policies, new member booklets, travel guides, catalogs, prospectuses, financial plans, and more – and they are doing this all with Exstream software. Publications such as these are often complex and contain multiple pages. Exstream lets you easily create documents that include tables of contents, footnotes, cross-references, and indexes.

No other enterprise document automation software provides a single platform to create complex documents as well as customer correspondence, statements, and marketing applications. Exstream lets you replace many existing software products and in-house systems with a single, comprehensive platform for creating all document types – regardless of delivery channel or environment (for example, high-volume, interactive, on-demand) – resulting in significant reductions in maintenance and development costs.

Publication applications built with Exstream can do the following:

- Simplify sophisticated publications
- Improve productivity and reduce costs
- Reduce document inventory
- Ensure compliance
- Boost response with color and graphics
- Make documents more accessible
- Eliminate repetition in section-based documents

Transactional Exstream offers an unparalleled ability to quickly and easily produce sophisticated, complex transactional documents, and helps you comply with the U.S. Securities and Exchange Commission (SEC) filing process, including creation of EDGAR HTML for SEC-required revision tracking. Our software automates document workflow processes and handles the most elaborate composition requirements, regardless of variability and customer delivery channel requirement, as much as 10 times faster than alternative solutions according to customer benchmark tests. And, the software's unique functionality to create automated, transactional- driven tables of any type is unmatched in the industry. From automated rows and columns to rounded corners, no other solution provides as much flexibility and as many options to easily present information in tabular format.

Additionally, with Exstream you can easily turn ordinary transactional documents into targeted communications that improve the customer experience. Through integration with your CRM system, you can deliver communications based on customer segmentation. For example, you might choose to send color statements (on higher quality paper if printing and mailing) to one set of customers, but send monochrome statements to another set of customers. Furthermore, the software's sophisticated whitespace management and rules-based messaging capabilities let you maximize the use of available white space and dynamically insert only relevant information and promotions into customer statements, bills, and other transactional documents based on individual demographics, preferences, and buying patterns.

Transactional applications built with Exstream can do the following:

- Integrate with leading BPM, CRM, and ECM solutions, as well as other front and backend office systems
- Utilize data from any source on any platform
- Automatically prioritize targeted messages
- Create messages in the customer's preferred language
- Store whether a customer responded or not
- Allow custom graphics based on customer-specific information
- Maximize the use of available white space
- Control message placement in regulated documents

2. Hardware guidelines

Objectives

On completion of this chapter participants should be able to:

- Identify the recommended hardware guidelines for an OpenText Exstream implementation
- Identify the recommended hardware for a development, test and productions environments
- Review enterprise configurations

Hardware guidelines

This section provides recommendations for the hardware requirements when planning and implementing OpenText Exstream solutions that involve Communications Server applications.

- High-level considerations that affect hardware sizing for Communications Server applications.
- General hardware recommendations.

It is assumed that issues regarding performance, architecture, and Communications Builder Project design have been considered prior to the implementation.

For environments with greater complexity and higher throughput, you can engage OpenText Exstream Professional Services to provide a customized hardware recommendation. Contact your local OpenText office for more information.

CPU

Tests show that two different Exstream 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.

OpenText recommends you use:

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

Disk

The core Communications Server is extremely write-intensive. File systems associated with the the Exstream working directories and temp directories should be write-optimized, and disk caching should be enabled. Write and read operations also occur against the runtime repositories, so these must also be write-optimized.

If one machine is used for both the Exstream repositories and the Communications Server applications, OpenText recommends using RAID 10 disks because they are optimized for write-intensive processes. RAID 5 is optimized for read operations and is not recommended for running Communications Server applications.

If the Communications Server applications and the Exstream repositories are running on separate machines, ordinary disk striping without parity is sufficient for the machine running the Communications Server 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

Hardware recommendations for development environments

The following recommendations are intended for development environments where the following Exstream components run a single machine:

- Communications Builder
- Communications Server application(s) used for development
- Exstream repositories

Communications Builder

Communications Builder can be quite resource-intensive and is not optimized for thin clients.

Managing Communications Server applications

Communications Server applications can be managed with Control Center by the management gateway application, through third-party tools, or from a command line. Communications Server 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 Communications Server 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	Microsoft Windows
Processor	Dual-core
Memory	8 GB, or more
Hard Disk	40 GB – can be more depending the data used.
Network	Fast Ethernet, TCP/IP
Display	1280x1024, or higher

Minimum hardware requirements for test environments

These are the minimum hardware requirements for running Communications Server applications in a test environment. It is assumed that only Communications Server applications run on this hardware, and that Communications Builder and the Exstream repositories do not. These recommendations are intended for small environments, since they are only capable of processing relatively small quantities of data.

Platform	Memory
Red Hat Enterprise Linux	1 GB per CPU, or more
SUSE Linux Enterprise Server	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 Communications Server application in small batches, with a Project of low-to-medium complexity. It is assumed that only Communications Server applications run on this hardware, and that Communications Builder and the Exstream repositories do not. Architecturally, a single centralized instance is assumed, with no wide area network bandwidth constraints.

Large size

- 2 servers with 16 CPUs.
- 64 GB RAM available for Communications Server applications per server.

Medium size

- 4 CPUs with 32 GB RAM available for Communications Server applications.

Small size

- 2 CPUs Minimum of 8 GB RAM available.

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 training provides preliminary suggestions for hardware planning.
- OpenText recommends that you engage Professional Services at this phase in order to create a custom Project plan.

Configuration and Project development

- Install and configure the Communications Server application.
- Scope and develop Exstream 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.

Testing and analysis

- Integration, performance, and stress tests under various scenarios.
- Verification of availability and disaster recovery.
- Detect and correct most significant issues in test systems prior to go-live.

Re-optimization

- Periodic system reevaluations should be performed as customer specifications evolve and new applications come online.
- Resizing is often necessary to compensate for changes in network topology, business processes, or increased user load.

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 Exstream 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 Exstream volumes are as follows:

- Large installation: About 10 million documents per month.
- Medium installation: 1-2 million documents per month.

3. Exstream installation

Objectives

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

- Identify the components installed by the OpenText Exstream installer
- Describe the steps of a typical OpenText Exstream installation
- Install OpenText Exstream Runtime and Exstream Design
- Install OpenText Exstream Live Editor

Installation components

The Exstream platform and related software packages are available for download from My Support.

You must download and use the Runtime and Design Tools common installers to install or upgrade Exstream on the Windows and Linux platforms. You must also separately download and install OpenText Directory Services.

Depending on your Exstream implementation, you might need to install additional Exstream components that are not included in the common installers. You can download installation packages for additional Exstream components from My Support.

Runtime components

The Exstream Runtime installer includes the server environment for Exstream and the necessary tools to operate the environment. You can install the runtime components on Linux or Windows.

Empower Server and Empower Editor are not included in the Runtime installer. For information about how to install Empower, see Installing and Upgrading OpenText Exstream Empower Editor.

Content Author is not included in the Runtime installer. For information about how to install Content Author, see Installation and Upgrade Information in Design and Production documentation.

PowerDocs Engine is not included in the Runtime installer on Windows.

The Runtime installer includes the following installation options:

Communications Server

The Communications Server option includes the following components:

- Communications Server
- Framework, including the management gateway and the core platform services.

Exstream Engine

The Exstream Engine option includes the production engine for Design and Production.

Web Applications The Web Applications option includes the following components:

- WorkShop
- Supervisor
- StoryBoard
- ReTouch
- Writer
- Rule Editor
- CAS Browser
- Control
- Communications Orchestrator

Design Tools components The Exstream Design Tools installer includes the tools used to design the customer communications. You can install the design tools on Windows.

The Design Tools installer includes the following installation options:

Design Manager The Design Manager option includes the following components:

- Design Manager
- Designer
- Logic Designer

Communications Builder The Communications Builder option includes the following components:

- Communications Builder
- StoryTeller
- TDTransformation
- Windows Driver Tool
- UTF Edit

Control Center The Control Center option includes the following components:

- Control Center
- Describer

Additional Exstream components

Depending on your Exstream implementation requirements, you might require additional platform components that are not included in the common installers. You must download installation packages for the following platform components separately from My Support:

Design and Production engines

- AIX 64-bit SBCS and DBCS engine ZIP files
- HPUX 64-bit SBCS and DBCS engine ZIP files
- Linux 64-bit SBCS and DBCS engine ZIP files
- Solaris 32-bit SBCS and DBCS engine ZIP files
- Windows 64-bit SBCS and DBCS engine ZIP files l z/OS 32-bit SBCS and DBCS engine ZIP files
- z/OS 64-bit SBCS and DBCS engine ZIP files

Exstream Content Author

Content Author WAR file

Exstream Empower

- Empower Editor ZIP file
- Empower Server ZIP file

PowerDocs Engine

PowerDocs server component WAR files (for Windows)

PowerDocs Editor

PowerDocs EXE file

PowerDocs Designer

- PowerDocs Designer EXE file
- PowerDocs Content Designer EXE file
- PowerDocs eSignature Client EXE file

Exstream installation roadmap

Complete the tasks described below to install and configure the Exstream platform. After you complete these tasks, you can configure Design and Production for your specific business requirements.

1. Use the Design Tools common installer to install the design environment for the Exstream platform.

   This step installs the components required to design customer communications. These components are installed on Windows systems.

2. Use the Runtime common installer to install the server environment for the Exstream platform.

   This step installs the server environment required for engine orchestration and output delivery.

3. Install and configure OpenText Directory Services (OTDS).

   OTDS is used for user management in the integrated Exstream platform. This step requires you to set up OTDS partitions, groups, and tenants for use with Exstream.

4. Configure the Exstream server environment.

   To use Design and Production in the Exstream platform, you must install and configure all of the components of the Exstream server environment. This step requires you to set up and configure Exstream repositories and services and connect them to your OTDS instance.

5. Create design and tracking databases for use with Design and Production.

   Design databases are used to store information about system configurations and design objects and are required to use Design and Production. Tracking databases are used to store tracking information that is generated when you run the Exstream engine to produce output.

6. Sign in to Design Manager and set up your licensing information.

   You must use your license key to sign in to Design and Production for the first time. If you are using a specific licensing method, you must also set up your licensing server and add the appropriate license files.

7. Download and install additional Exstream components, if required.

   If your Exstream implementation requires additional Exstream components that are not installed using the common installers (for example, Exstream Live, Exstream Empower, Exstream Content Author, or additional Exstream production engines), you must download and install these components separately.

Steps before the installation

This section describes the steps you can follow to install Exstream software using the Exstream Runtime and Design Tools installers.

- Confirm that you have the software and hardware required for each Exstream component you want to install. For information about supported software and versions, see OpenText Exstream 16.6 Release Notes.
- Download the Runtime and Design Tools installers from the Exstream product area on OpenText My Support (https://support.opentext.com).

Installation steps

The procedures below describe the high-level installation steps. The order of the steps may differ in your environment depending on the specific requirements of the Exstream solution, the Exstream components you will run, the hardware used, and whether you want to install in different environments, such as development, testing, and production.

To install Exstream Runtime

1. Install Communications Server on the computers you want to use to run Communications Server and other types of Communications Server applications.
2. Typically you install Communications Server on several computers, such as development, testing, and production computers. In production environments, you may also install and run Communications Server on more than one computer for scalability and performance reasons.
3. You can install and run Communications Server on the same computer as other Exstream components or on a separate computer.
4. Install Exstream Engine on the computers you want to use to process Design and Production applications.
5. To enable Communications Server to invoke Exstream Engine, you must install Communications Server before you install Exstream Engine. You must then install Exstream Engine on the same computer as Communications Server.
6. Install Web Applications on a computer from which you can access the computer that runs the Java Application Server or web server.
7. Install PowerDocs Engine on a computer from which you can access the computer that runs the Java Application Server or web server.
8. You can install Web Applications and PowerDocs Engine on the same computer as other Exstream components or on separate computers. You must run the web applications on the same computer as a supported Java Application Server or web server and a supported Java Runtime Environment. Depending on the type of installer and the accessibility of the Java Application Server or web server, you might be able to deploy the web applications directly from the installer.

To install Exstream Design Tools

1. Install Communications Builder on the computers used to develop and test Communications Builder Projects.
2. You can install and run the design tools on the same computer as other Exstream components or on a separate computer. In production environments, the design tools are typically installed on other computers than the production computers.
3. Install Design Manager on the computers used to develop and test Design and Production applications.
4. To enable Design and Production to connect to the CAS, you must install Communications Builder before you install the Design Manager component. You must then install Design Manager on the same computer as Communications Builder.
5. Install Control Center on the computers used to administer the environment. You can install and run Control Center on the same computer as other Exstream components or on a separate computer.

Exstream bootstrap

The Exstream bootstrap consists of a set of scripts and configuration files that allow you to install and set up the following OpenText Exstream components:

- Communications Server on either Microsoft Windows or Linux platforms
- Communications Builder, Control Center, and Describer on Windows

Using the bootstrap with the default settings creates an environment with the following configuration:

- Local installation of Communications Server.
- Local installations of Communications Builder on Windows, which includes Control Center, StoryTeller, and Describer.
- Database schemas that include all the repositories.
- Single tenant and one application domain.
- A metadata model imported into the tenant repository, which contains the types and properties used in the InteractiveStatement application.
- A Communications Server application called InteractiveStatement.
- A release package in the CAS, deployed to a running InteractiveStatement demo application (on Windows only).
- A running Service Gateway.
- StoryBoard, WorkShop, ReTouch, and Supervisor Web applications deployed and running on a local Apache Tomcat.

It is also possible for you to configure the bootstrap to set up the environment in different ways.

To provide different options for installing the prerequisite software, there are three bootstrap packages available for each platform. Depending on which package you use, you can:

- Install Exstream, OpenText Directory Services (OTDS), Apache Tomcat, and PostgreSQL
- Install Exstream, OTDS, Tomcat, and connect to an existing database
- Install Exstream and connect to existing OTDS, Tomcat, and database installations

By using the bootstrap you can quickly set up and get started with Exstream in a test or demonstration environment. After the bootstrap is run, you can starting working in the Web applications, Communications Builder, Control Center, and Describer.

Lab: Install the Exstream Runtime

The Runtime installer on Windows includes the components below. For detailed information, see “Runtime components” on page 10. You can either run a typical install, which installs all components on one computer, or you can run an advanced install and select the components to install.

- Communications Server
- Exstream Engine
- Web Applications: The Web application ARchive (WAR) files are installed in the directory below. After the installation, you must deploy the WAR files to a supported Java Application Server or web server.

  <Installation directory>
  \<Version>\Server\solutions\management\web

Prerequisites

- To install or remove software on Windows, you must have administrator rights.
- Before you install Communications Server, you must close all web browsers and stop any Java Application Server or web server running on the computer.
- You must install Communications Server before you install Exstream Engine. You must then install Exstream Engine on the same computer as Communications Server. (If you run a typical install, the components are automatically installed in the correct order.)

Install the Exstream Runtime

1. *In Windows Explorer navigate to* ***C:\Training\16.6\Software\16.6.0_GA_2019-04-03_Build_443_RUNTIME*** *and launch* ***ExstreamSetup.exe*** *as Administrator.*
2. *In the User Account Control popup click* ***Yes****.*

3. *Select **Advanced Install** and click the **Next** button.*

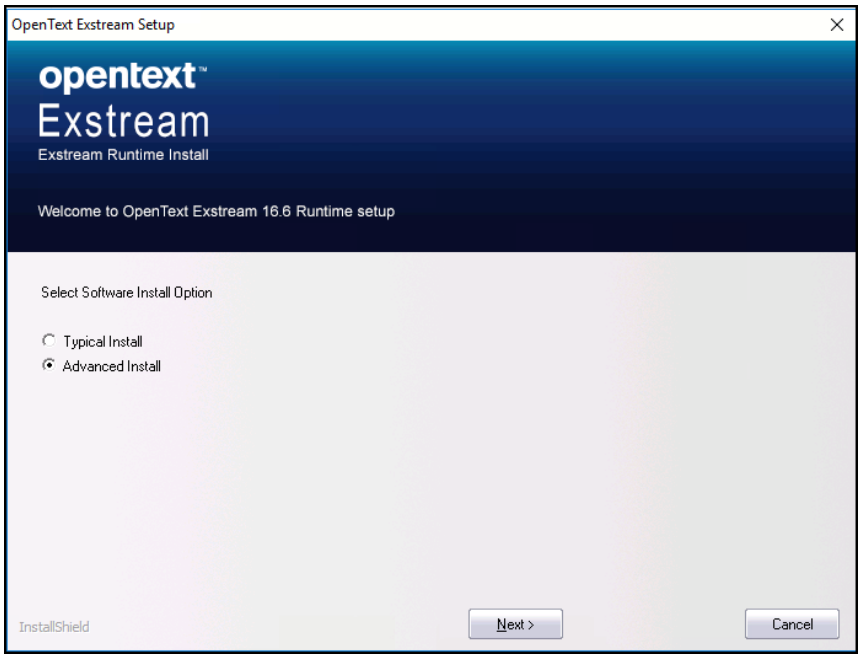


Figure 3-1:

Installation type

4. *Select all the features and click **Next**.*

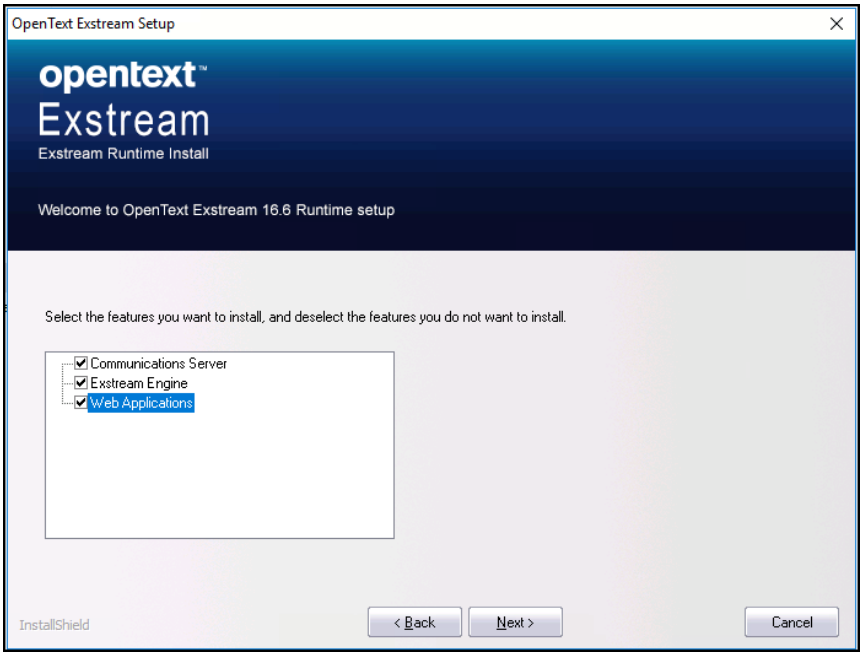


Figure 3-2:

Features to install

5. *Set the installation folder to **C:\OpenText\Exstream** and click **Next**.*

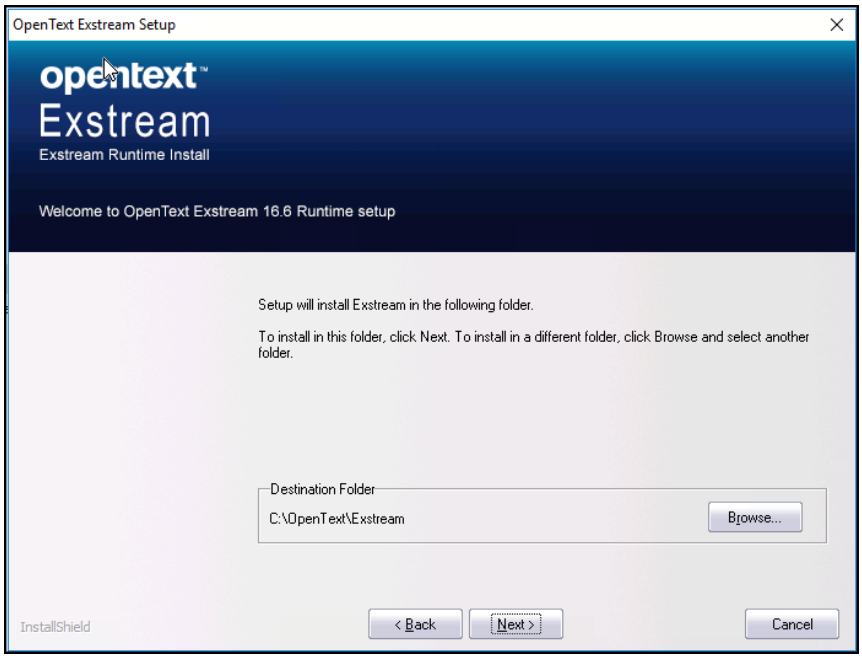


Figure 3-3:

Exstream installation folder

6. *Accept the default installation folder for the Management Gateway and click **Next**.*

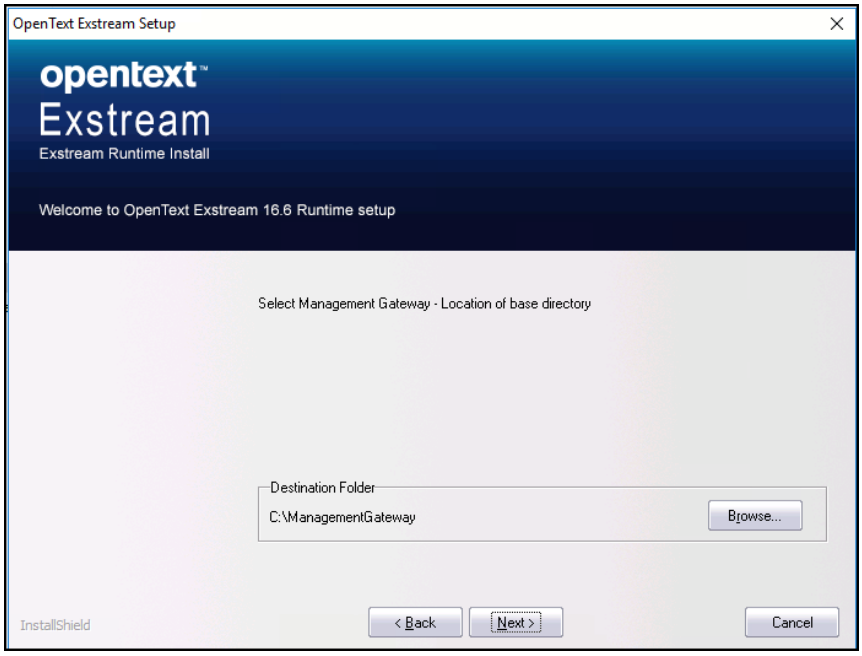


Figure 3-4:

Management Gateway installation folder

7. *Accept the default Java installation folder and click **Next**.*

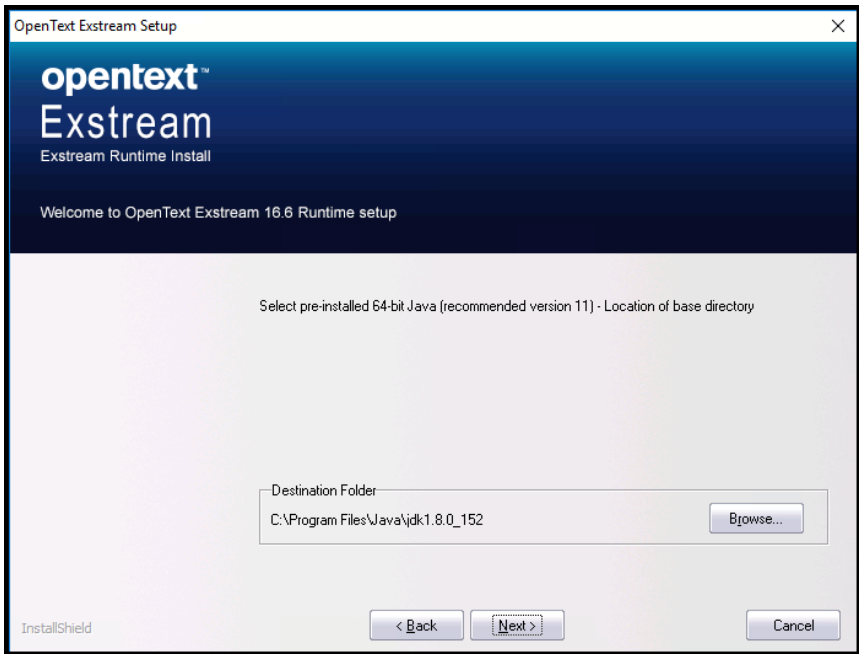


Figure 3-5:

JVM installation folder

8. *Click the **I Agree** button.*

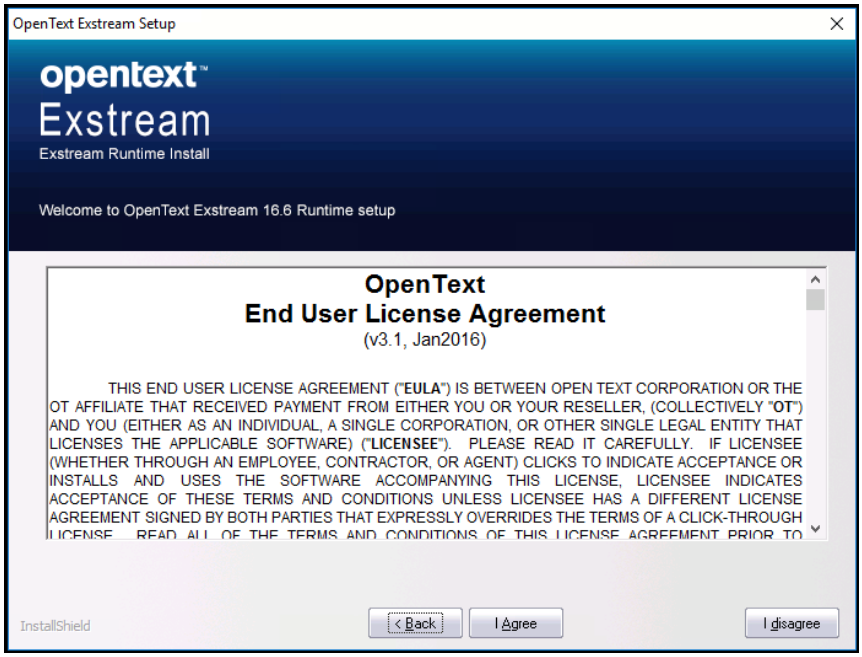


Figure 3-6:

License Agreement

9. *Click **Install**.*

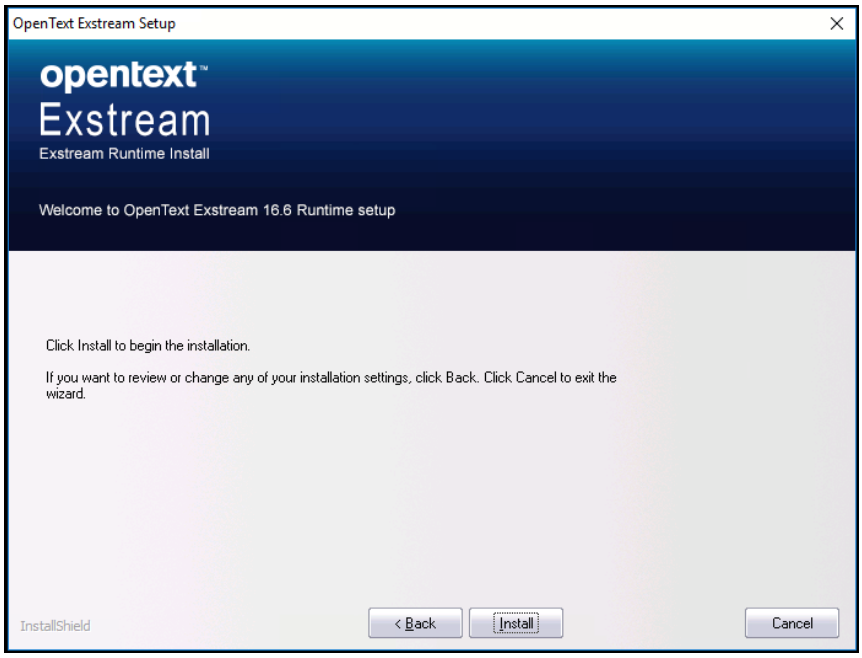


Figure 3-7:

Ready to install

10. *Once the installation completes, click the **Finish** button.*

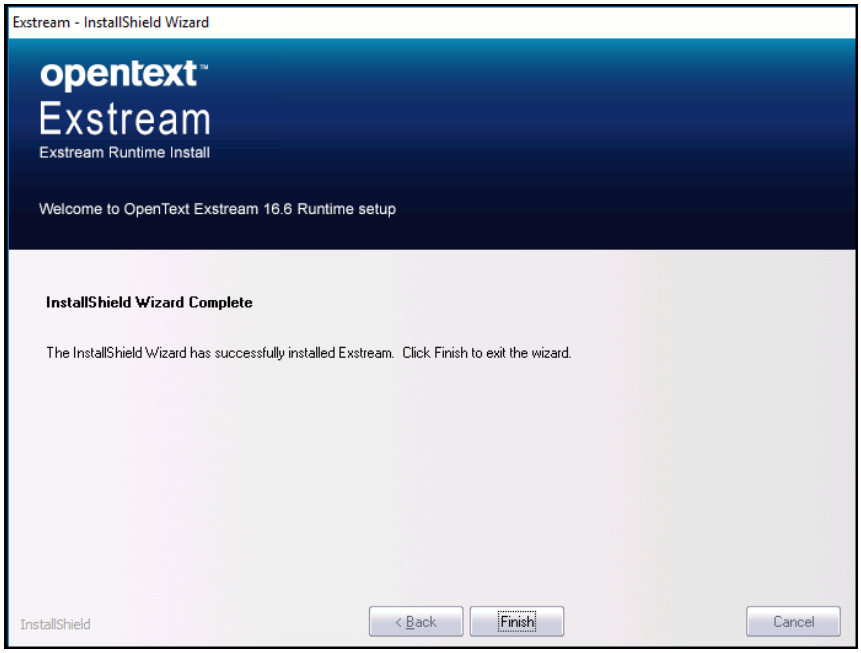


Figure 3-8:

Installation complete

Lab: Install the Design Tools

The Design Tools installer includes the components below. You can either run a typical install, which installs all components on one computer, or you can run an advanced install and select the components to install:

- Design Manager
- Communications Builder
- Control Center

Communications Builder and Control Center require third-party software. During the installation, the installer does a prerequisite check and installs the missing software automatically.

Prerequisites

- To install or remove software on Windows, you must have administrator rights.
- Windows Internet Explorer must be installed on the computer where you install Communications Builder.
- You must install Communications Builder before you install the Design Manager component. You must then install Design Manager on the same computer as Communications Builder.

If you run a typical install, the components are automatically installed in the correct order.

Before starting the next activity wait a couple of minutes to make sure the previous installation has completed all its processes.

Install the Exstream Design Tools

1. *In Windows Explorer navigate to* ***C:\Training\16.6\Software\16.6.0_GA_2019-04-03_Build_443_DESIGN*** *and launch* ***ExstreamSetup.exe*** *as Administrator.*

2. *Select **Advanced Install** and click the **Next** button.*

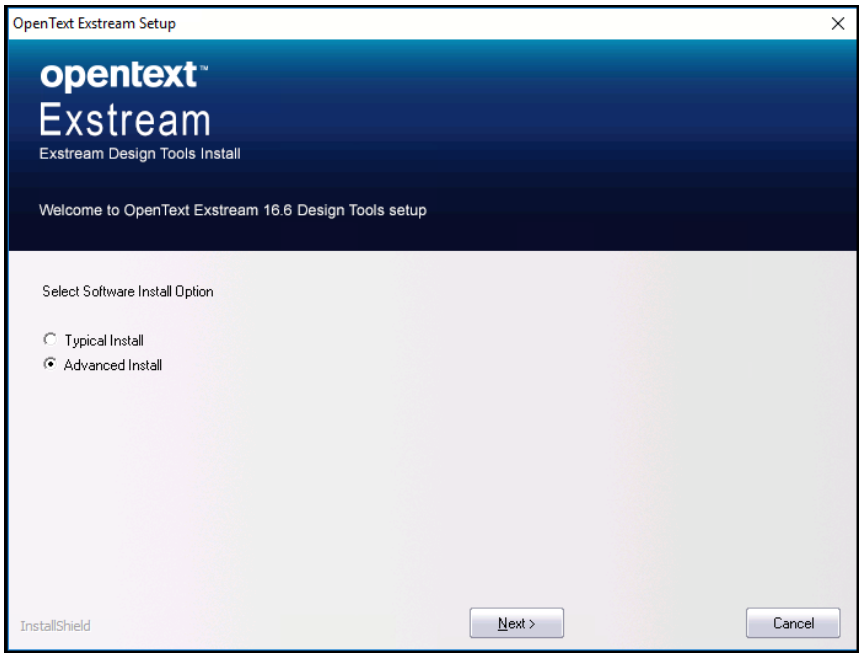


Figure 3-9:

Installation type

3. *Select all the features and click **Next**.*

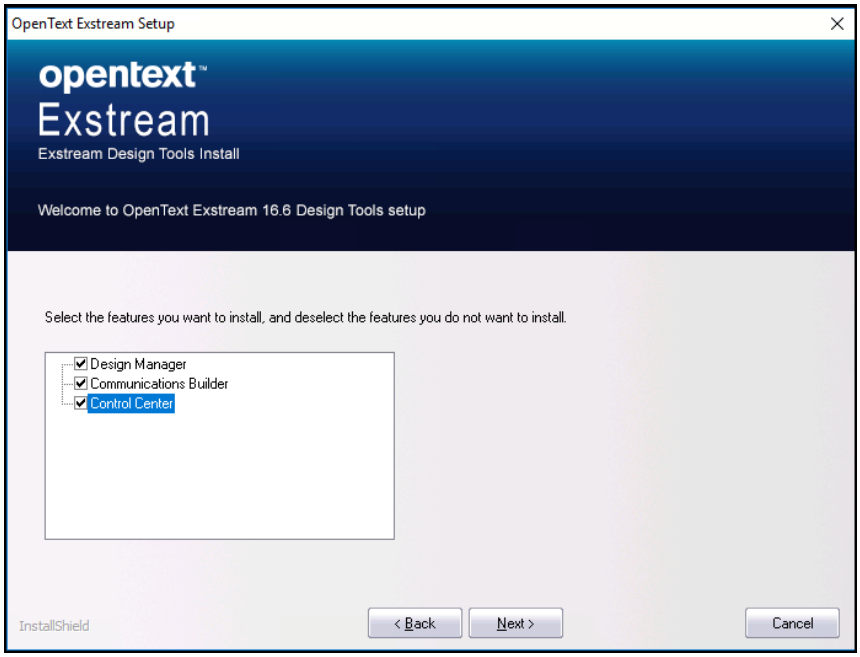


Figure 3-10:

Features to install

4. *Set the installation folder to **C:\OpenText\Exstream** and click **Install**.*

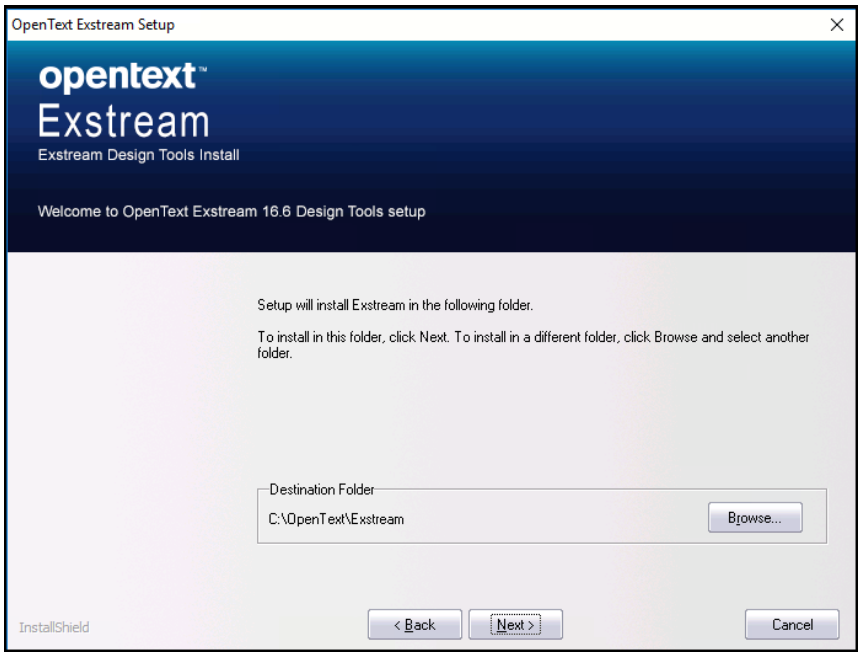


Figure 3-11:

Installation folder

5. *Accept the default Java installation folder and click **Next**.*

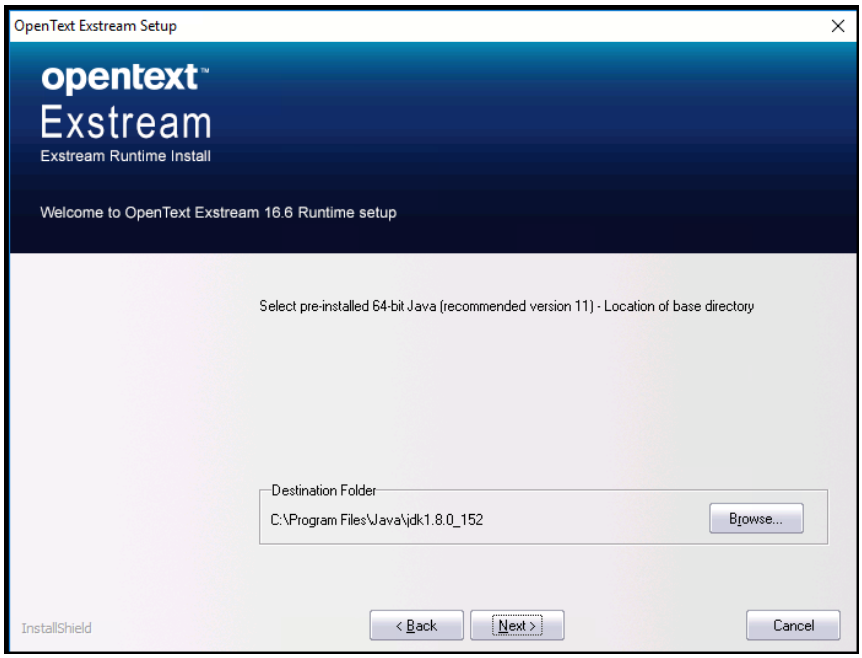


Figure 3-12:

JVM installation folder

6. *In the OpenText End User License Agreement click the **I Agree** button.*

Figure 3-13:

License Agreement

7. *Click **Install**.*

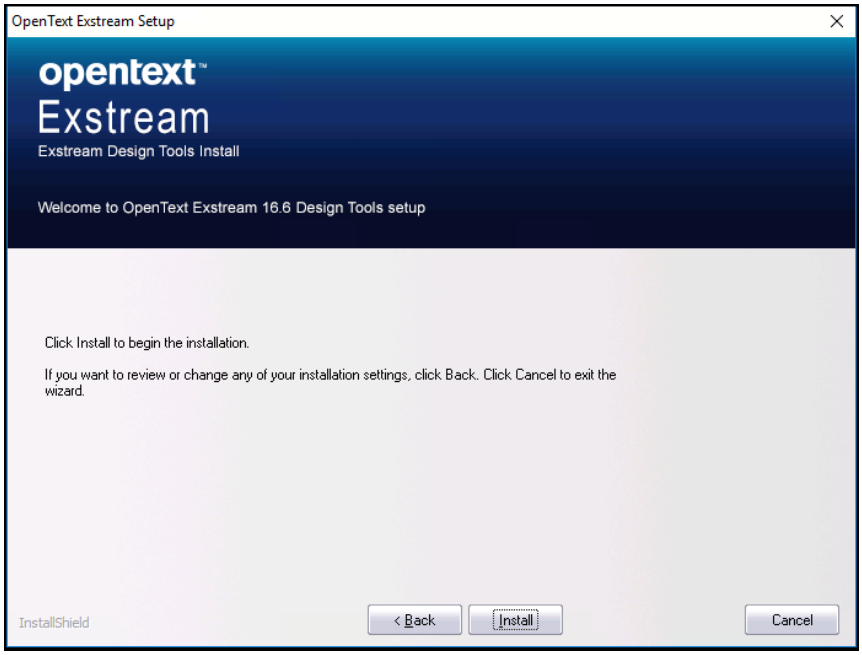


Figure 3-14:

Install

8. *Once the installation completes, click the **Finish** button.*

Install the Single Byte Engine

This is an optional step, and it is required only if you are planning to use a single byte engine.

1. *Copy the **engine_16_6_0_windows_64_sb** folder:*
   - *From: **C:\Training\16.6\Software***
   - *To: **C:\OpenText\Exstream***

The next is a step to facilitate the training.

Merge the Exstream program groups

1. *Navigate to Programs, right-click **OpenText Exstream 16.6** and select **Open**.*

The content of the OpenText Exstream 16.6 program group opens in Explorer.

2. *Navigate to Programs, right-click **OpenText Exstream 16.6.0** and select **Open**.*

The content of the OpenText Exstream 16.6.0 program group opens in Explorer.

3. *Drag-and-drop the 4 shortcuts from **OpenText Exstream 16.6.0** window on the **OpenText Exstream 16.6** window.*
4. *Navigate to Programs, right-click **OpenText Exstream 16.6.0** and select **Delete**.*

Now, all the Exstream applications are listed in the OpenText Exstream 16.6 program group.

4. Setting up OTDS for Exstream

Objectives

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

- Define some basic Access control concepts
- Describe the role of OTDS in an OpenText Exstream installation
- Identify the OTDS options and configuration recommendations
- Identify the requirements for Exstream tenants
- Configure a multi-tenant OTDS
- Create a tenant
- Configure an OTDS tenant

Access control concepts

This chapter describes the steps to set up OTDS for Exstream.

Each Exstream tenant has its own users

One Exstream environment can have a single tenant or can have multiple tenants. A tenant can be a customer, company, or organizational unit that has its own set of users. When users log on as a tenant, their view of the system is based on their tenant. Users from one tenant cannot see or Access another tenant's applications or data.

In the image below, John, Robert, and Amy from Tenant A have Access to the applications, repositories, and data that belong to Tenant A. While Mary, Sam, and Fiona have Access to the applications, repositories, and data that belong to Tenant B.

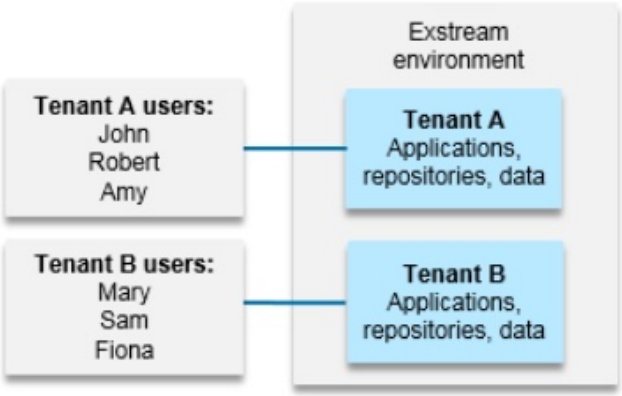


Figure 4-1:

Each Exstream tenant has its own users

Roles control Access and permissions

The users for each Exstream tenant are assigned to one or more roles. The roles control the following:

- Which desktop applications and web applications users can Access.
- Which permissions users have in each application.

For example, if John is assigned to the tenant administrator role, John can Access Control Center, Communications Builder, Describer, and the web applications. Whereas, if Robert is assigned to the Reviewer role, Robert can only Access the web applications.

Tenant OTDS

The users for one tenant are created in a component called the tenant OTDS. This component is created in OTDS. The tenant OTDS contains one group for each of the Exstream roles: tenant administrators, tenant users, reviewers, on demand users, and any customized roles specific for the tenant. Users are assigned to these groups in the tenant OTDS.

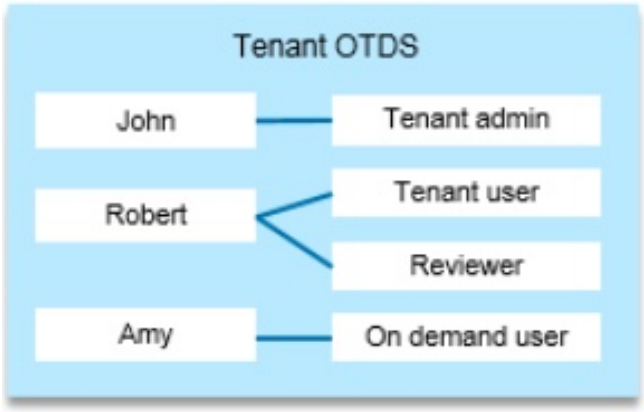


Figure 4-2:

Tenant OTDS

There is one tenant OTDS for each Exstream tenant. When a tenant is added to the Exstream environment, it is connected to its tenant OTDS.

This connection controls who can log on to the tenant's Exstream environment.

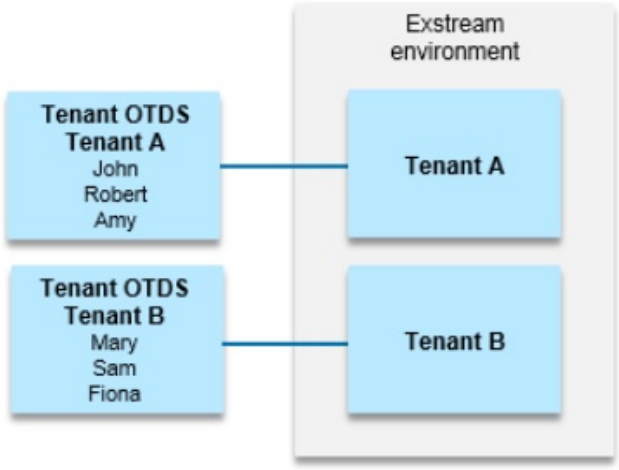


Figure 4-3:

Tenants

Multi-tenant users manage the overall environment

In order to manage the overall environment, there is another group of users who can add tenants via the management gateway. These users cannot see or Access any of the applications, repositories, or data that belong to the tenants.

The image below shows how Linda and Bob have Access to the management gateway, but no Access to either tenant's applications or data.

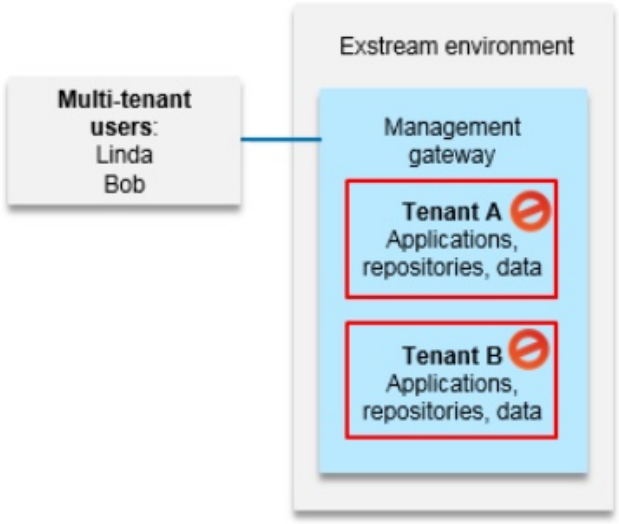


Figure 4-4:

Multi-tenant

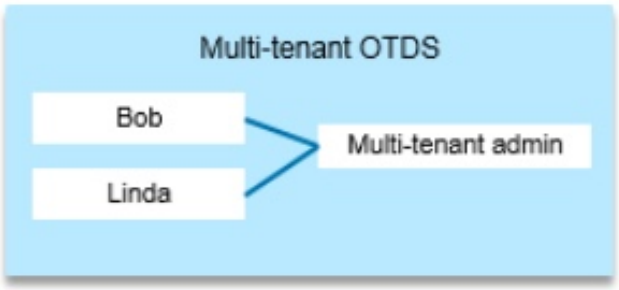
The group of users who manage the overall environment are assigned the multi- tenant administrator role. Multi-tenant administrators only have Access to the Exstream ss_tenantadmin utility.

Multi-tenant OTDS

The multi-tenant administrator users are created in a component called the multi-tenant OTDS. This component is created in OTDS and contains the multi-tenant administrators group. The users who will manage the overall environment must be assigned to this group.

Figure 4-5:

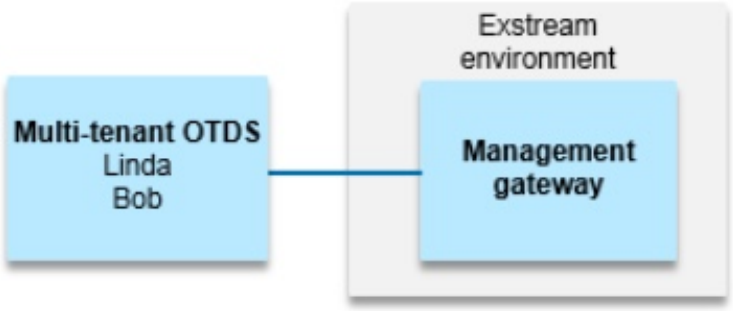
Multi-tenant OTDS



When the management gateway is configured, it is connected to the multi-tenant OTDS. After this, users assigned to the multi-tenant administrator group can use the ss_tenantadmin utility.

Figure 4-6:

Multi-tenant OTDS



Options and recommendations

To set up OTDS for Exstream you need to configure the following components:

- The multi-tenant OTDS, which contains the group for the multi-tenant administrators who manage the overall environment.
- One tenant OTDS for each Exstream tenant. The tenant OTDS contains the groups for the tenant administrators, tenant users, reviewers, on demand users, and any customized groups. The groups can Access the tenant's environment in Exstream via Communications Builder, Control Center, and the web applications.

Understanding the OTDS tenancy concept

OTDS supports running multiple tenant backends in a single OTDS server. Each backend has its own set of OTDS data: resources, user partitions, Access roles, authentication handlers, and system attributes. The default installation of OTDS includes a default backend. Tenant backends can be added to OTDS via the command line interface.

Requirements for Exstream tenants

There are many ways you can set up OTDS for Exstream. However, the following requirements always apply:

- Each tenant in the Exstream environment must always have its own tenant OTDS. The tenant OTDS contains the groups of users that can Access the tenants environments in Communication Center (i.e. tenant administrators, tenant users, reviewers, on demand users, and any customized groups).
- Each tenant OTDS must always be created in a separate OTDS backend or separate OTDS server. It is not possible for two tenants to share an OTDS server or backend. Having a separate OTDS backend or OTDS server for each Exstream tenant is necessary so that users from one tenant cannot log on to another tenant's Exstream environment.
- Each Exstream tenant can only be connected to one tenant OTDS. The tenant OTDS must be created in a single OTDS backend or OTDS server. It is not possible to connect an Exstream tenant to several different OTDS backends or OTDS servers.

Recommended method of setting up OTDS for Exstream

To set up OTDS for Exstream, OpenText recommends that you use an OTDS installation with multiple backends and that you configure OTDS in the following way:

- Configure the multi-tenant OTDS in the default OTDS backend. This backend has its own Exstream partition, resource and users. It will only contain the group for the multi-tenant administrators (OTDS group name: strsmultitenantadmins).
- Configure the tenant OTDS for each Exstream tenant in a separate OTDS tenant backend. This requires that you add a tenant to OTDS for each tenant in your Exstream environment. Each Exstream tenant then has its own OTDS backend with an Exstream partition, resource, users, and groups (i.e. tenant administrators, tenant users, reviewers, on demand users, and any customized groups with the OTDS group names: strstenantusers, strstenantadmins, strsreviewers, strsondemandusers, and <customized groups>).

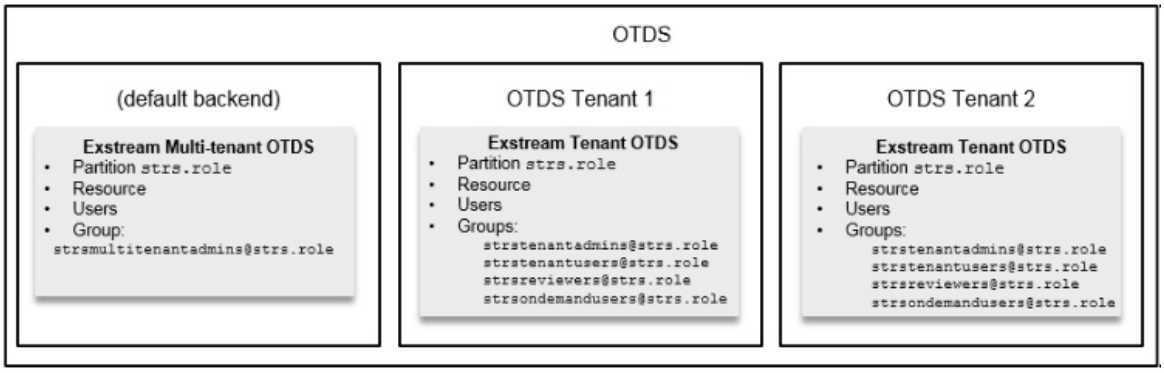


Figure 4-7: Exstream multi-tenant environment

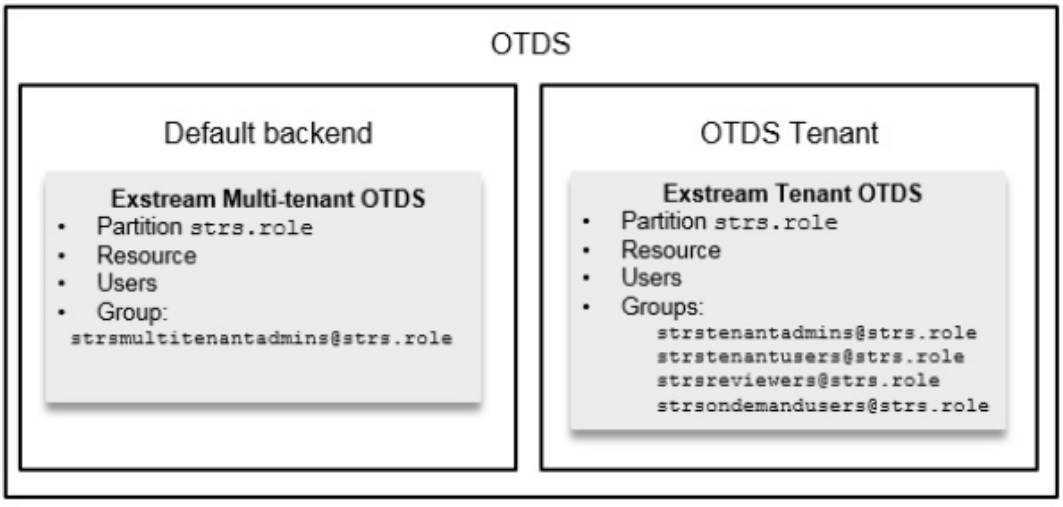


Figure 4-8: Exstream single tenant environment

Alternative options – Exstream environments with multiple tenants

There are other ways you can set up OTDS for Exstream. Some examples of how you can do this in an Exstream environment with multiple tenants include:

- Using separate OTDS servers for each tenant OTDS. In this scenario, each tenant will still have its own Exstream resource, user partition, groups, etc., but these are created in separate OTDS servers.
- Configuring the multi-tenant OTDS in the same backend as a tenant OTDS. In this scenario, you create the multi-tenant administrators in the same backend as the groups for the tenant users, tenant administrators, reviewers, on-demand users, and any customized groups.

Alternative option – Exstream environments with single tenants

In an Exstream environment with a single tenant, you can configure both the multi- tenant OTDS and tenant OTDS in one backend. In this scenario, all the groups are in one OTDS server or backend.

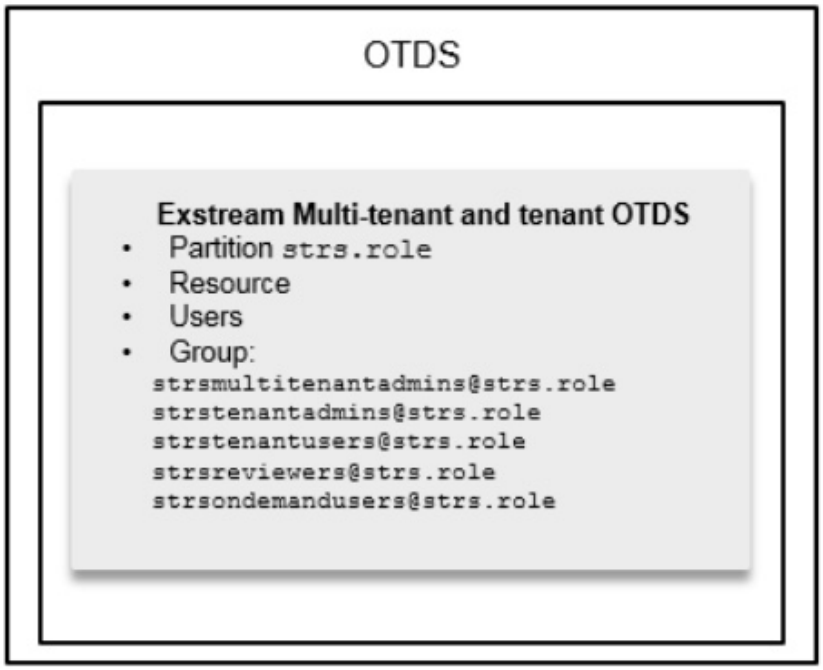


Figure 4-9:

Exstream environments with single tenant

OTDS configuration requirements

This section describes the configurations that are required in an OTDS for Exstream.

Supported OTDS versions

For information about the supported OTDS versions, see OpenText Exstream – Release Notes.

Steps overview

If you use the recommended way of setting up OTDS, you can follow these overall steps:

1. Download OTDS from OpenText My Support and install OTDS. For more information about installing OTDS, see OpenText Directory Services - Installation and Administration Guide (OTDS-IWC).
2. In the default OTDS backend, configure the multi-tenant OTDS (for the multi- tenant administrators).
3. For each Exstream tenant:
   - Add a tenant to OTDS, which adds a tenant backend.
   - In the tenant backend, configure the tenant OTDS (with the tenant administrators, tenant users, etc.).

Required OTDS backend configurations for Exstream

This section describes the configurations that are required in an OTDS backend for an Exstream multi-tenant OTDS or tenant OTDS.

You need to make these configurations in each OTDS backend or OTDS server separately. You can use the OpenText Administration Client or the OTDS web administration client to make these configurations in the default OTDS backend. You must use the web administration client to make these configurations in an OTDS tenant backend. You activate the resource via the OTDS API.

Required configurations

1. You must create a partition with the name:strs.role.
2. You must create the following groups:
   – In the multi-tenant OTDS, create the group:

Groups name	Description
strsmultitenantadmins	Multi-tenant administrators (MTA)

   – In the tenant OTDS, create the groups:

Groups name	Description
strstenantadmins	Tenant administrators
strstenantusers	Tenant users
strsreviewers	Reviewers
strsondemandusers	On-demand users
strsbcausers	Content Author users
strscommdesigners	Communications Designer users
<customized group>	<Customized group>

3. Assign users to groups.

   You must assign users or groups to the Exstream groups.

   It is recommended that the user or users assigned to the strsmultitenantadmins group have “Never expire” as a password policy in OTDS.

   The user name and password for the user assigned to the strsmultitenantadmins group is required to configure the Exstream environment.

4. Create the Exstream resource.

   You must create and activate a resource for Exstream. The resource can have any name.

   The resource ID and secret key are required to configure the Exstream environment.

5. Assign the strs.role partition to the resource.

   You must create and activate a resource for Exstream. The resource can have any name.

6. Create a browser user.

   You must create a user with read Access to OTDS. This user must be created in the strs.role partition but does not need to be a member of any group. This user must have Never expire as a password policy in OTDS.

Design and Production and CAS Browser requirements

For information about how to add users and groups from OTDS to Design and Production and the CAS Browser, see Adding External Users to a Design Group in OpenText Exstream Design and Production System Administration in the Exstream Design and Production documentation set.

HTTPS and HTTP communication

By default, Exstream uses HTTPS to communicate with OTDS. To use unsecure HTTP communication, you must disable HTTPS in both OTDS and in Exstream.

To disable HTTPS communication in Exstream, you use the -unsecure flag when configuring the connection profiles for OTDS.

To disable HTTPS in OTDS, you must add the system attribute directory.auth.EnforceSSL and set the value to false on the System Attributes page in the OTDS web client.

Assigning Access roles

You assign users and groups to the Exstream Access roles in OTDS.

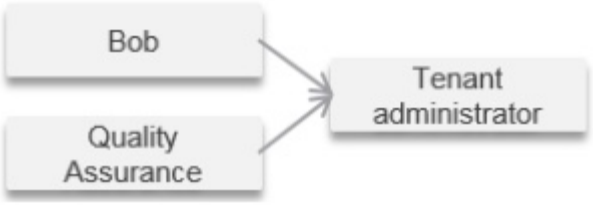
After OTDS is set up for Exstream, tenants can use the OTDS web administration client to manage their users and assign Access roles.

About assigning roles

The Exstream Access roles are represented as groups in OTDS. In OTDS, it is possible to assign one or more users, groups, or organizational units to each Exstream group. For example, you can assign both the user Bob and the organizational unit Quality Assurance to the tenant administrator group.

Figure 4-10:

Several members assigned to one role



You can also assign one user, group, or organizational unit to several roles.

Permissions are exclusive for each role

The permissions for the Access roles are exclusive to the roles. Permissions are not inherited by other Access roles. For example, tenant user permissions are only available to users with the tenant user role, and tenant administrator permissions are only available to users with the tenant administrator role.

If a user is assigned several roles, the user has permissions according to all the assigned roles.

In WorkShop (in the Unversioned resources view), you can control which domains the OTDS group behind a role can Access. For example, you can provide a group Access to the development domain, but prevent the group from Accessing the production domain. When a user logs in to a domain, the user will have the sum of the permissions according to all the assigned roles with Access to the current domain.

Role descriptions

This section describes the default Exstream roles and their permissions.

If you have added customized groups to the strs.role partition in the tenant OTDS, you must use the Supervisor application to create the corresponding roles and assign the roles the appropriate permissions. In Supervisor, you can also change the permissions for the default Exstream roles for the tenant.

Multi-tenant administrator (MTA)

Full Access to the ss_tenantadmin command line utility for adding tenants and updating tenant information, etc.

No Access to Communications Builder, Control Center, Describer, or the web applications.

Tenant administrator

Full Access to Communications Builder, Control Center, Describer, and the ss_territory, ss_scm, ss_deploy, and ss_rcp command line utilities.

Access to web applications:

- **WorkShop** – Access to all views. For example, can publish themes and delete resources in the Resources view.
- **Supervisor** – Access to all views, but cannot edit documents in the Review view.
- **StoryBoard** – For example, can preview themes.
- **ReTouch** – For example, can preview documents.

Communications Designer and Content Author

When you set up OpenText Directory Services (OTDS) authentication for Exstream, you must also set up access roles that are used to manage user permissions. You can use Supervisor to manage the default Exstream access roles or to create new access roles. To use Communications Designer or Content Author, users must be assigned to an Exstream access role that provides the required permissions.

For example, if you are a tenant administrator, you can access both Communications Designer and Content Author, and you will be able to see both widgets on the Content Launcher page. However, if you are assigned only the Content Author user role, then you will not be able to access Communications Designer and will see only the Content Author widget on the Content Launcher page.

The following table lists the OTDS groups for each default Exstream access role and the corresponding permissions for Communications Designer and Content Author:

Role	OTDS group	Communications Designer access	Content Author access
Content Author user	strsbcausers	No access	Full access
Communications Designer user	strscommdesigners	Full access	No access
On-demand user	strsondemand	No access	No access
Reviewer	strsreviewer	Partial access	No access
Tenant user	strstenantusers	Full access	Full access
Tenant administrator	strstenantadmins	Full access	Full access

Tenant user (Tenant developer in the web applications)

Full Access to Communications Builder.

Some Access to the following applications:

- **Control Center** – For example, starting and stopping applications, and viewing properties and logs.
- **Describer** – Viewing models.

Access to web applications:

- **WorkShop** – Access to all views. For example, can edit and check in themes in the Resources view.
- **Supervisor** – Access to all views except the Roles view.
- **StoryBoard** – Full Access.
- **ReTouch** – Full Access.

Reviewer (Document reviewer in the web applications)

No Access to Communications Builder, Control Center, Describer, or the command line utilities.

Access to web applications:

- **WorkShop** – Some Access to the Resources view. For example, can review and approve resources, but cannot edit or delete resources.
- **Supervisor** – Some Access to the Review view. For example, can review and approve documents, but cannot edit or delete documents.
- **StoryBoard** – For example, can preview themes.
- **ReTouch** – For example, can preview documents.

Content Author user

This group has the same access permissions in the web applications as the tenant user, however it can only view approved resources in the CAS Browser.

Communication Designers

This group has access to the Communications Designer application.

On-demand user (Document composer in the web applications)

No Access to Communications Builder, Control Center, Describer, or the command line utilities.

Access to web applications:

- **WorkShop** – Some Access the Resources view. For example, can examine images and texts.
- **Supervisor** – No Access.
- **StoryBoard** – No Access.
- **ReTouch** – Full Access.

Setting up OTDS for Exstream

Setting up OTDS for Exstream requires the following steps:

1. Install and configure Tomcat.
2. Install OTDS.
3. Configure the multi-tenant OTDS.
4. Add a tenant to OTDS.
5. Configure the tenant OTDS.

Lab: Configure a multi-tenant OTDS

OTDS has already been installed in the training computer using the default configuration and has been deployed to Tomcat (in C:\Tomcats\OTDS).

Configuring a multi-tenant OTDS comprises the following steps:

1. Create the strs.role partition.
2. Create the Exstream resource.
3. Create the multi-tenant administrators group.
4. Create a multi-tenant administrator and assign it to the multi-tenant administrators group.
5. Create a read-only (browser) user.
6. Configure the Access roles.
7. Set the trusted site.

Create the strs.role partition

1. *In Chrome navigate to* ***https://thecompany.com:8443/otds-admin*** *(this URL is bookmarked under OTDS > Multitenant OTDS).*

The OpenText Directory Services login windows displays.

2. *Log in to Directory Services using the following credentials:*
   - *User name:* ***otadmin@otds.admin***
   - *Password:* ***opentext***

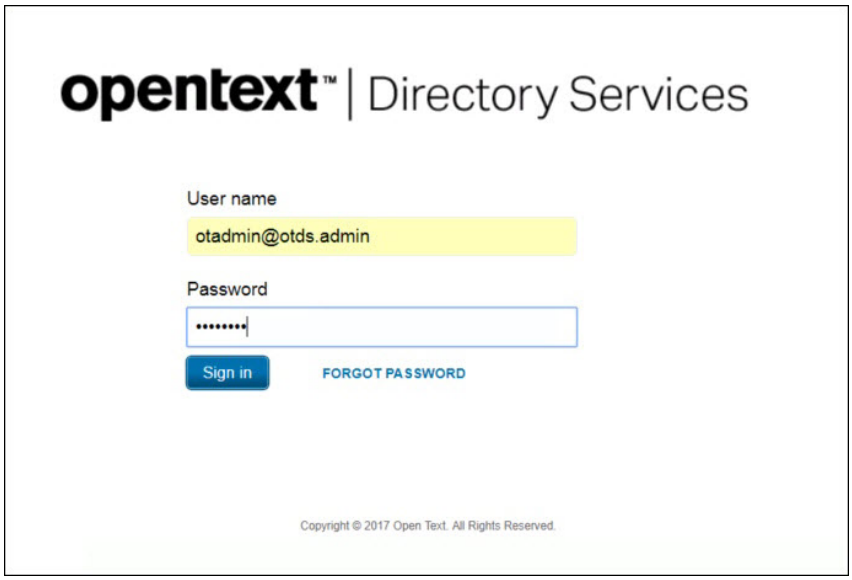


Figure 4-11:

OTDS login page

3. *Select **Partitions > Add > New Non-synchronized User Partition**.*

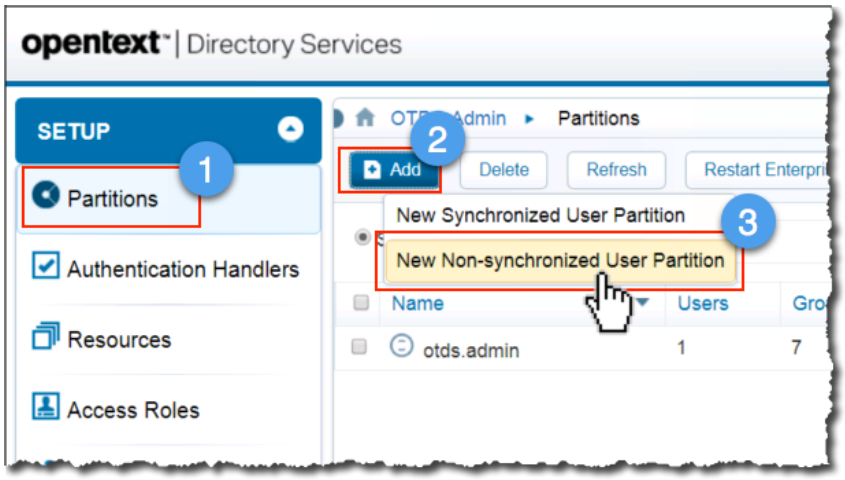


Figure 4-12: Partitions

4. *Name the new partition **strs.role** and click **Save**.*

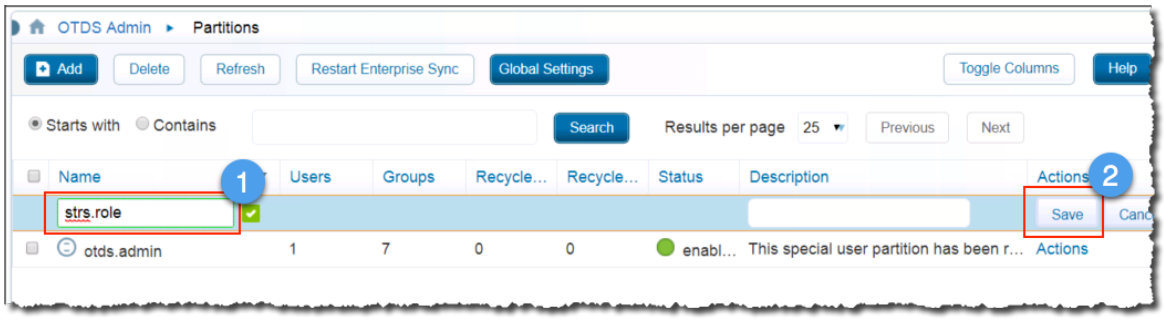


Figure 4-13: Save

The strs.role partition is added to the partition list.

Create the Exstream resource

1. *Select **Resources > Add**.*

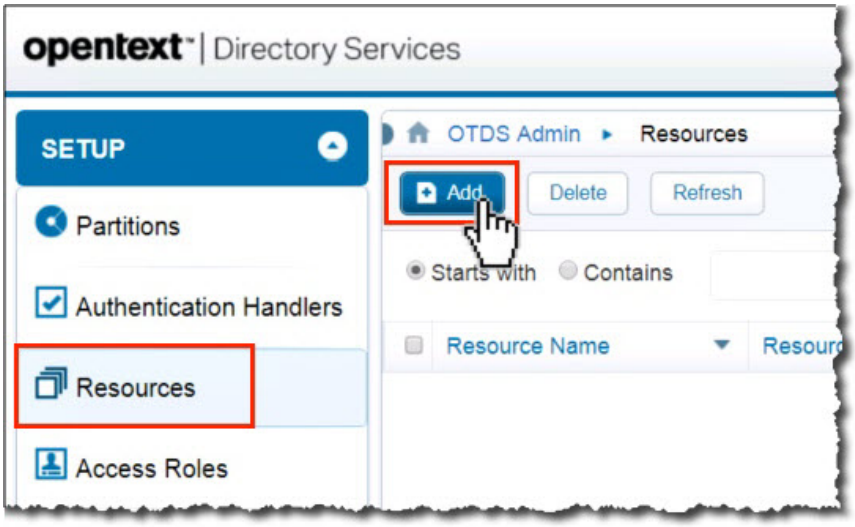


Figure 4-14:

Add resource

2. *Name the new resource **ExstreamResource** and click the **Next** button twice (notice the available settings on each screen).*

Figure 4-15:

Resource

3. *Click **Save**.*

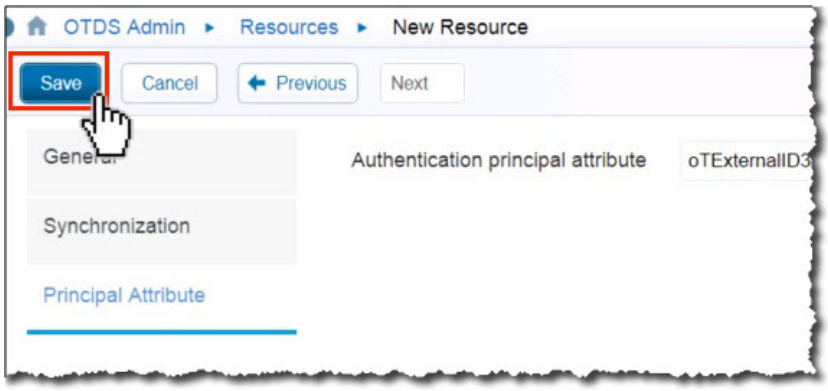


Figure 4-16:

Save resource

4. *Copy the value of the ExstreamResource resource identifier (you will use this value later, you can paste it to Notepad++).*
   - *Resource Identifier: _____*

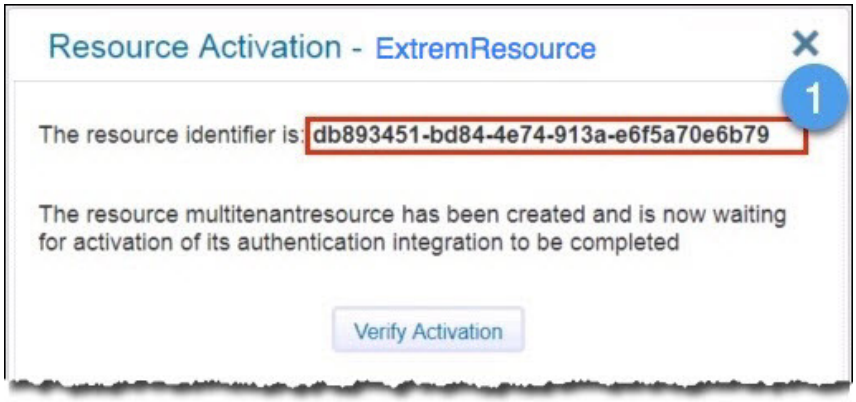


Figure 4-17:

Resource ID

5. *In a new tab in Chrome (WITHOUT closing any previously opened tab) navigate to **http://thecompany.com:8084/otdsws/api/index.html** (this URL is bookmarked under OTDS > OpenText Directory Services REST API).*
6. *Click **resources: Operations on resources > /resources/{resource_id}/activate**.*
7. *Click the **Try it out** button.*

8. *Paste the value of the resource identified in the Resource ID text box and click **Execute**.*

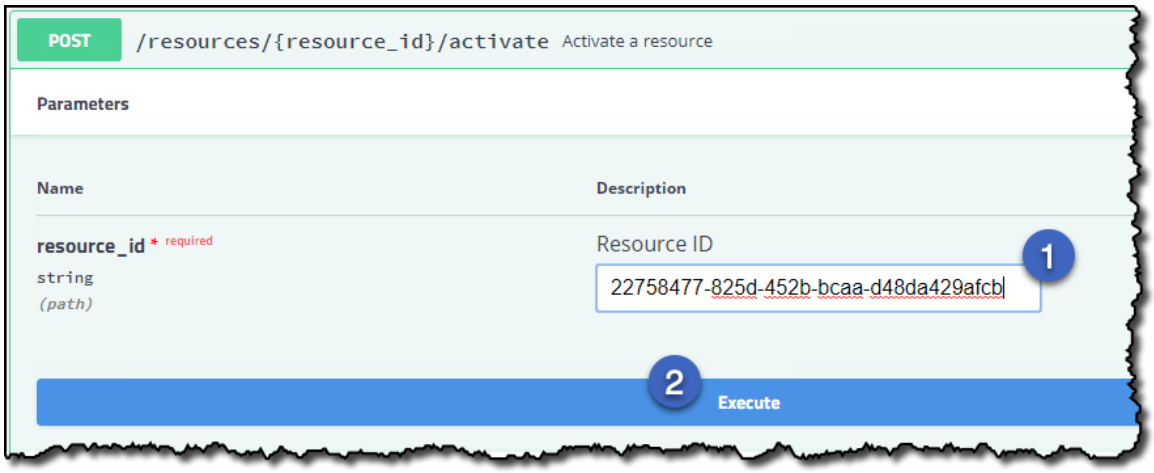


Figure 4-18: Execute

9. *Copy the value of the **secret_key** (you will use this value later, you can also paste it to Notepad++ along with the ExstreamResource resource identifier).*
   - *secret key: _____*

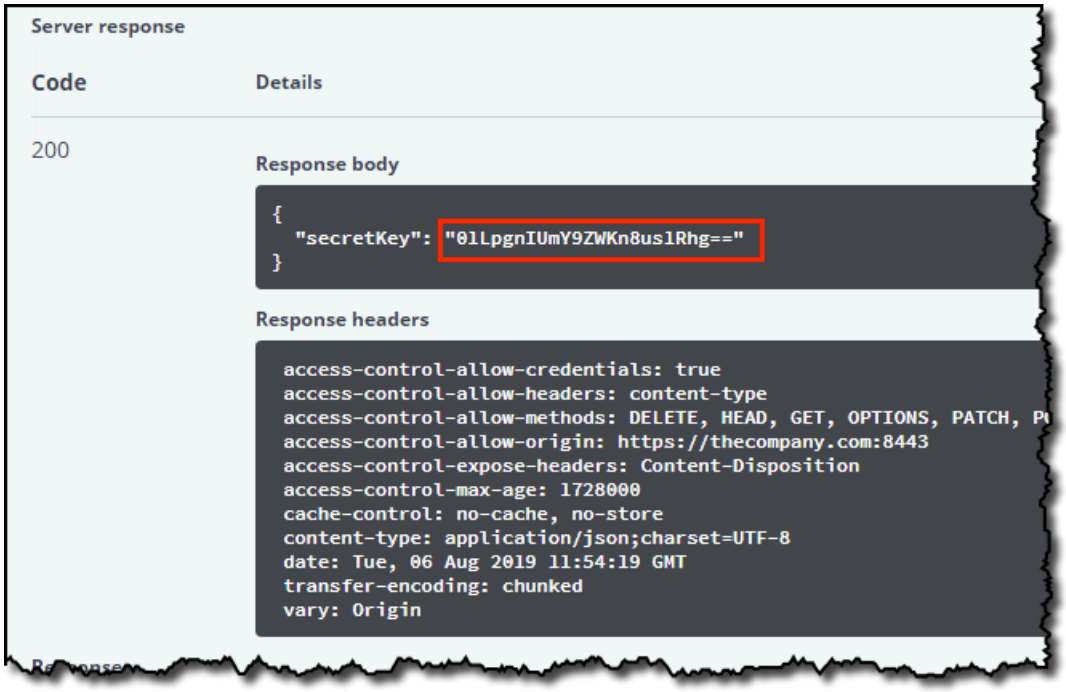


Figure 4-19: Secret key

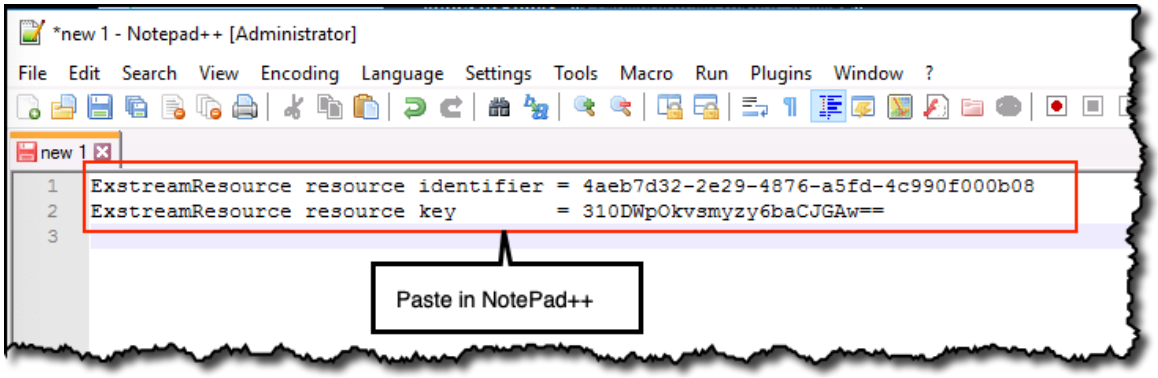


Figure 4-20: Values in NotePad++ for later use

10. *Return to the other Chrome tab (OTDS console) and click **Verify Activation**.*

Make sure you get a “The resource is activated” message.

11. *Click **OK**.*

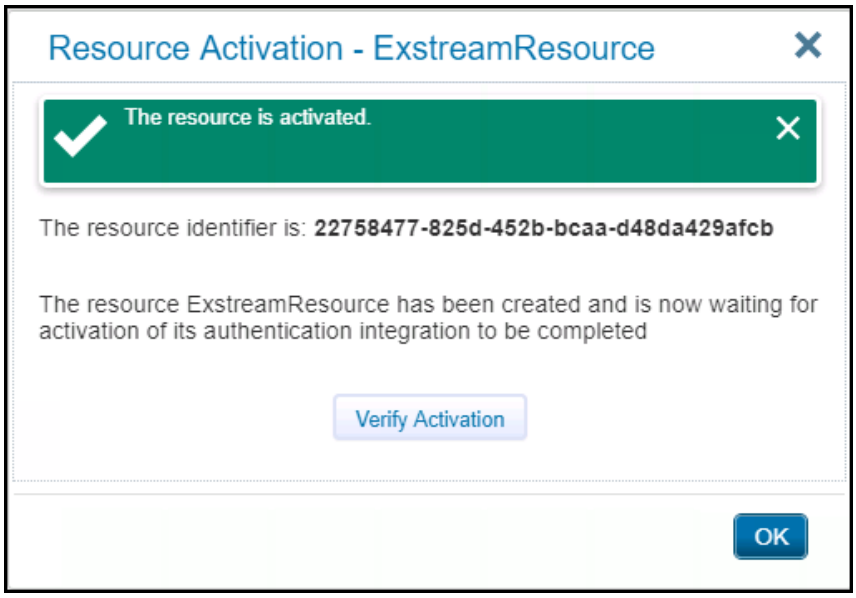


Figure 4-21:

Resource activation verified

12. *You can now close the Chrome tab containing the REST API.*

In the next activity we will modify the default Password Policy to be less restrictive. This is done only for training purposes so that you don’t have to be entering complex passwords during the exercises.

Create a password policy

1. *In Chrome, navigate to **Partitions**.*
2. *Select **Password Policy** from the **Actions** drop-down for the **strs.role** user partition.*
3. *In the Password Policy - strs.role window, set the following values:*
   - *Use global policy: **Unchecked***
   - *Minimum number of characters: **4***
   - *Minimum number of digits: **0***
   - *Minimum number of symbols: **0***
   - *Minimum number of uppercase characters: **0***
   - *Minimum number of lowercase characters: **1***
   - *Minimum number of changes to previous password: **0***
   - *Number of unique passwords before an old password can be reused: **3***
4. *Click **OK**.*

Create the multi-tenant administrative group and users

1. *In the **Partitions** page select **View Members** from the **Actions** drop-down for the strs.role user partition.*

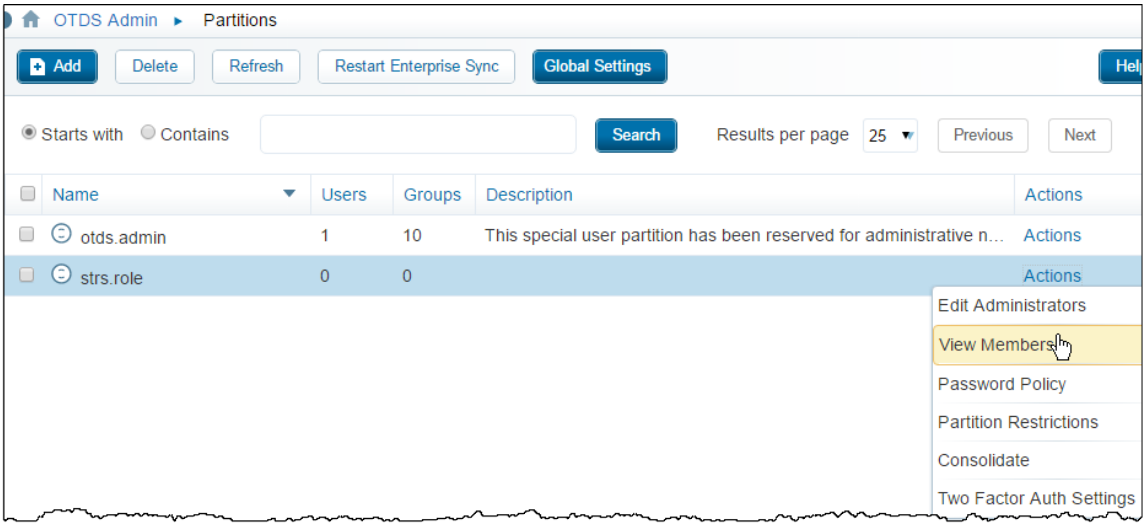


Figure 4-22: View Members

2. *Select **Add > New Group**.*

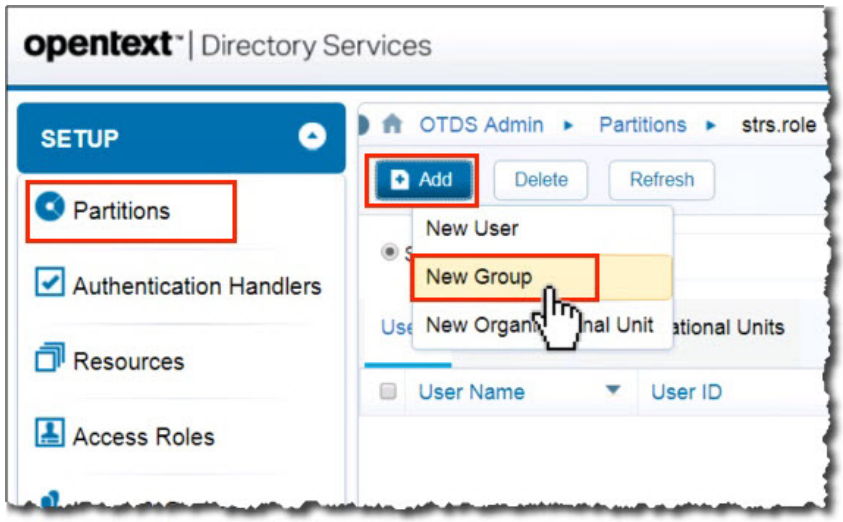


Figure 4-23:

Add new group

3. *Name the new group **strsmultitenantadmins** and click **Save**.*

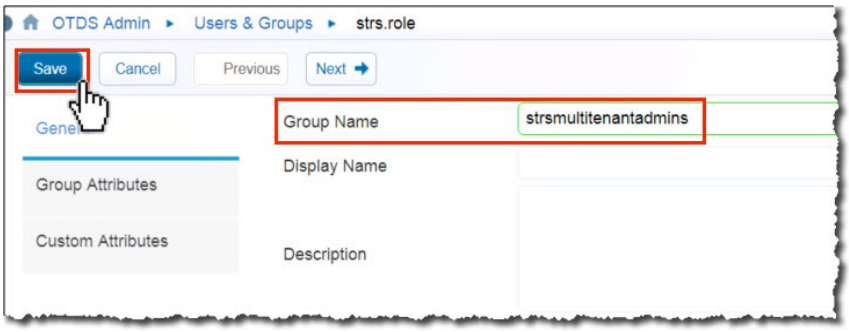


Figure 4-24:

Save new group

4. *Select **Add > New User**.*

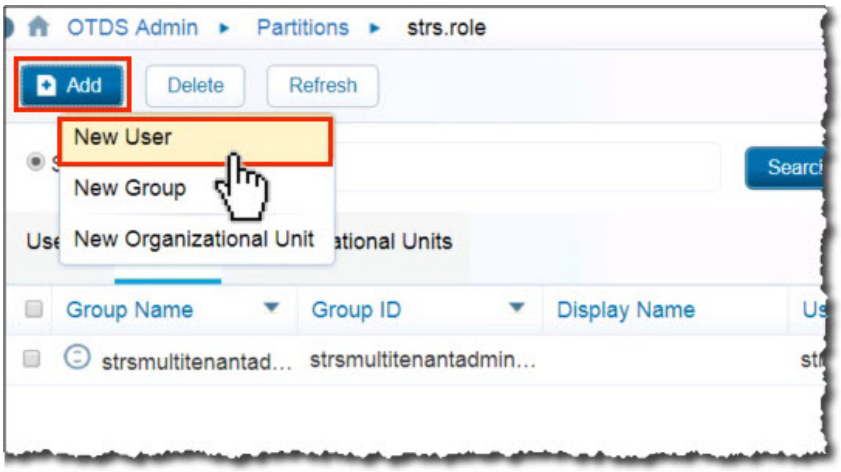


Figure 4-25:

New user

5. *Enter the following information for the new user and click **Next**:*
   - *User ID: **exadmin***
   - *First Name: **multitenantadmin***

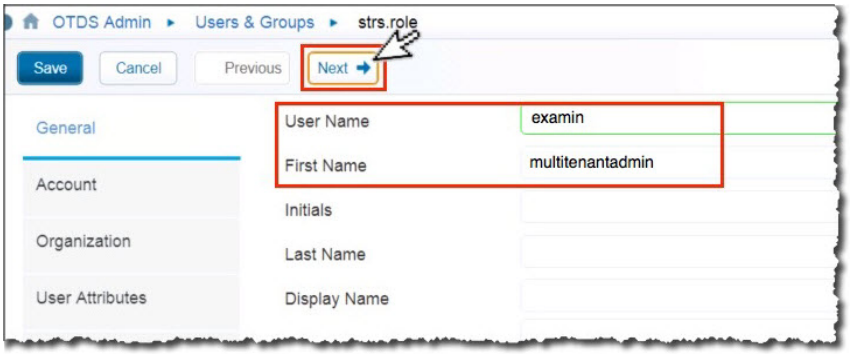


Figure 4-26:

Adding a user

6. *Enter the following information for the new user and click **Save**:*
   - ***Do not require password change on reset***
   - *User cannot change password: **selected***
   - *Password never expires: **selected***
   - *Password: **opentext***

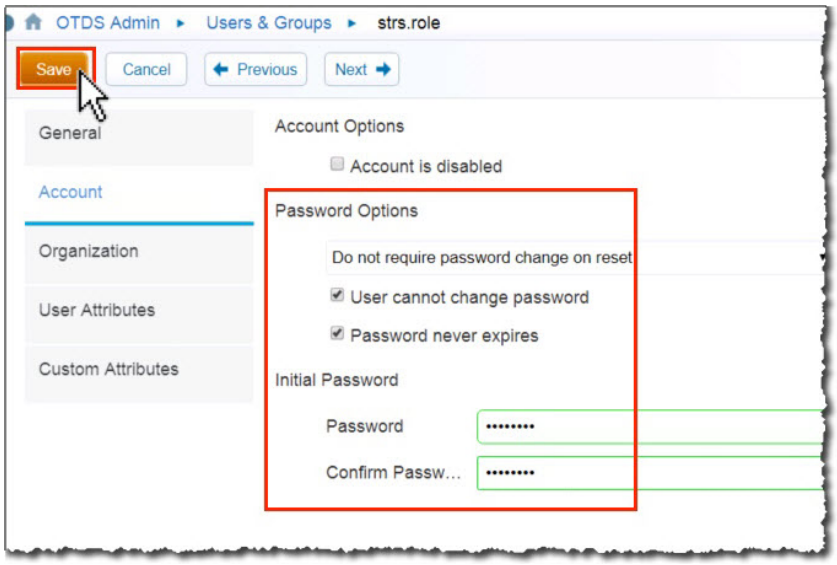


Figure 4-27:

Password

Setting the password this way is an optional step, only for training purposes.

7. *Select **Add > New User**.*
8. *Enter the following information for the new user and click **Next**:*
   - *User ID: **exbrowser***
9. *Enter the following information for the new user and click **Save**:*
   - ***Do not require password change on reset***
   - *User cannot change password: **selected***
   - *Password never expires: **selected***
   - *Password: **opentext***

10. *Select **Edit Membership** from the Actions drop-down for **exadmin**.*

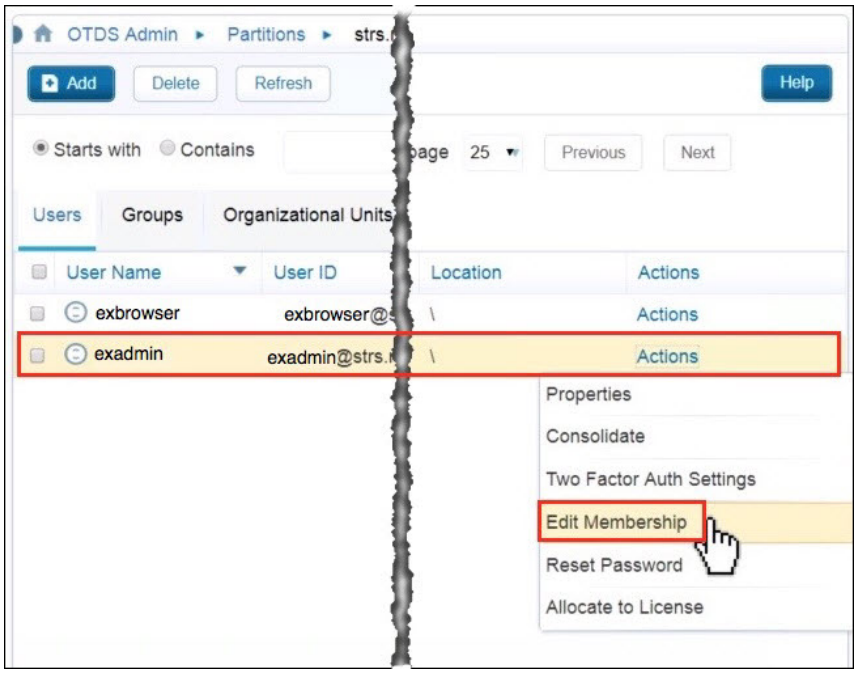


Figure 4-28:

Edit membership

11. *Click **Add To Group**.*

Figure 4-29:

Add to Group

12. *Select **strsmultitenantadmins** and click **Add Selected**.*

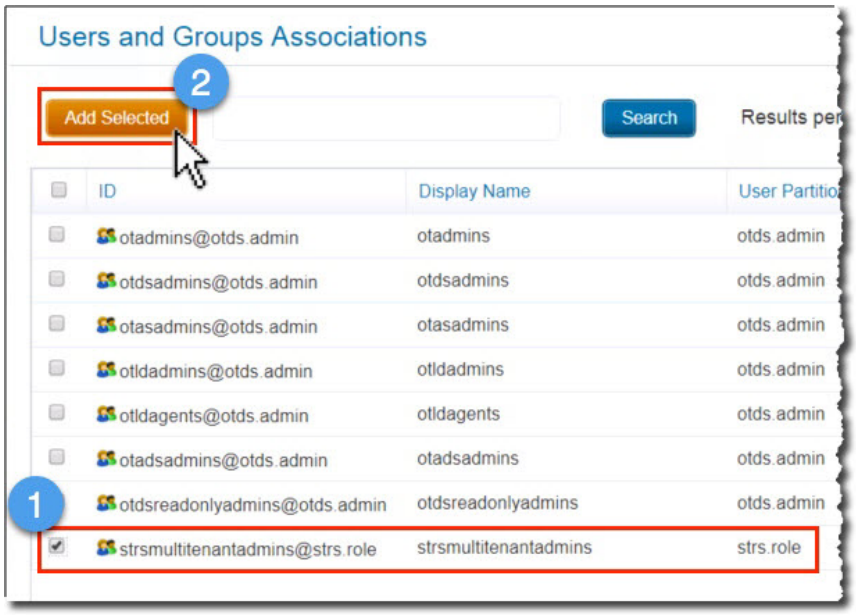


Figure 4-30:

Add to selected group

13. *Also select **otadmins@otds.admin** and click **Add Selected**.*
14. *Click **Close**.*

Configure access roles

1. *Click **Access Roles**.*

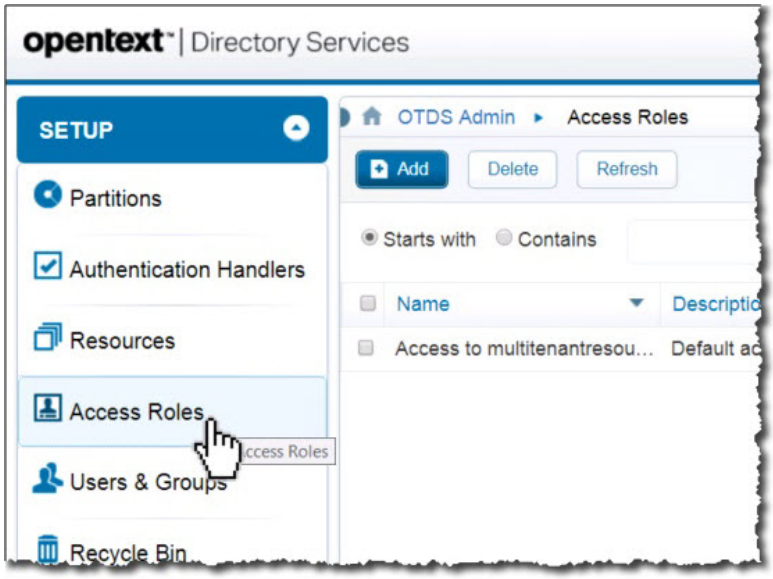


Figure 4-31:

Access Roles

2. *Select **View Access Roles Details** from the Actions drop-down.*

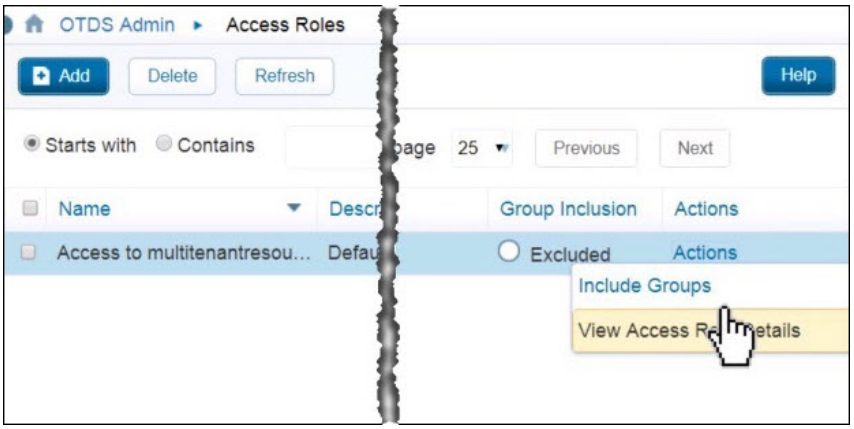


Figure 4-32:

View Access roles details

3. *Click **Add**.*

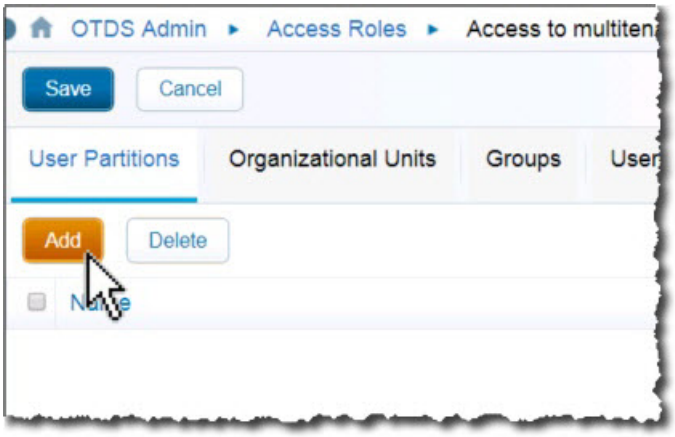


Figure 4-33:

Add

4. *Select **strs.role**, click **Add Selected items to Access Role** and then click **Close Dialog**.*

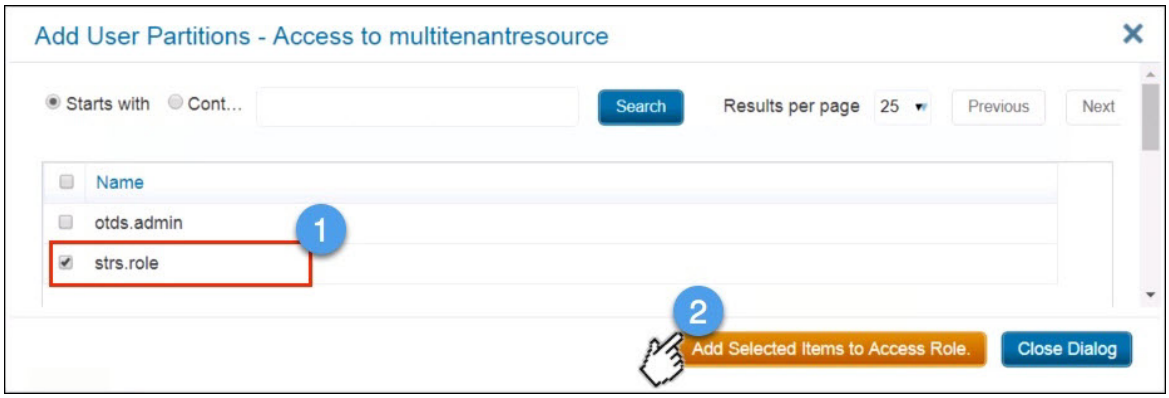


Figure 4-34: Added to Access role

5. *Click **Save**.*

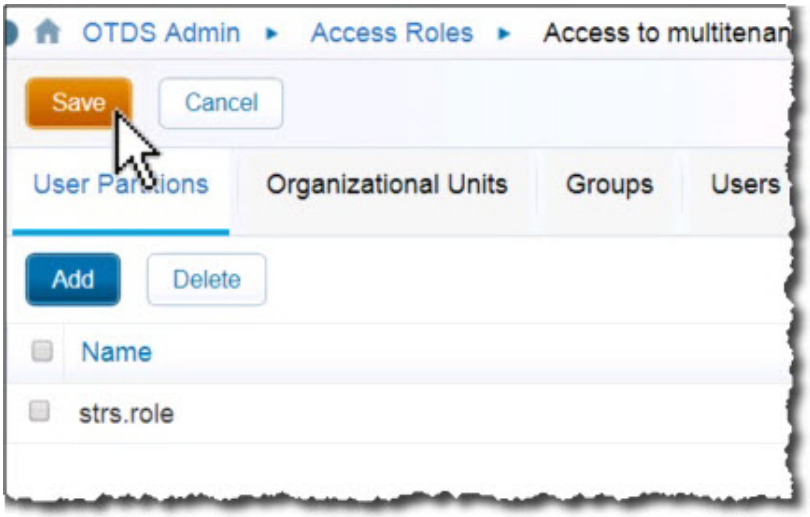


Figure 4-35: Save

Add a trusted site

1. *Select **Trusted Sites**.*

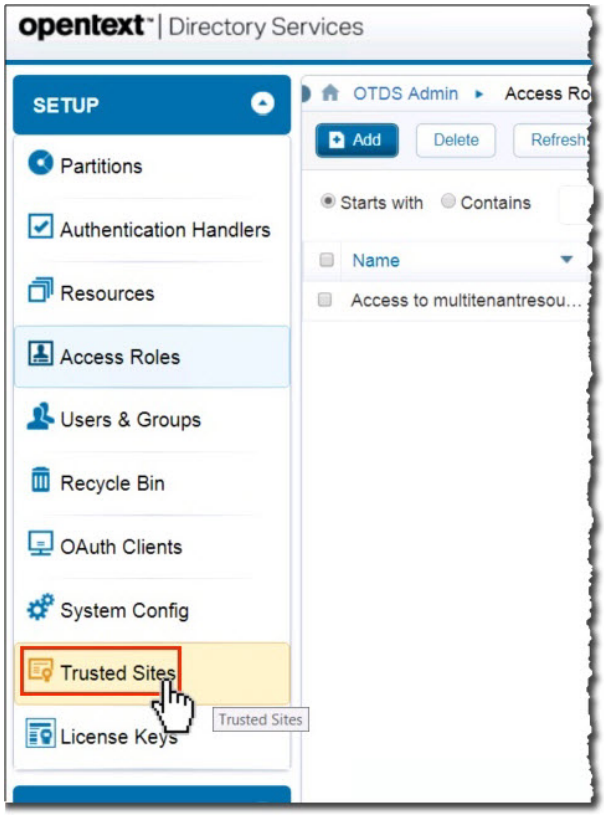


Figure 4-36:

Trusted sites

2. *Click **Add**.*

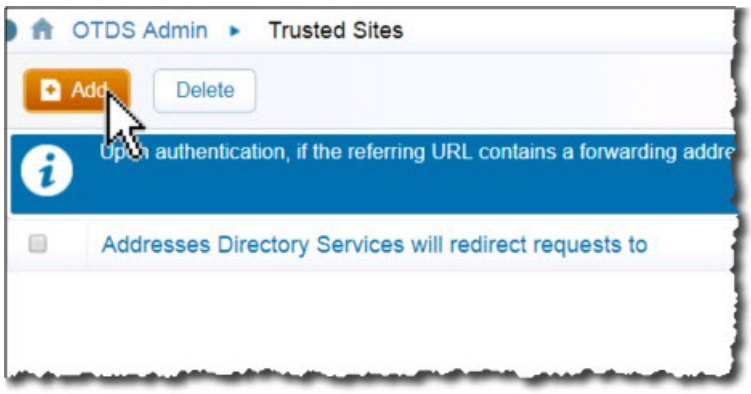


Figure 4-37:

Add

3. *Enter **https://thecompany.com**.*

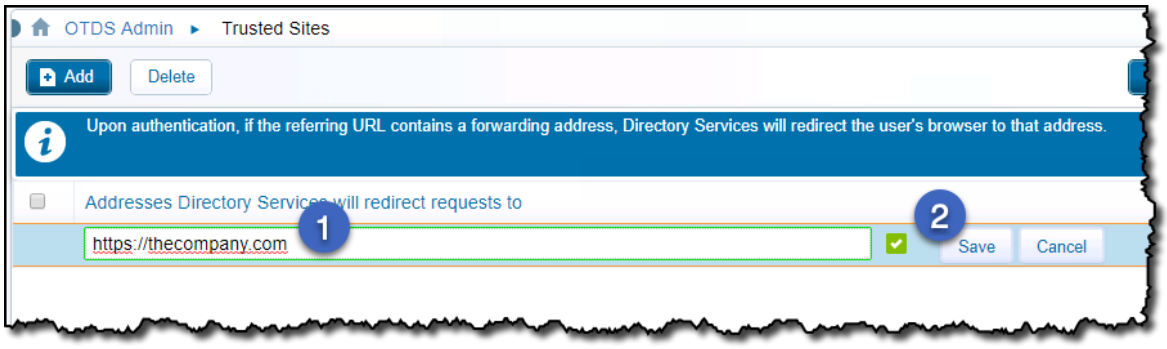


Figure 4-38: Trusted site added

4. *Click **Save.***

Lab: Create and configure a tenant

Create a tenant

1. *In a DOS window, navigate to **C:\OpenText\OTDS16.4.2\install** and enter the following command at the prompt:*

   otdstenant -addtenant tenant1 opentext

Make sure no error is reported.

2. *Once completed close the DOS window.*

Add the strs.role partition in OTDS for the new tenant

1. *In Chrome, navigate to **https://thecompany.com:8443/otdstenant/tenant1/otds-admin** (this URL is bookmarked under OTDS > OTDS tenant1).*
2. *Log in using the following credentials:*
   - *User name: **otadmin@otds.admin***
   - *Password: **opentext***
3. *Select **Partitions > Add > New Non-synchronized User Partition**.*
4. *Name the new partition **strs.role** and click **Save**.*

Create a password policy

1. *Navigate to **Partitions**.*
2. *Select **Password Policy** from the **Actions** drop-down for the strs.role user partition.*
3. *In the Password Policy - strs.role window, set the following values:*
   - *Use global policy: **Unchecked***
   - *Minimum number of characters: **4***
   - *Minimum number of digits: **0***
   - *Minimum number of symbols: **0***
   - *Minimum number of uppercase characters: **0***
   - *Minimum number of lowercase characters: **1***
   - *Minimum number of changes to previous password: **0***
   - *Number of unique passwords before an old password can be reused: **3***
4. *Click **OK**.*

Add the groups and users in OTDS for the new tenant

1. *Select **View Members** from the Actions drop-down for the **strs.role** partition.*
2. *Select **Add > New Group**.*
3. *Name the new group **strstenantadmins** and click **Save**.*
4. *Repeating the previous steps add the following groups:*
   - ***strstenantusers***
   - ***strsreviewers***
   - ***strsondemandusers***
   - ***strsbcausers***
   - ***strscommdesigners***
5. *Select **Add > New User**.*
6. *Enter the following information for the new user and click **Next**:*
   - *User ID: **exadmin***
7. *Enter the following information for the new user and click **Save**:*
   - ***Do not require password change on reset***
   - *Password never expires: **selected***
   - *Password: **opentext***
8. *Select **Add > New User**.*

9. *Repeating the previous steps, add the following users:*
   - *User ID:* ***exuser***
     - ***Do not require password change on reset***
     - *Password never expires:* ***selected***
     - *Password:* ***opentext***
   - *User ID:* ***ondemanduser***
     - ***Do not require password change on reset***
     - *Password never expires:* ***selected***
     - *Password:* ***opentext***
   - *User ID:* ***exreviewer***
     - ***Do not require password change on reset***
     - *Password never expires:* ***selected***
     - *Password:* ***opentext***
   - *User ID:* ***exbrowser***
     - ***Do not require password change on reset***
     - *Password never expires:* ***selected***
     - *Password:* ***opentext***
   - *User ID:* ***excauser***
     - ***Do not require password change on reset***
     - *Password never expires:* ***selected***
     - *Password:* ***opentext***
   - *User ID:* ***excomdeuser***
     - ***Do not require password change on reset***
     - *Password never expires:* ***selected***
     - *Password:* ***opentext***

Assign users to groups

1. *Making sure the* ***Users*** *tab is active, select* ***Edit Membership*** *from the* ***Actions*** *drop-down for the* ***exadmin*** *user.*
2. *Click* ***Add To Group****.*
3. *Select* ***strstenantadmins@strs.role****,* ***otadmins@otds.admin*** *and click* ***Add Selected*** *and then click* ***Close****.*

4. *Repeating the previous last steps associate the following users with the specified group:*
   - ***exuser** with **strstenantusers**, **strscommdesigners**, and **strsbcausers***
   - ***exreviewer** with **strsreviewers***
   - ***ondemanduser** with **strsondemandusers***
   - ***excauser** with **strsbcausers***
   - ***excomdeuser** with **strscommdesigners***

Add the tenant resource

1. *Select **Resources > Add**.*
2. *Name the new resource **ExstreamResource**, click **Next** twice and then click **Save**.*
3. *Copy the value of the ExstreamResource resource identifier (you will use this value later).*
   - *Resource Identifier: _____*
4. *In Chrome open a new tab (without closing any open tab) and navigate to **http://thecompany.com:8084/otdstenant/tenant1/otdsws/api/index.html** (this URL is bookmarked under OTDS > OpenText Directory Services REST API Tenant1).*
5. *Click **resources: Operations on resources > /resources/{resource_id}/activate**.*
6. *Click **Try it out**.*
7. *Paste the value of the ExstreamResource resource identified in the Resource ID text box and click **Execute**.*

8. *Copy the value of the ExstreamResource **secret_key** (you will use this value later, you can paste it to NotePadd++).*
   - *secret key: _____*

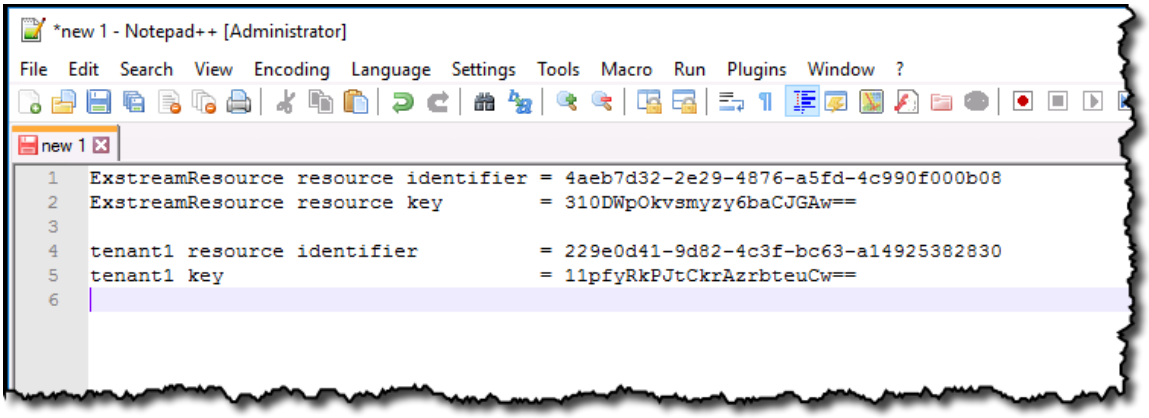


Figure 4-39: Values in NotePad++ for later use

9. *Return to Chrome tab where the tenant1 OTDS console is opened and click **Verify Activation**.*

Make sure you get a “The resource is activated message”.

10. *Click **OK**.*

Add Access roles to the tenant

1. *Click **Access Roles**.*
2. *Select **View Access Roles Details** from the **Actions** drop-down.*
3. *Click **Add**.*
4. *Select **strs.role**, click **Add Selected items to Access Role** and then click **Close Dialog**.*
5. *Click **Save**.*

Add a trusted site

1. *Select **Trusted Sites**.*
2. *Click **Add**.*
3. *Enter **https://thecompany.com** and click **Save**.*

5. Setting up the Exstream environment

Objectives

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

- Identify the use of the ss_tenantadmin utility
- Create the multi-tenant repository and the multi-tenant repository connection profile
- Create the connection profile to the multi-tenant OTDS
- Configure the management gateway to use the connection profiles
- Create the connection profile to the tenant repository
- Create the connection profile to the tenant OTDS
- Create a tenant repository
- Create a tenant that uses the connection profiles

About the ss_tenantadmin utility

The ss_tenantadmin command line utility is used to configure the management gateway, set up and manage Exstream tenants, and create the multi-tenant and tenant repositories.

To use the ss_tenantadmin utility:

1. In the command line window, browse to the following directory:
   <Exstream_Installation_directory>\Server\bin
2. Run the following command:
   ss_tenantadmin.exe <arguments>

Configuring the management gateway and creating the multi-tenant repository

To set up a new Exstream environment, you need to create the multi-tenant repository, and then connect the management gateway to the multi-tenant repository and to the multi-tenant OTDS.

To do this, you use the ss_tenantadmin utility to do the following:

1. Create a connection profile for the multi-tenant repository.
2. Create a connection profile for the multi-tenant OTDS.
3. Create the multi-tenant repository in the database (based on the connection profile created earlier).
4. If you do not want to create the multi-tenant repository directly from the ss_tenantadmin.exe utility, you can generate scripts used to create the repository and then run the scripts using an external tool.
5. Configure the management gateway to use the connection profiles. This step connects the management gateway to the multi-tenant OTDS and multi-tenant repository.

After these steps, you can start the management gateway.

Prerequisites

- There is a Communications Server installation on the computer where you will run the commands.
- The database is prepared for use with Exstream.
  You have the database administration user name and password.
- The multi-tenant OTDS is configured in OTDS.
- You have the following information for the multi-tenant OTDS:
  - URL and port
  - Resource ID and resource password
  - User name and password for a user with read access to the OTDS server

Step 1 – Create the multi-tenant repository connection profile

To create the multi-tenant repository connection profile use the ss_tenantadmin.exe utility which is located in <Exstream Install Directory>\Server\bin.

Syntax:

```
ss_tenantadmin
-action configure_multitenant_repository
-dbhost value
-dbport value
-dbvendor value
-dbname value
-dbusername value
-dbpassword value
-output <path>\multitenant_respository_profile.xml
```

Where:

-dbhost <database_host> (Optional) 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>\<instance name>. For example: gbg5000\instance1.

-dbport <db_port_number> (Optional) The port used for communication with the database server.

-dbvendor <vendor> (Optional) The database vendor for the repository. This can be sqlserver, oracle, or postgres.

-dbname <db_name> (Optional) Applicable for SQL Server and Postgres – The name of the database for the repository.

-dbservicename <SID> (Optional) Applicable for Oracle – The Oracle Service name for the repository.

-dbusername <user_name/ schema_owner> (Optional) The database administration user that the management gateway uses to connect to the repository. On Oracle, this is the schema owner for the repository. The user is automatically created when the repository is created. You cannot use the system administrator as user name (for example, sa).

-dbpassword <password> (Optional) A password to access the repository.

-output <path_file> (Optional) A path and file name where the XML file with the connection profile will be saved. If this is not specified, the file is created in the directory where the command is run with the default name.

-env <path_file> (Optional) The path and file name of the environment file.

Step 2 – Create the connection profile to the multi-tenant OTDS

To create the connection profile to the multi-tenant OTDS use the ss_tenantadmin.exe utility which is located in <Exstream Install Directory>\Server\bin.

Syntax:

```
ss_tenantadmin
-action configure_multitenant_otds
-otdsusername value
-otdspassword value
-otdsresource value
-otdsresourcepassword value
-output <path>\multitenant_otds_profile.xml
```

Where:

-otdsURL <host> (Optional) The host name or IP address of the computer with the OTDS server.

-otdsport <port> (Optional) The port used for communication with OTDS.

-otdsusername <user_name> The OTDS user name. This user requires read access to OTDS. Communications Server applications use this user to browse OTDS.

-otdspassword <password> The OTDS password.

-otdsresource <resource_ID> The OTDS resource ID.

-otdsresourcepassword <password> The password for the OTDS resource.

-output <path_file> (Optional) A path and file name where the XML file with the connection profile will be saved. If this is not specified, the file is created in the directory where the command is run with the default name.

-env <path_file> (Optional) The path and file name of the environment file.

Step 3 – Create the multi-tenant repository

To create the multi-tenant repository use the ss_tenantadmin.exe utility which is located in <Exstream Install Directory>\Server\bin.

Syntax:

```
ss_tenantadmin.exe
-action create_multitenant_repository
-multitenantdbprofile
<path>\multitenant_respository_profile.xml
-MGWRoot value
-dbadminusername value
-dbadminpassword value
```

Where:

-multitenantdbprofile <path_file> The path to and name of the XML file containing the connection profile for the multi-tenant repository.

-dbadminusername <user_name> The database administration user that is used to create the repository. For example, the system administrator sa.

-dbadminpassword <password> The password for the database administration user.

-MGWRoot <path> (Optional) Specifies the location of the management gateway root directory.

-env <path_file> (Optional) The path and file name of the environment file.

Step 4 – Configure the management gateway to use the connection profiles

To configure the management gateway to use the connection profiles use the ss_tenantadmin.exe utility which is located in <Exstream Install Directory>\Server\bin.

Syntax:

```
ss_tenantadmin
-action configure_mgw
-MGWRoot value
-multitenantdbprofile
<path>\multitenant_respository_profile.xml
-multitenantotdsprofile
      <path>\multitenant_otds_profile.xml"
```

Where:

-multitenantdbprofile <path_file> The path to and name of the XML file containing the connection profile for the multi-tenant repository.

-multitenantotdsprofile <path_file> The path to and name of the file containing the connection profile for the multi- tenant OTDS.

-MGWRoot <path> (Optional) Specifies the location of the management gateway root directory.

-env <path_file> (Optional) The path and file name of the environment file.

Step 5 – Start the management gateway service or process

You can start the management gateway from Windows Control Panel.

Adding a tenant

Adding a tenant to the Communications Center environment involves connecting the tenant to the appropriate tenant OTDS, and either creating a new tenant repository or connecting to the tenant to an existing repository (shared tenant repository scenario).

To do this, you use the ss_tenantadmin utility to do the following:

- Create a connection profile for the tenant OTDS.
- Create a connection profile for the tenant repository.
- If required, create the tenant repository in the database.
- Add the tenant to Communications Center environment. This step connects the tenant to the tenant OTDS and tenant repository.

After a tenant is added, you can (optionally) use the ss_tenantadmin utility to obtain the tenant ID.

Prerequisites

- There is a Framework and Exstream Communications Server installation on the computer where you will run the commands.
- You have the multi-tenant administrator user name and password.
- The database is prepared for use with Communications Center.
- You have the database administration user name and password.
- The tenant OTDS is configured in OTDS.

  You have the following information for the tenant OTDS:

  – URL and port
  – Resource ID and resource password
  – User name and password for a user with read access to the OTDS server

Step 1 – Create the connection profile to the tenant repository

To create the connection profile to the tenant repository use the ss_tenantadmin.exe utility which is located in <Exstream Install Directory>\Server\bin.

Syntax:

```
ss_tenantadmin
-action configure_tenant_repository
-dbhost value
-dbport value
-dbvendor value
- dbname value
-dbusername value
-dbpassword value
-output <path>\tenant_respository_profile.xml
```

Where:

-dbhost <database_host> 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>\<instance name>. For example: gbg5000\instance1.

-dbport <database_port_number> The port used for communication with the database server.

-dbvendor <vendor> The database vendor for the repository. This can be sqlserver, oracle, or postgres.

-dbname <database_name> Applicable for SQL Server and Postgres – The name of the database for the repository.

-dbservicename <SID> Applicable for Oracle – The Oracle Service name for the repository.

-dbusername <user_name/ schema_owner> The database administration user that the management gateway uses to connect to the repository. On Oracle, this is the schema owner for the repository. The user is automatically created when the repository is created. You cannot use the system administrator as user name (for example, sa).

-dbpassword <password> A password to access the repository.

-output <path_file> A path and file name where the XML file with the connection profile will be saved. If this is not specified, the file is created in the directory where the command is run with the default name.

-env <path_file> The path and file name of the environment file.

Step 2 – Create the connection profile to the tenant OTDS

To create the connection profile to the tenant OTDS use the ss_tenantadmin.exe utility which is located in <Exstream Install Directory>\Server\bin.

Syntax:

```
ss_tenantadmin
-action configure_tenant_otds
-otdsURL value
-otdsusername value
-otdspassword value
-otdsresource value
-otdsresourcepassword value
-output <path>\tenant_otds_profile.xml
```

Where:

-otdsURL <host> (Optional) The host name or IP address of the computer with the OTDS server.

-otdsport <port> (Optional) The port used for communication with OTDS.

-otdsusername <user_name> The OTDS user name. This user requires read access to OTDS. Communications Server applications use this user to browse OTDS.

-otdspassword <password> The OTDS password.

-otdsresource <resource_ID> The OTDS resource ID.

-otdsresourcepassword <password> The password for the OTDS resource.

-output <path_file> (Optional) A path and file name where the XML file with the connection profile will be saved. If this is not specified, the file is created in the directory where the command is run with the default name.

-env <path_file> (Optional) The path and file name of the environment file.

Step 3 (optional) – Create a tenant repository

To create a tenant repository use the ss_tenantadmin.exe utility which is located in <Exstream Install Directory>\Server\bin.

Syntax:

```
ss_tenantadmin
-action create_tenant_repository
-mtauser value
-mtapassword value
-tenantdbprofile
<path>\tenant_respository_profile.xml
-dbadminusername value
-dbadminpassword value
```

Where:

-mtauser <user_name> The multi-tenant administrator user name. This user must be a member of the multi-tenant administrator group.

-mtapassword <password> The password for the multi-tenant administrator user.

-multitenantdbprofile <path_file> The path to and name of the XML file containing the connection profile for the multi-tenant repository.

-dbadminusername <user_name> The database administration user that is used to create the repository. For example, the system administrator sa.

-dbadminpassword <password> The password for the database administration user.

-MGWRoot <path> (Optional) Specifies the location of the management gateway root directory.

-env <path_file> (Optional) The path and file name of the environment file.

Step 4 – Create a tenant that uses the connection profiles

To create a tenant that uses the connection profiles use the ss_tenantadmin.exe utility which is located in <Exstream Install Directory>\Server\bin.

Syntax:

```
ss_tenantadmin
-action create_tenant
-mtauser value
-mtapassword value
-tenantdbprofile
<path>\tenant_respository_profile.xml
-tenantotdsprofile <path>\tenant_otds_profile.xml
-tenantname value
```

Where:

-mtauser <user_name> The multi-tenant administrator user name. This user must be a member of the multi-tenant administrator group.

-mtapassword <password> The password for the multi-tenant administrator user.

-tenantdbprofile <path_file> The path to and name of the XML file containing the connection profile for the tenant repository.

-tenantotdsprofile <path_file> The path to and name of the file containing the connection profile for the tenant OTDS.

-tenantname <tenant_name> A name for the tenant. This name is needed by tenant administrators and tenant users, etc., to log on to the Communications Center tools, such as Control Center and Describer.

-tenantdesc <description> (Optional) A description of the tenant.

-env <path_file> (Optional) The path and file name of the environment file.

Lab: Configuring the management gateway and creating the multi-tenant repository

The values provided for the ss_tenantadmin parameters in the activities of the following activities should match the values used during the installation.

Make sure you get no errors before proceeding with each next activity.

Step 1 - Create the connection profile to the multi-tenant repository

1. *Launch a **DOS** window.*
2. *Enter the following command **md C:\ConnectionProfiles**.*

This folder will contain the generated xml files used during the configurations. The name can be any name.

3. *In the DOS window enter the following command **CD C:\OpenText\Exstream\16.6\Server\bin** (to navigate to the directory to where the ss_tenantadmin.exe utility is located).*
4. *Enter the following command (all in one line):*

```
ss_tenantadmin
-action configure_multitenant_repository
-dbhost thecompany.com
-dbport 1433
-dbvendor sqlserver
-dbname DB_MultiTenant
-dbusername MultiTenUsr
-dbpassword opentext
-output
"C:\ConnectionProfiles\multitenant_repository_prof
ile.xml"
```

Make sure that you get no errors before proceeding with next activity.

Step 2 - Create the connection profile to the multi-tenant OTDS

1. In the DOS window enter the following command (all in one line):

> For the -otdsresource and -otdsresourcepassword parameter values make sure to use the “Resource Identifier” and “secret key” that you got in the “Create the Exstream resource” on page 4 - 16 activity.

```
ss_tenantadmin
-action configure_multitenant_otds
-otdsURL thecompany.com
-otdsport 8443
-otdsusername exbrowser
-otdspassword opentext
-otdsresource REPLACE_WITH_Resource_Identifier
-otdsresourcepassword “REPLACE_WITH_secret_key”
-output
C:\ConnectionProfiles\multitenant_otds_profile.xml
```

Make sure that you get no errors before proceeding with next activity.

Step 3 - Create the multi-tenant repository

1. In the DOS window enter the following command (all in one line):

```
ss_tenantadmin
-action create_multitenant_repository
-multitenantdbprofile
"C:\ConnectionProfiles\multitenant_repository_prof
   ile.xml"
-dbadminusername sa
-dbadminpassword opentext
```

Make sure that you get no errors before proceeding with next activity.

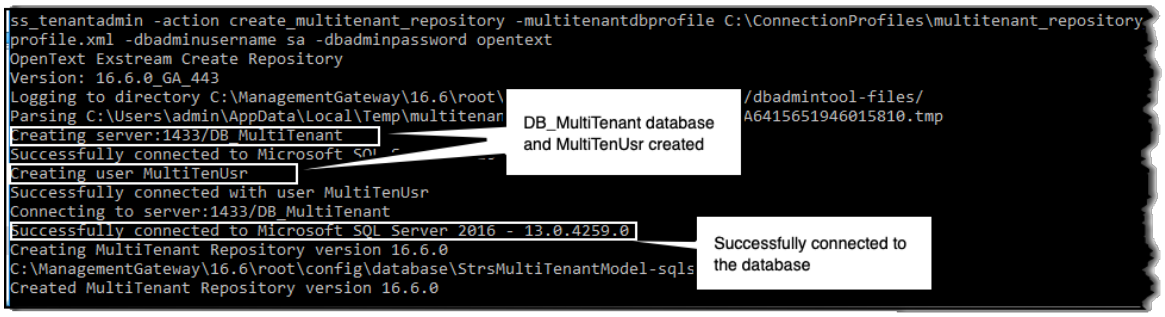


Figure 5-1: Create the multitenant repository

Step 4 - Configure the management gateway to use the connection profiles

1. *In the DOS window enter the following command (all in one line):*

```
ss_tenantadmin
-action configure_mgw
-multitenantdbprofile
"C:\ConnectionProfiles\multitenant_repository_prof
   ile.xml"
-multitenantotdsprofile "C:\ConnectionProfiles\
   multitenant_otds_profile.xml"
```

The values provided here should match the values used during the installation. Make sure you get no errors before proceeding with the next step.

Figure 5-2: Configure the Management Gateway

Step 5 - Restart the Management Gateway

1. *Open Windows **Services** (the shortcut to Services is available in the Taskbar).*
2. *Start the **StreamServe Management Gateway 16.6.0** Windows service.*

Make sure the Status is set to Running.

Lab: Adding a tenant

In this second exercise you will add a tenant (tenant1) to the Exstream environment which involves connecting the tenant to the appropriate tenant OTDS, and creating the new tenant repository.

Step 1 - Create the connection profile to the tenant repository

1. *In the DOS window enter the following command (all in one line):*

```
ss_tenantadmin
-action configure_tenant_repository
-dbhost server
-dbport 1433
-dbvendor sqlserver
-dbname DB_DEVTenant
-dbusername DEVTenantUsr
-dbpassword opentext
-output
C:\ConnectionProfiles\tenant_repository_profile.
xml
```

Step 2 - Create the connection profile to the tenant OTDS

1. *In the DOS window enter the following command (all in one line):*

For the -otdsresource and -otdsresourcepassword parameter values make sure to use the “Resource Identifier” and “secret key” that you got in the “Add the tenant resource” on page 4 - 32 activity.

```
ss_tenantadmin
-action configure_tenant_otds
-otdsURL thecompany.com/otdstenant/tenant1
-otdsport 8443
-otdsusername exbrowser
-otdspassword opentext
-otdsresource REPLACE_WITH_Resource_Identifier
-otdsresourcepassword "REPLACE_WITH_secret_key"
-output
"C:\ConnectionProfiles\tenant_otds_profile.xml"
```

Step 3 - Create a tenant repository

1. *In the DOS window enter the following command (all in one line):*

```
ss_tenantadmin
-action create_tenant_repository
-mtauser exadmin
-mtapassword opentext
-tenantdbprofile
"C:\ConnectionProfiles\tenant_repository_profile.x
   ml"
-dbadminusername sa
-dbadminpassword opentext
```

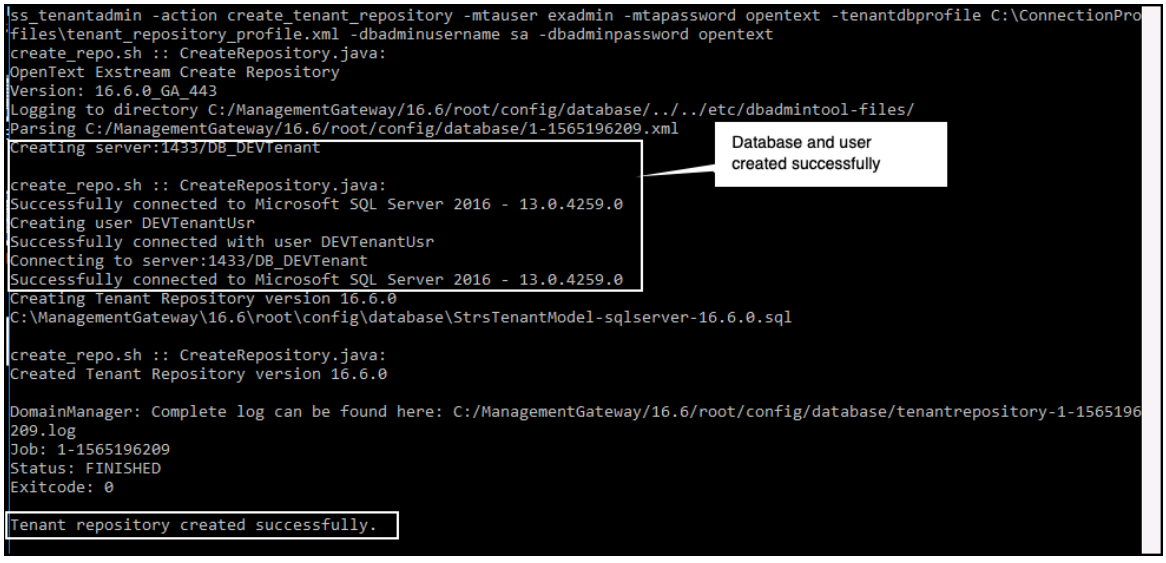


Figure 5-3: Tenant repository created

Step 4 - Create a tenant that uses the connection profiles

1. *In the DOS window enter the following command (all in one line):*

```
ss_tenantadmin
-action create_tenant
-mtauser exadmin
-mtapassword  opentext
-tenantdbprofile
"C:\ConnectionProfiles\tenant_repository_profile.x
   ml"
-tenantotdsprofile
   "C:\ConnectionProfiles\tenant_otds_profile.xml"
-tenantname tenant1
```

Figure 5-4: Tenant created

6. Application domains

Objectives

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

- Define application domains
- Plan the application domains
- Create application domains
- Describe the relationship between repositories and domains
- Describe the relationship among service gateways, the web applications, and the application domains
- Create an application domain

About domains

A domain is a logical grouping of Communications Server applications. A common scenario for using multiple domains is to separate the applications used for development, testing, and production. Domains can also be used to separate applications that are used for different business processes or geographical regions. The number of domains used in your environment and the grouping of applications depends on the individual requirements of the tenant, company, or organization.

To help you decide how set up the domains, this section describes how domains work with the repositories, the service gateway, and the Communications Center web applications.

Creating application domains

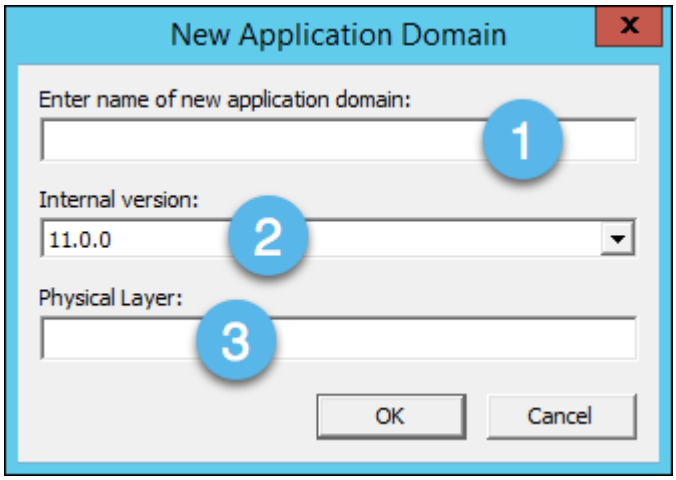


Figure 6-1:

Application domains

Option	Description
Name	Name of the application domain with which the application domain will be listed in Control Center.
Internal version	Note that the internal version number of Exstream v16.6 is 16.6. (In some versions the release and internal versions are not the same.)
Physical layer	(Optional) Name of the physical layer used by the application domain.

Repositories and domains

All the applications in one domain share the same set of statistics, queue, message, tracking, logging, document broker, and temporary data repositories. Each application domain can be connected to one repository of each type. The repositories required for each domain will depend on what kind of solutions the Communications Server applications in the domain will run.

Shared or separate repositories

In environments with multiple domains, there are different options for setting up the repositories: using separate repositories, using shared repositories, or using combinations of shared and separate repositories. Because all information in the repositories is assigned a domain ID, the data can be filtered by domain even in environments with shared repositories.

Separate repositories With this option, each domain has its own set of repositories. This approach is suitable for environments where a higher level of data isolation is required or where you expect a high load on the repositories.

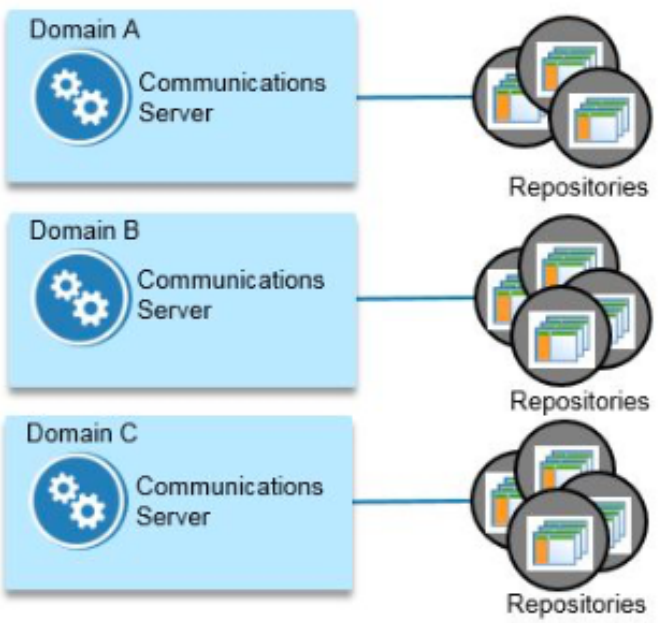


Figure 6-2:

Separate repositories

Shared repositories With this option, several domains share the same set of repositories.

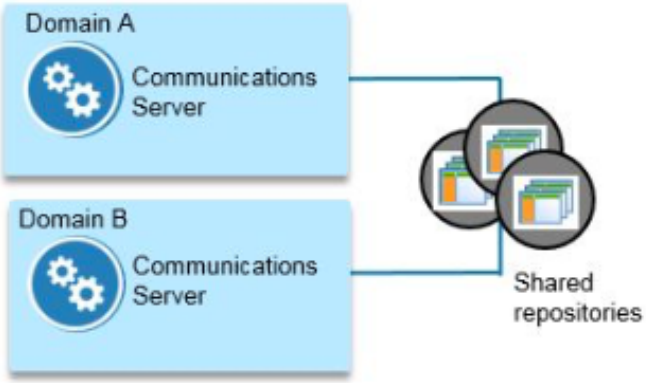


Figure 6-3:

Shared repositories

Combinations of shared and separate repositories

Each domain can also have some of its own repositories and share some repositories. For example, two domains may share a central tracking repository and a central statistics repository, but use separate queue repositories.

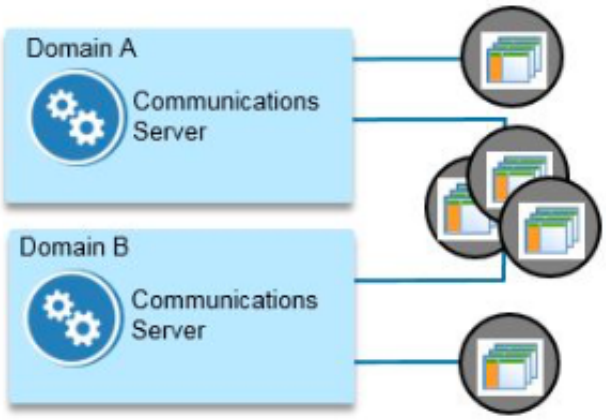


Figure 6-4:

Combination

Logical databases

Each repository is a logical database. This means you can create all the repositories in a single Oracle schema or SQL Server database. Or you can create each repository in a separate schema or database.

Hosts and domains

Domains are independent from hosts. This means you can use different hosts to run applications in the same domain, or you can use the same host to run applications in different domains.

Service gateways, the web applications, and domains

The Exstream web applications use the service gateway application to access the repositories, the common asset service, and the metadata model.

Each domain requires its own service gateway with unique port numbers. For load balancing and failover, it is possible to set up several service gateways in a domain. When being accessed, a web client sends a request to the management gateway, asking for a service gateway endpoint. The request includes information about the tenant and the domain to be used.

In environments with multiple application domains, the service gateway used controls which domain's data, design resources, and services are accessed. The following images provide some examples of which data users can access from each domain.

WorkShop In WorkShop, users only have access to the templates, themes, images, texts and rules in the domain they sign in to.

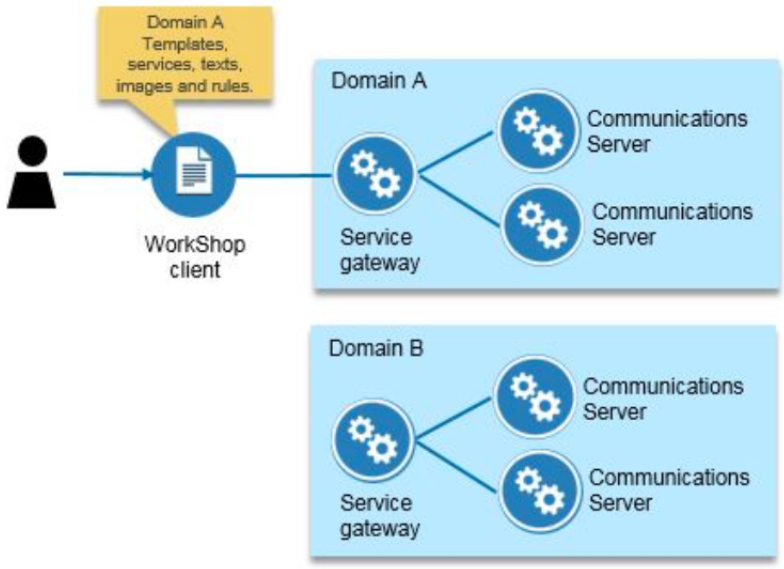


Figure 6-5:

WorkShop and domains

Supervisor In the Review view in Supervisor, users only have access to the paused messages in the domain they sign in to.

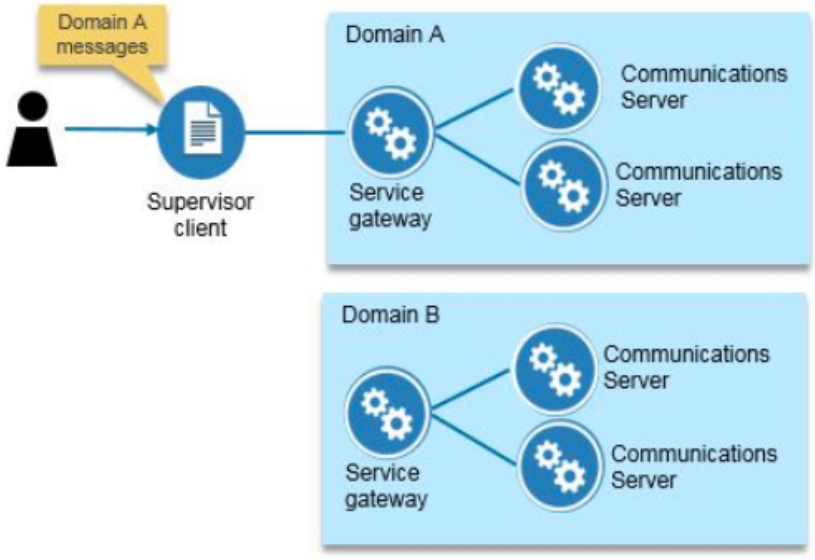


Figure 6-6:

Review view in Supervisor and domains

StoryBoard and Retouch In StoryBoard and ReTouch, users only have access to the image, text and rule resources in the domain they sign in to.

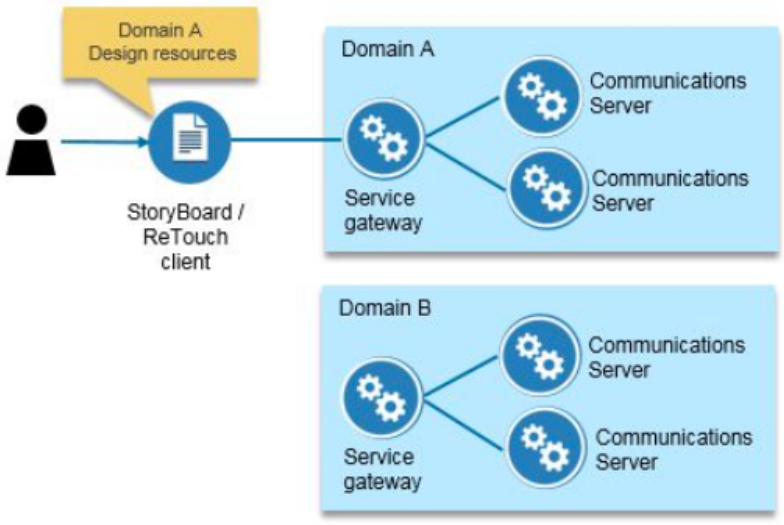


Figure 6-7:

StoryBoard, ReTouch, and domains

Lab: Creating a domain

Control Center is an administration tool used to deploy, run, and administer Communications Server applications. From Control Center you can manage Communications Server applications, which include Communications Server applications, service gateways, and Task Scheduler applications. You can also use Control Center to create the Communications Center repositories and application domains.

Before you can log on to Control Center, you need to set up a connection for your tenant to a management gateway.

You only need to set up one tenant connection to manage Communications Server applications on the other management gateways in your environment. If you work with several tenants, you need to set up one connection for each tenant. You can use the same management gateway to connect to each tenant, but with different tenant names.

In cases where the management gateway you are connected to is no longer available, you can remove the tenant node in Control Center. Removing the tenant node does not remove the tenant or management gateway from the overall Communications Center environment.

Set up a connection for a tenant

1. *Navigate to **Programs > OpenText Exstream 16.6** and launch **Control Center**.*

Control Center opens.

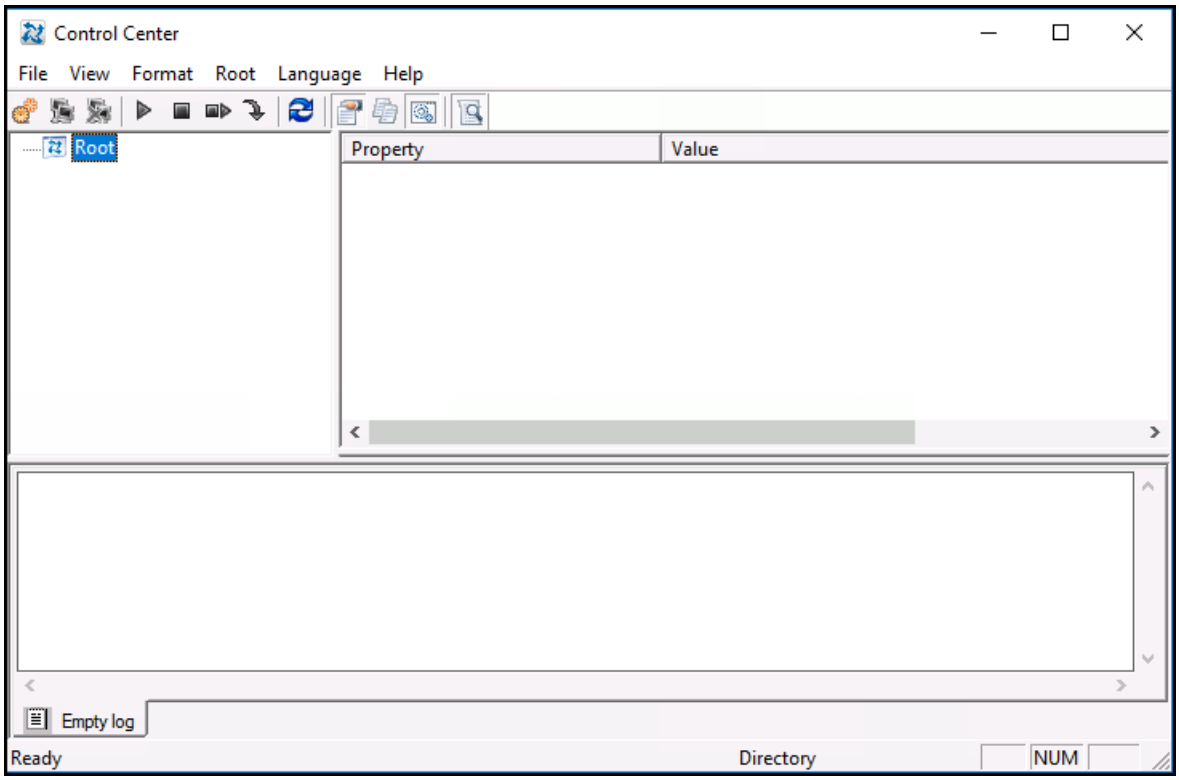


Figure 6-8: Control Center

2. *Right-click the **Root** node and select **New Tenant Connection** from the pop-up menu.*

3. *In the Select Management Gateway window click the **Add** button (green plus icon).*

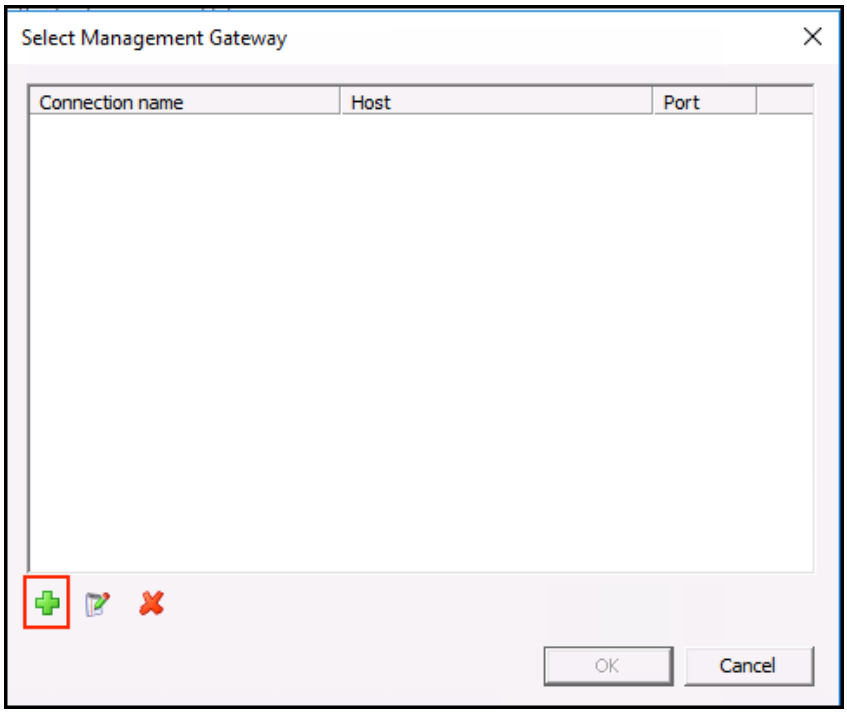


Figure 6-9:

Add icon

4. *In the Add New Connection window enter the following information (leave the other field default values unchanged) and click the **OK** button:*
   - *Connection Name: **tenant1Connection***
   - *Host: **https://thecompany.com***

You are redirected back to the Select Management Gateway window, which now displays the new tenant1Connection connection.

5. *In the Select Management Gateway window, select **tenant1Connection** and click the **OK** button.*
6. *In the Login to Management Gateway window enter the following information and click he **OK** button:*
   - *Tenant name: **tenant1***
   - *User name: **exadmin***
   - *Password: **opentext***
   - *Remember password: **selected***

You are redirected back to Control Center, which now displays the new tenant1.

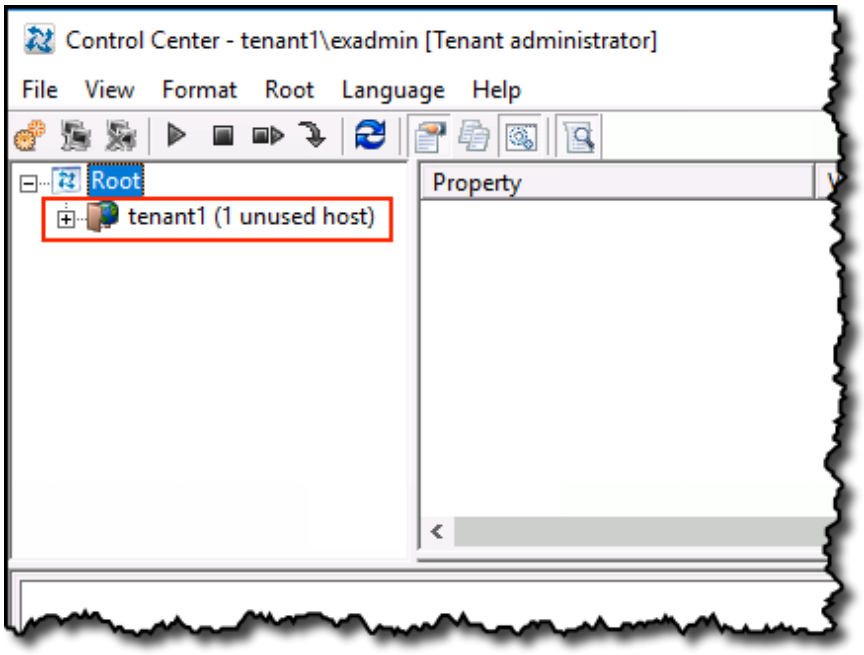


Figure 6-10:

tenant1

Create a Domain

1. *Expand the **tenant1 (1 unused host)** node.*
2. *Right-click **Application domains** and select **New Application Domain**.*

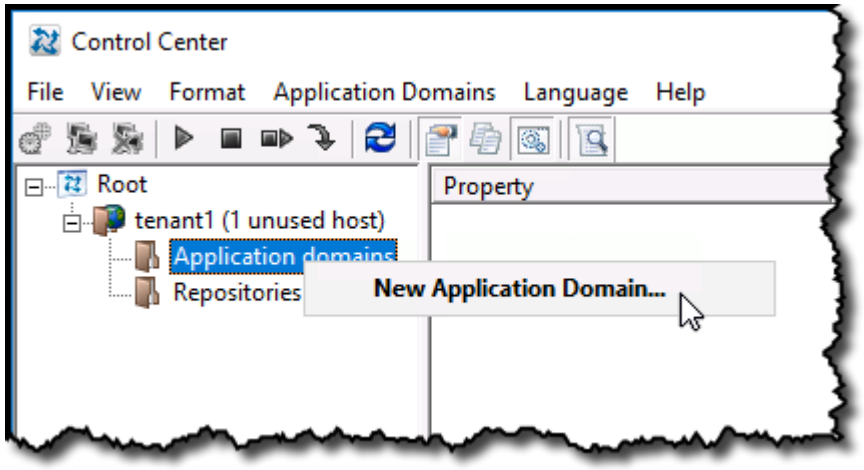


Figure 6-11:

Add new domain

3. *In the New Application Domain enter the following information (leave the other field default values unchanged) and click the **OK** button:*
   - *Enter name of new application domain: **domain1***

You are redirected back to Control Center, which now displays the new domain1.

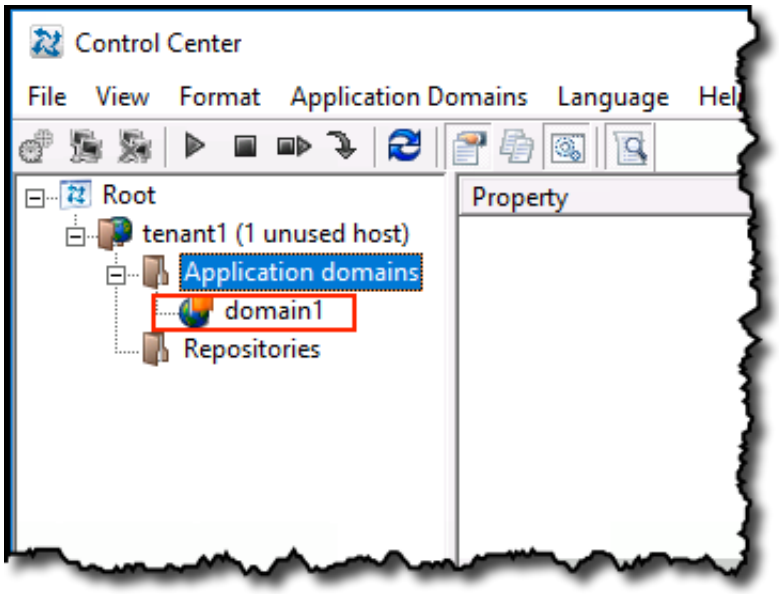


Figure 6-12:

domain1

7. Repositories

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

- Define repository
- Identify the different Exstream repository types
- Describe the relationship between repositories and application domains
- Identify the domain-required repositories
- Describe indexing and partitioning repositories
- Estimate the size of the database and transaction logs
- Create a repository

Repositories overview

Repositories are used by Exstream to store different types of information.

You can either create the Exstream repositories directly in Control Center, or you can generate the database scripts in Control Center and then run the scripts using an external tool. For example, 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.

If Control Center is not available, you can carry out the corresponding procedures using the command line utilities.

Repository types

Exstream uses the following types of repositories:

Collector Archive Stores information for archiving purposes.

Document Broker Stores Document Broker Documents.

Logging Stores log messages.

Messagestore Stores documents created in ReTouch, Messages that are paused by exceptions rules, and Message or Document properties.

Queue Stores input jobs, output jobs, and job information in queues as specified in the Communications Builder Project configuration.

Statistics Stores processing statistics about the usage of Communications Server applications.

Tracking Stores Tracker IDs and status information for top jobs.

Temporary Data Images used in StoryBoard and ReTouch previews.

EasyLink reports The EasyLink connectors let you send output via the OpenText EasyLink cloud- based fax and notification services. By using these connectors, you can send output via Fax, Text message (SMS) and Email.

Solr Stores information used by Solr.

Content Server Stores information used by Content Server.

Repositories and domains

Repositories are linked to domains.

Domain-required repositories This section describes the scenarios when each repository is required and what information is stored in the repository.

Statistics Required by every domain that runs Communications Server applications.

Queue Required if queues are used in any of the Communications Builder Projects that are run in the domain.

Tracking Required if your company or organization wants to use the Track view in Supervisor or implement another method of tracking top jobs for the Communications Server applications in the domain.

Messagestore Required if any of the applications in the domain will run a Communications Builder Project with a service-enabled message. Examples include:

- Communications Server applications used with Supervisor (the Review view), ReTouch, or StoryBoard
- Communications Server applications with a Template Engine Process
- Communications Server applications that store properties for the Message

Logging Required only if you want to use database logging for any of the applications in the domain.

Document Broker Required to run Communications Server applications that are part of a Document Broker Plus solution.

The documents can be administered in the Produce view in Supervisor.

Temporary Data It is optional and is used if you want to temporarily cache images used in previews.

Collector Archive It is optional and is used if you want to store documents in a Collector Archive repository, and be able view and delete these documents in the Supervisor web application.

EasyLink reports It is optional and is used if you want to use OpenText EasyLink cloud-based fax and notification services.

Adjusting repositories

Before using a Communications Center repository in a testing or production environment, you should review the information in this section and make any necessary adjustments to the repository. How to carry out the adjustments in third party products is not described. For such information, see user documentation from the database vendor.

Which adjustments are necessary depend on the amount of data in the repository. The queue repository, Message repository, and Document Broker repository are examples of repositories that usually include large amounts of data.

If the company has any performance requirements on the repository, this also affect the adjustments. For example, if there are specific performance requirements on the tenant repository.

Indexing

For most tables, performance is improved if you index one or several of the database columns.

If the Communications Center repository contains dynamically created database columns with user defined properties configured in Describer, these columns are usually suitable for indexing. The tenant repository, Message repository, and Document Broker Plus repository are examples of repositories that contain dynamically created database columns.

You should also adjust your indexes to the queries being used. For example, if a column is frequently accessed from a Communications Center web application, you should consider indexing this column.

You index columns using the appropriate tool from the database vendor. For example, SQL Server Management Studio or Oracle SQL Developer.

Partitioning

If the Communications Center repository contains large tables, performance might be improved if you partition tables and indexes. The tables and indexes are then split into smaller components, where each component can be managed and manipulated individually.

You partition tables and indexes using the appropriate tool from the database vendor. For information about which database editions support partitioning, see the user documentation from the database vendor.

Estimating sizes of database and transaction log

To prevent the disks from running out of space, you should specify maximum sizes for the database file and the transaction log for the Communications Center repository.

Database file

You can estimate the size of the database file based on the sizes of the included database tables. If the database vendor provides a tool with a reporting function, you can use this tool to find out the actual size of each table.

Transaction log

The transaction log size should be 20-25% of the database size. However, the smaller the size of the database, the greater the size of the transaction log, and vice versa. For example, if the estimated database size is 10 MB, you set the size of the transaction log to 4-5 MB. If the estimated database size is over 500 MB, you set the size of the transaction log to 50 MB.

OpenText recommends that you store the transaction log on a separate physical disk.

About job tracking

This section describes how top job tracking works when a tracking repository is used.

Input and output jobs

When a Communications Center application receives input, for example from an ERP system, it creates an input job. When the input job is processed, it can generate several output items. Output items may include:

- Documents delivered by the output connectors
- Messages that are paused and stored in the Message repository
- Documents stored in a Document Broker repository
- Resources stored in the common asset service by the Resource filter or Resource output connector

Top jobs

When a tracking repository is used, a top job is also created when a Communications Server application receives input. The top job is assigned a tracker ID, and is connected to the input job and each of the output items.

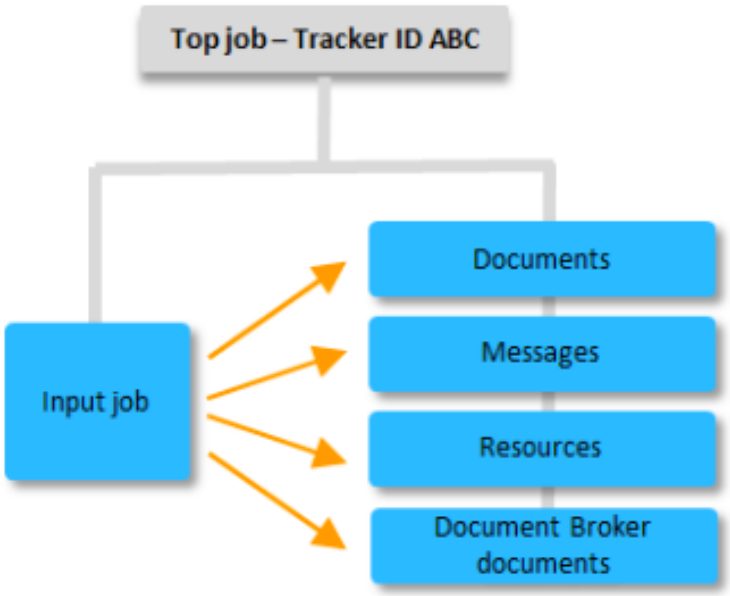


Figure 7-1:

Top jobs

Each top job has a counter that tracks the status of the job. Each time an output item is added to the top job the counter increases by 1. When each output item is completed, either successfully or with errors, the counter is decreased by 1. When the counter returns to 0, the top job is marked as either:

- Successfully completed: if all the output items were successfully completed.
- Completed with errors: if one or more of the output items were completed with errors.

When a top job that is connected to Document Broker documents or Messages in the message repository is marked as completed (either successfully or with errors), a new top job is generated. The new top job can be used if Document broker needs the document again or the message is released from the message repository again. During repository maintenance the initial top job can be deleted from the repository.

Creating the repositories

Creating the database directly from Control Center

You can create the database for the Communications Server repositories 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.

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

New Repository dialog

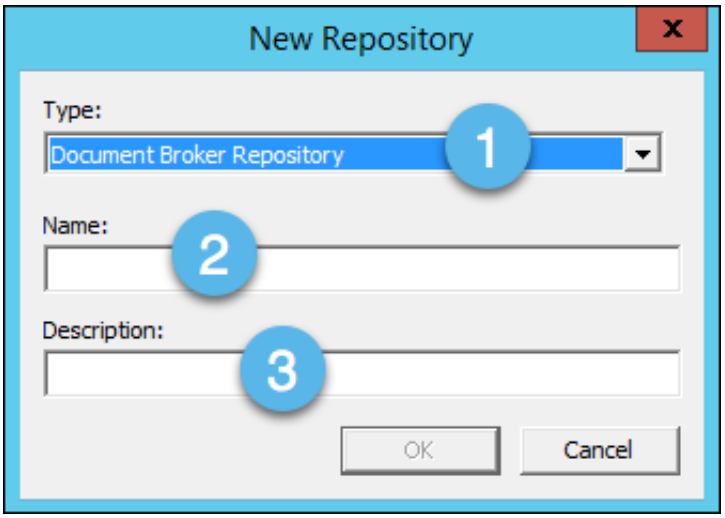


Figure 7-2:

New Repository

Option	Description
Type	See “Repository types” on page 7 - 1 for a description of each type.
Name	Repository name under which the repository will be listed in Control Center.
Description	A description of the purpose of the repository.

Settings – Create Database dialog

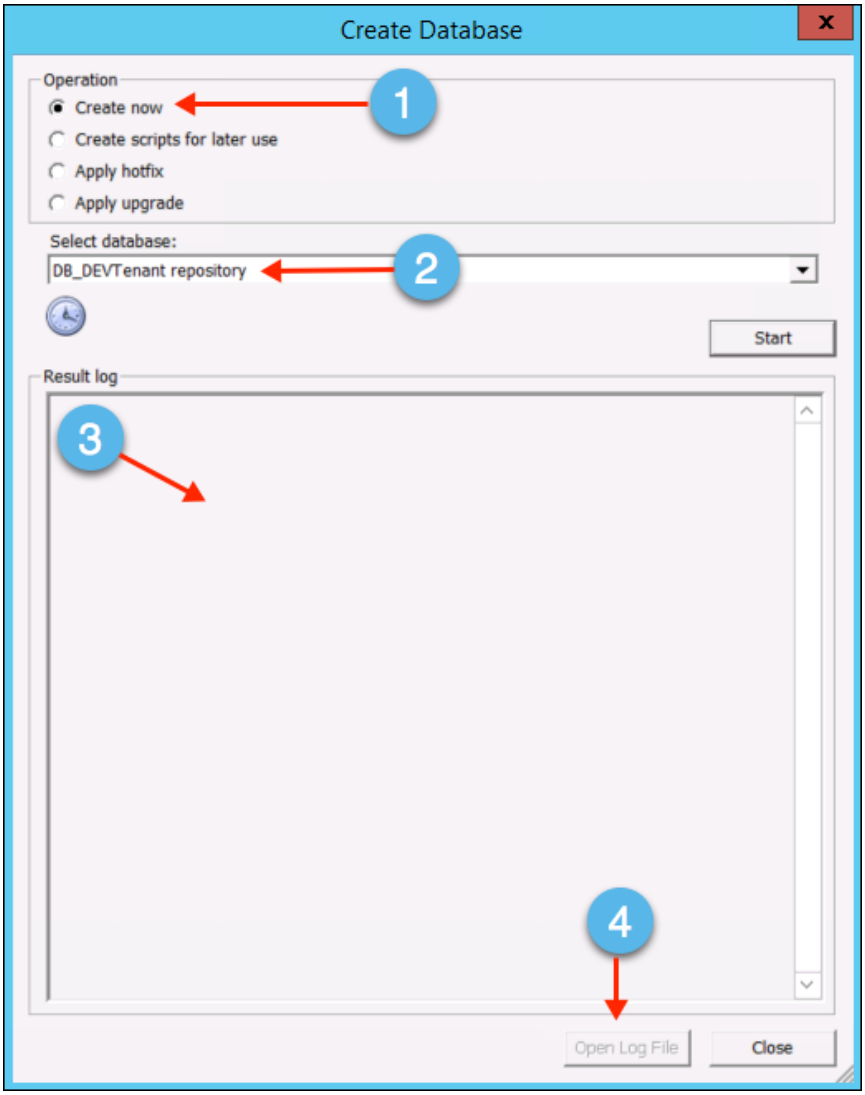


Figure 7-3:

Repository Create Database

Option	Description
Create now	Runs the scripts to create the repository selected in the drop down list.
Select database	Lists the Communications Server 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.

Lab: Creating the repositories

Create a repository

1. *In Control Center right-click the **Repositories** node and select **New Repository** in the pop-up menu.*

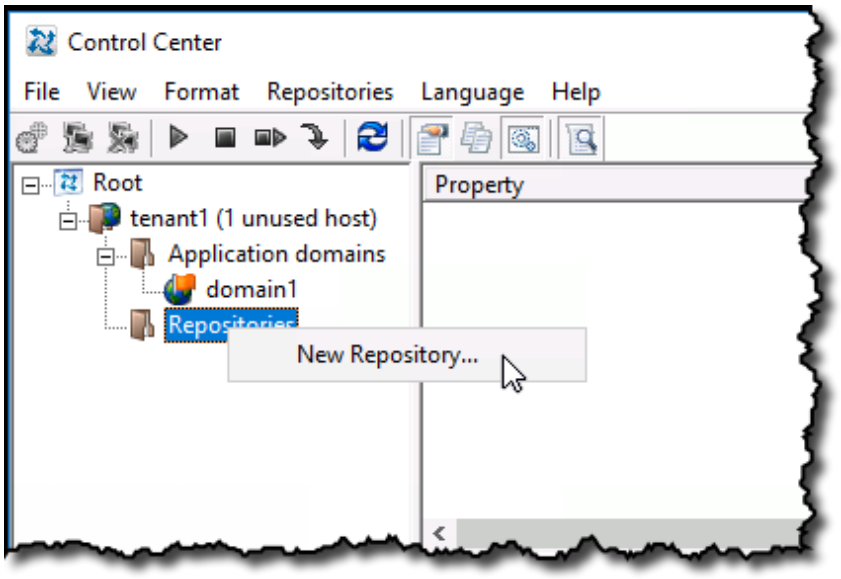


Figure 7-4:

New repository

2. *In the New Repository window enter the following information and click the **OK** button.*
   - *Type: **Document Broker Repository***
   - *Name: **DocBrokRepository***

The Configuration window displays.

3. *In the Configuration window enter the following information and click the **OK** button.*
   - *Host name: **server***
   - *Database name: **DB_DEVTenant***

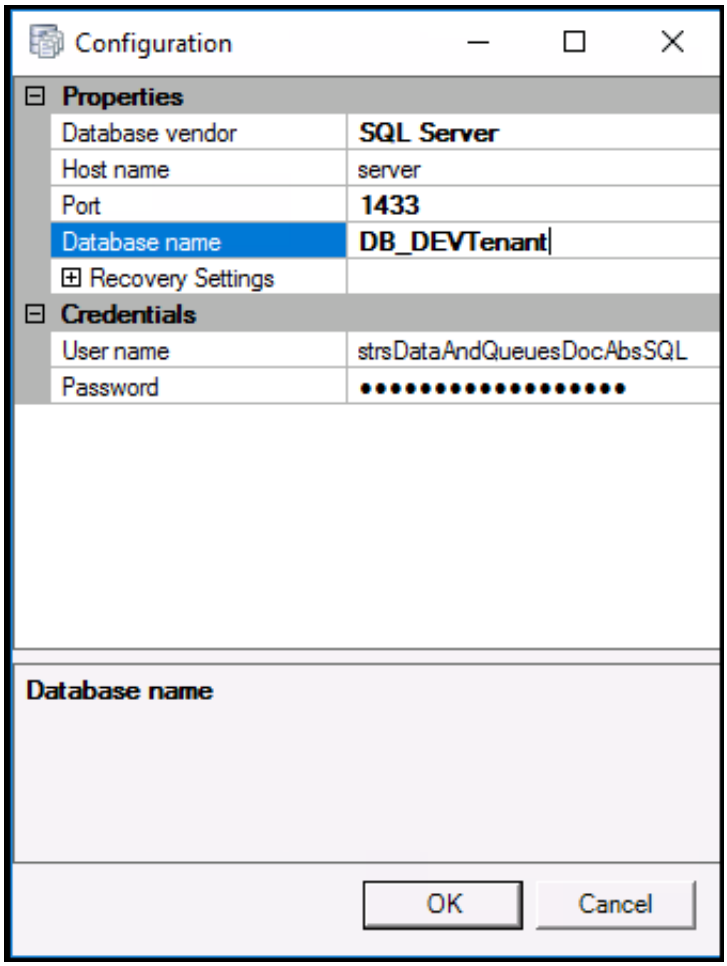


Figure 7-5:

Configuration

You are redirected back to Control Center displaying the newly created DocBrokRepository.

4. *Right-click **DocBrokRepository** and select **Create Database** from the pop-up menu.*

5. *In the Create Database window select the following information and click the **Start** button.*
   - *Operation: **Create now***
   - *Select database: **DB_DEVTenant repository***

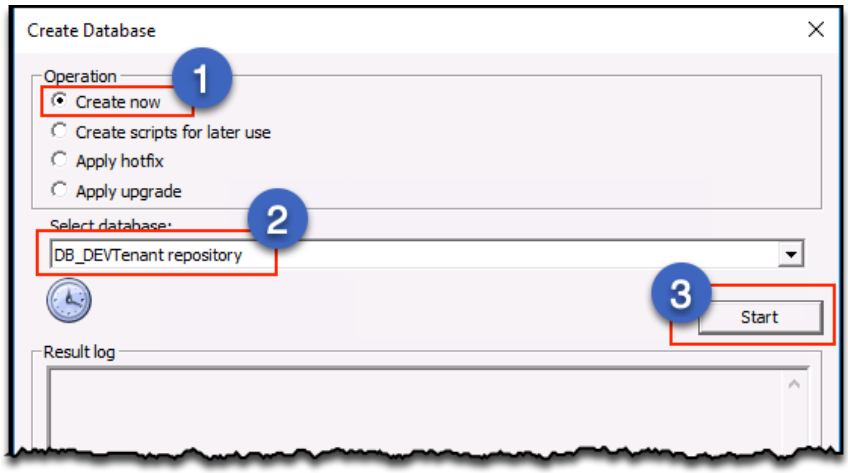


Figure 7-6:

Create database

6. *In the Connect window enter the following information and click the **OK** button.*
   - *User name: **sa***
   - *Password: **opentext***
   - *Remember password: **selected***

Once completed the Create Database window displays the Result log. (Make sure that there is no error.)

7. *Click **Close**.*

Link a repository to an application domain

1. *In Control Center right-click **DocBrokRepository** and select **Link Application Domain** from the pop-up menu.*

The Link Application Domains window is displayed.

2. *In the Available Applications Domains click the **domain1** and then click the **">"** button to move it to the Linked Application Domains panel.*

3. *Click the **OK** button.*

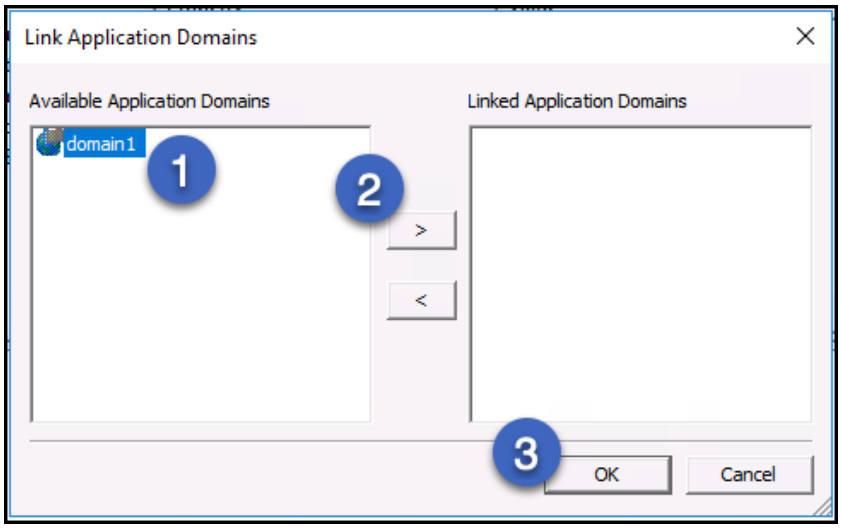


Figure 7-7:

Linking domain

The DocBrokRepository is linked to the domain1 application domain.

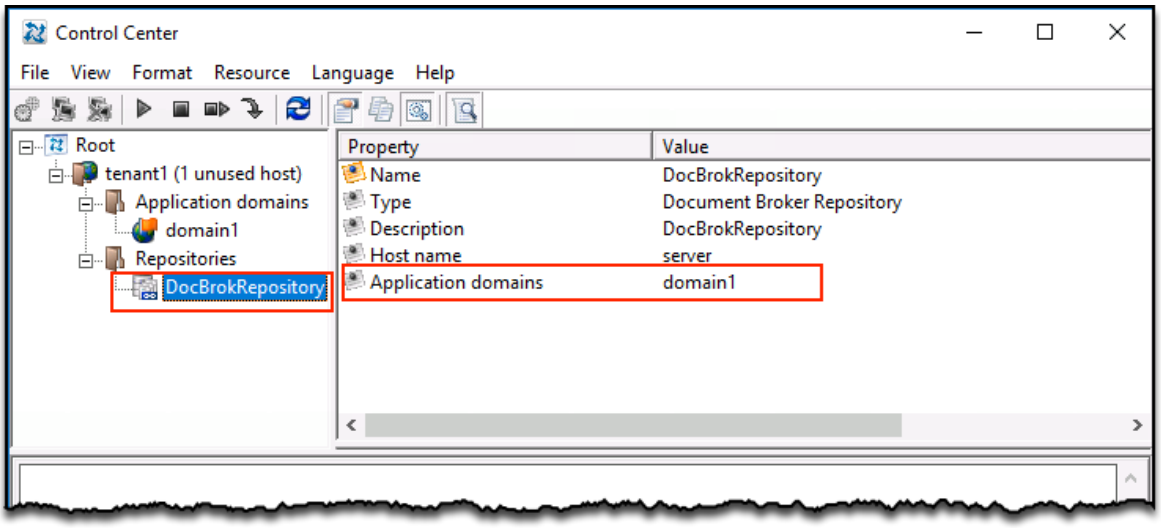


Figure 7-8: Linked repository

Create the other repositories

1. *Repeating the procedures from the two previous activities, create and configure the remaining repositories using the information provided below:*

Repository type	Repository name
Logging Repository	LogRepository
Messagestore Repository	MsgRepository
Queue Repository	QRepository
Statistics Repository	StatsRepository
Temporary Data Repository	TempRepository
Tracking Repository	TrackRepository

   For all the repositories use the following host and database:

   - *Host name:* ***server***
   - *Database name:* ***DB_DEVTenant***

2. *Link all the repositories to the* ***domain1*** *application domain.*

8. Communications Builder projects

Objectives

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

- Define project
- Identify the design tools for project creation
- Describe the use of Communications Builder
- Identify the project components
- Define platform
- Define platform layers
- Define message
- Define events and processes
- Define runtime
- Define resource set
- Identify the different administrative tasks that can be performed with projects
- Describe project packing, unpacking, exporting, versioning, checking in, releases, templates and deployment

Definition of a project

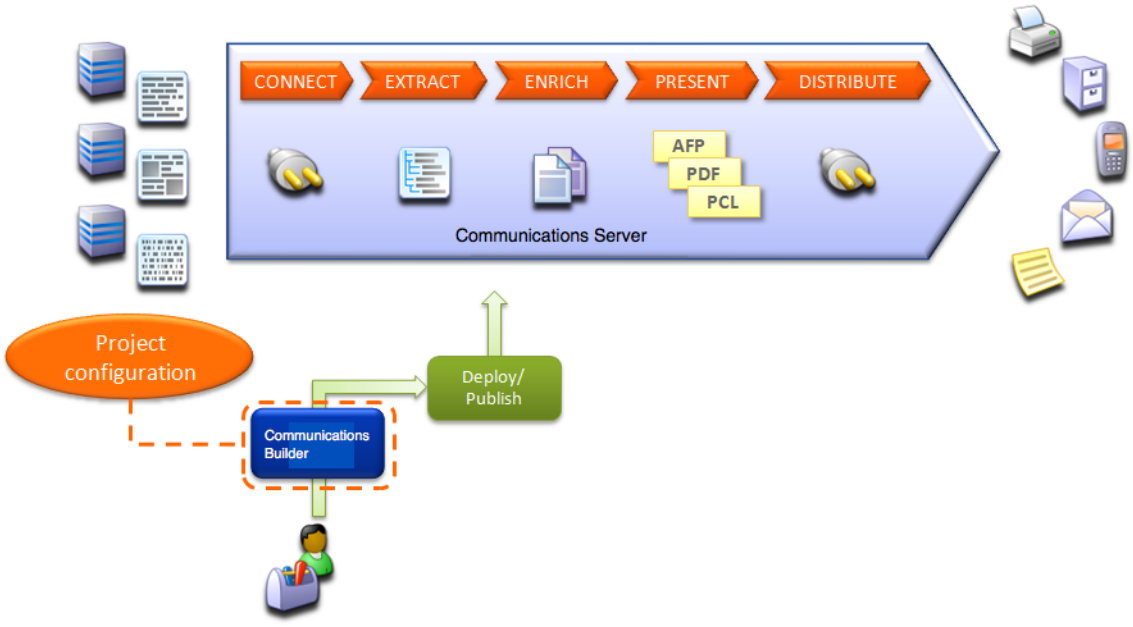


Figure 8-1: Definition of a project

A Communications Builder project is the design of how to:

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

In the project, you configure all the settings needed by the Exstream Communications Server application.

Communications Builder

The Communications Builder graphical user interface (GUI) consists of four windows, and one view per Project component. The component views are displayed in the main window.

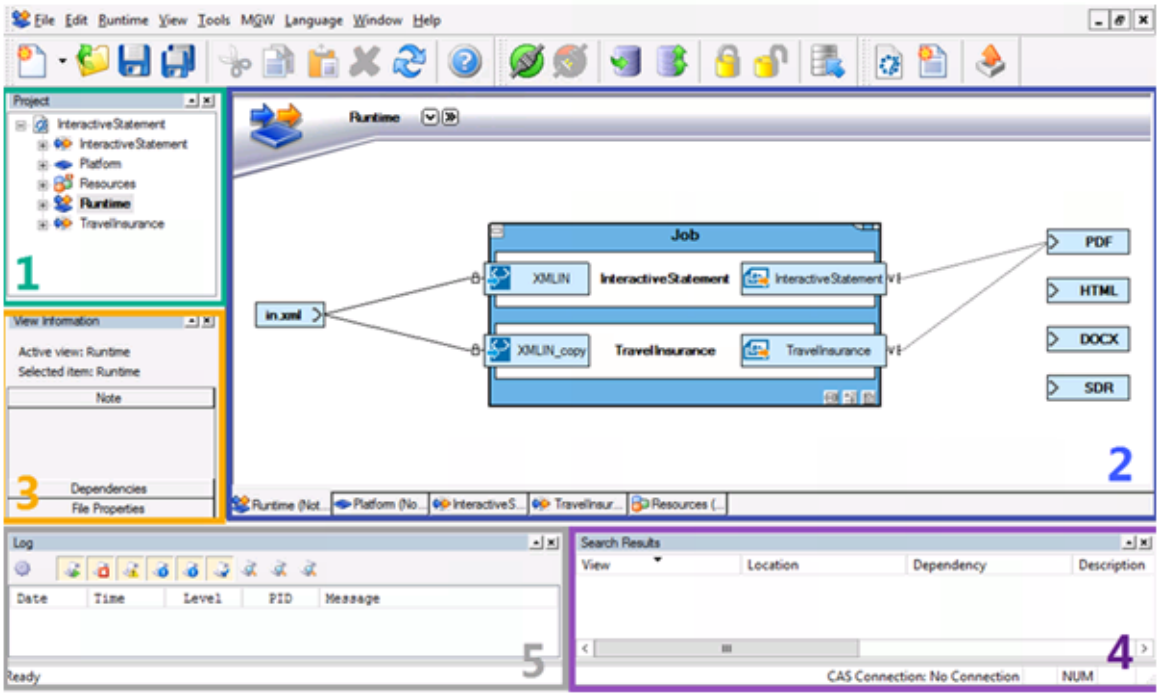


Figure 8-2: Communications Builder UI

Project browser The Project browser is where you create and structure the Project. The Project is displayed as a tree, and all Project components are added as nodes to the Project tree.

Main window The Main window is where you configure the Project components.

Property window The Property window displays properties for the active view in the Main window.

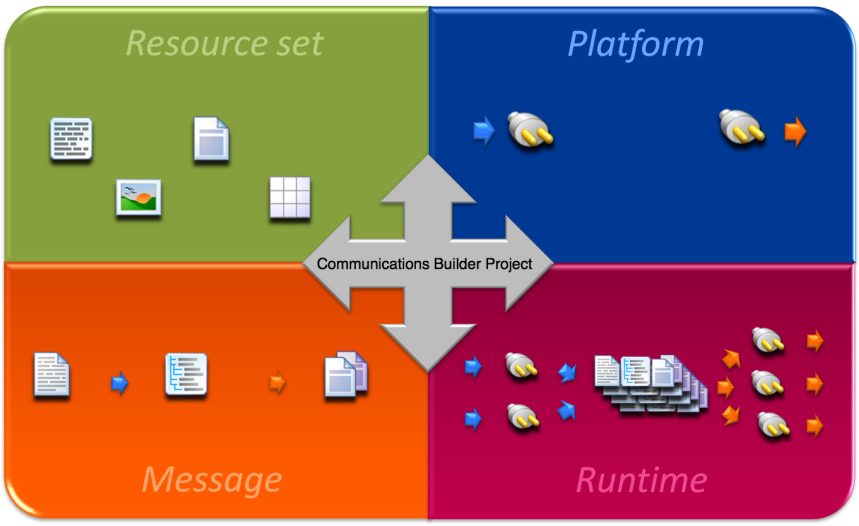
Search results window The Search results window displays the search results after submitting Edit> Find.

Log window The Log window displays the Communications Builder log.

Communications Builder project components

Figure 8-3:

Communications Builder project components



A Communications Builder project contains the following main components:

- Platform
- Message/Processing Engine
- Runtime configuration
- Resource set

Platform

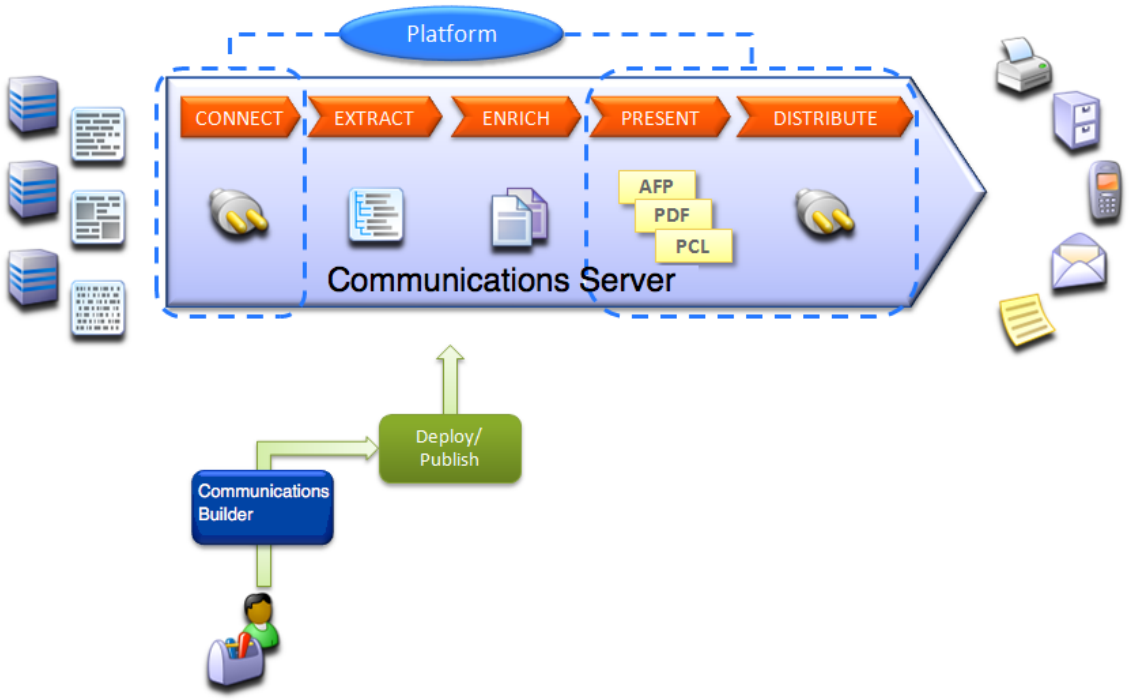


Figure 8-4: Platform

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:

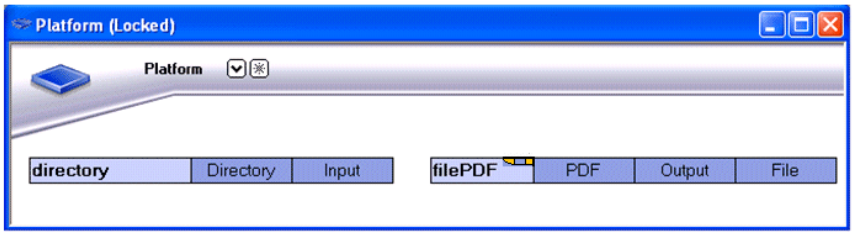
- Input connectors specify how to receive input.
- Output connectors specify how to deliver output.

Platform view

The Platform view is where you configure the Platform settings.

Figure 8-5:

Platform view



Connectors All connectors are added to the Platform view. A connector contains labels that helps you identify the name, type, queue, etc. specified for the connector.

Input connectors have the labels in the following order:

Name >> Type >> Queue.

Output connectors have the labels in the following order:

Name >> Driver >> Queue >> Type.

Platform layers

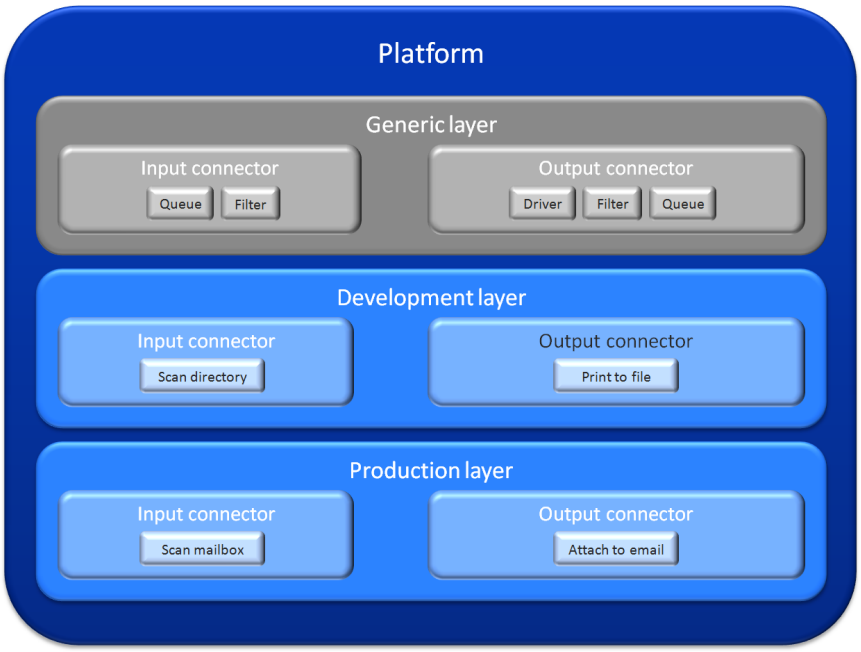


Figure 8-6:

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 Communications Builder 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:

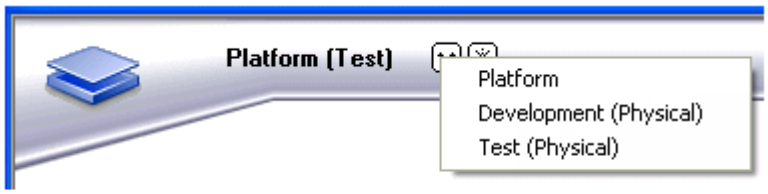
- The connector type.
- The settings for the selected connector type, for example, which host to receive the input data from.
- Log level.

Navigating between layers

The generic layer and all physical layers are activated in the same Platform view. The banner at the top of the view shows which layer is active. You can use the drop-down list in the banner to navigate between the different layers.

Figure 8-7:

Nvigating layers



Message

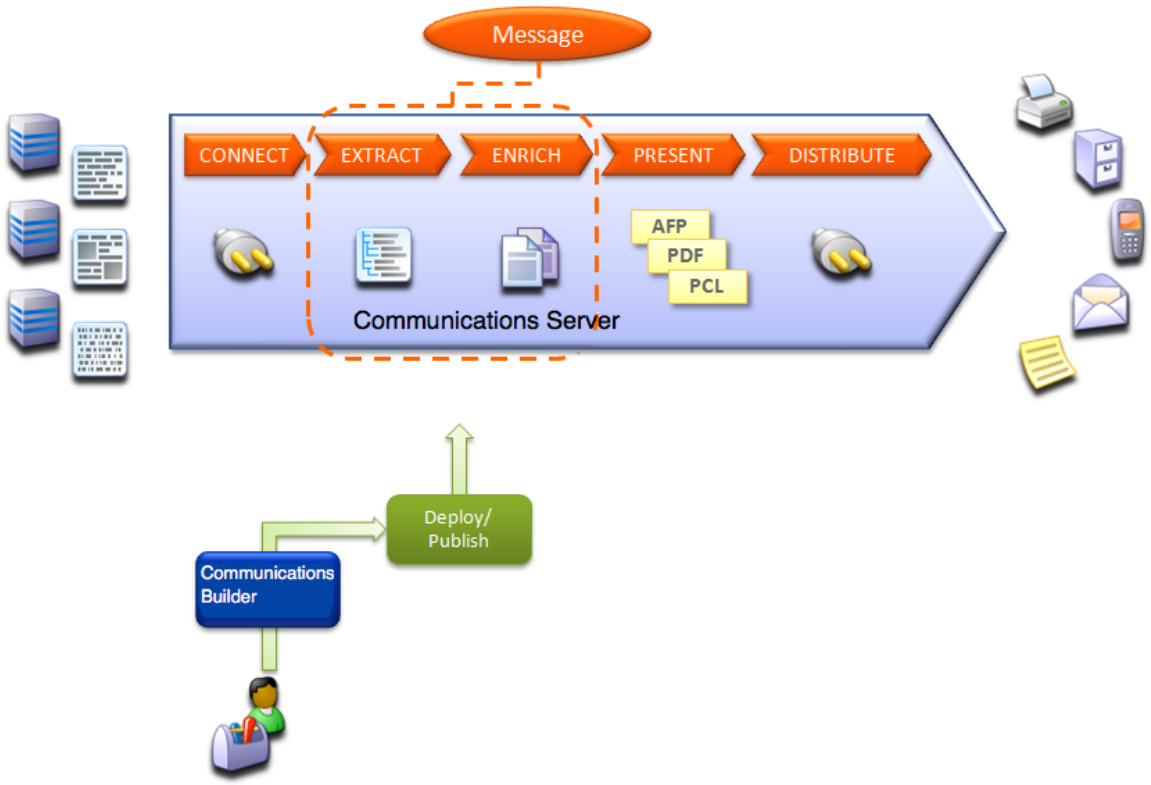


Figure 8-8: Message

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.

Message View

The Message view is where you configure Messages.

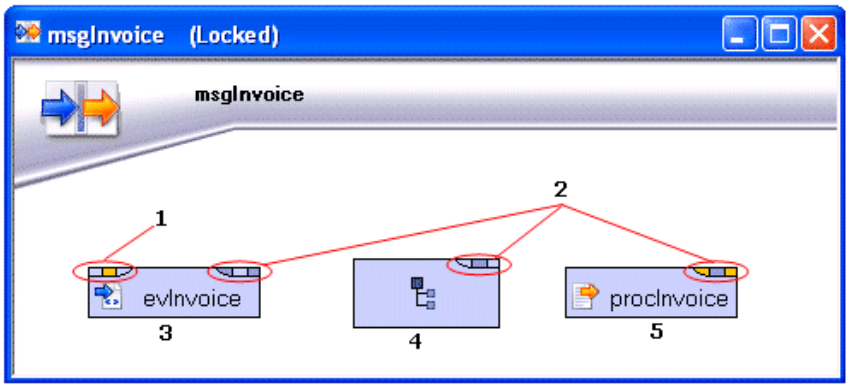


Figure 8-9:

Navigating layers

Where:

1. Event order indicators
2. Script indicators
3. Event
4. Message
5. Process

Events and processes

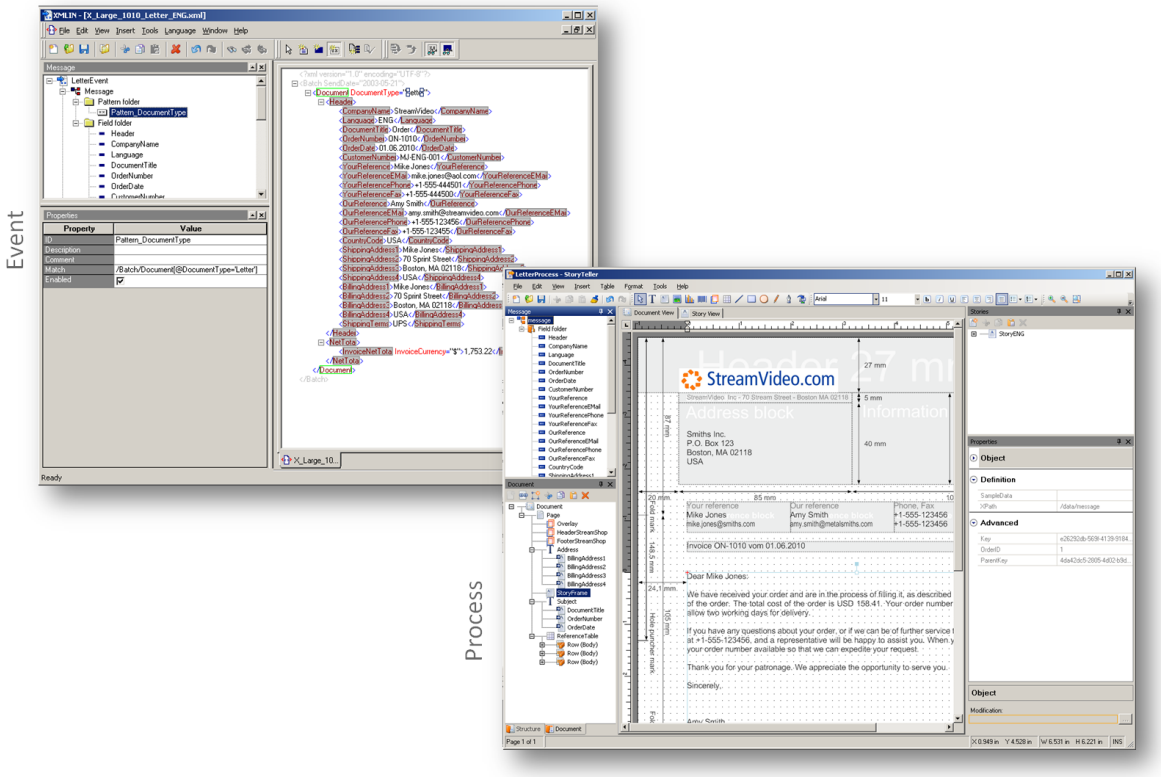


Figure 8-10: Events and processes

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.

Processing Engine

Processing engines are plugins for external services integrated into Communications Server. For example, configurations made in Exstream Design Manager can be exported from Exstream Design Manager and included as a processing engine in Communications Builder.

Figure 8-11:

Processing engine

Where:

Input channels The input channels corresponds to Events in a Message configuration.

Output channels The output channels corresponds to Processes in a Message configuration.

Runtime configuration

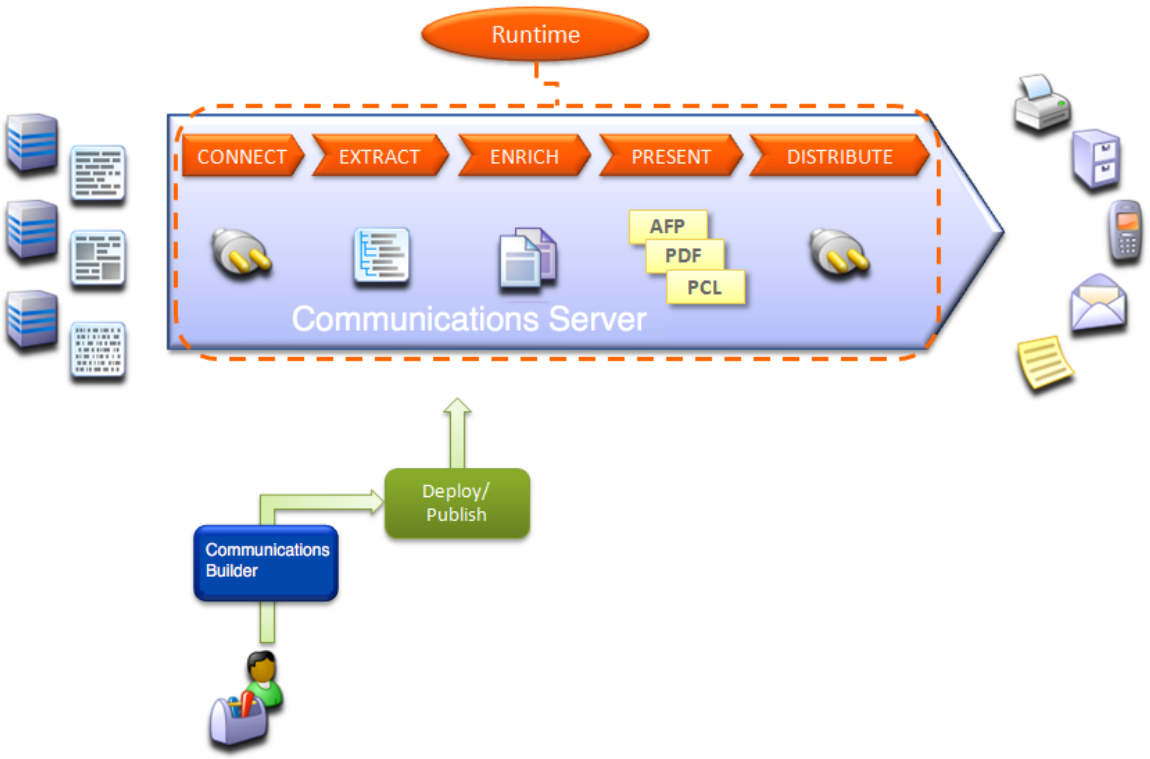


Figure 8-12: Runtime configuration

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.

Runtime view

The Runtime configuration view is where you connect processing components to the connectors, and configure runtime specific settings.

Runtimes contain:

Jobs A job is created by default when you create a Runtime configuration. You can add new jobs to the Runtime configuration if you need to.

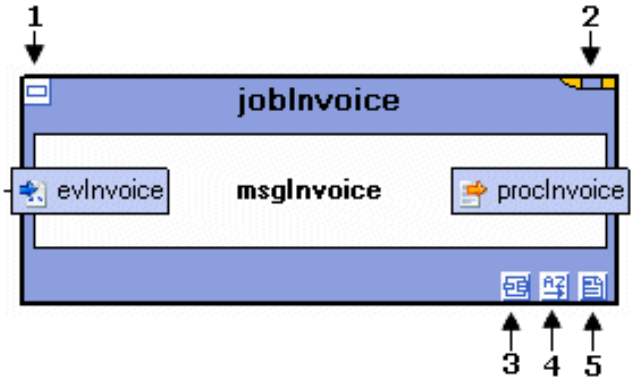


Figure 8-13: Jobs

Where:

1. Expand/collapse job
2. Script indicators
3. Add Message or processing engine
4. Edit sort criteria
5. Edit script

Post-processors You must add a post-processor to the Runtime configuration to retrieve documents from a post-processor repository.

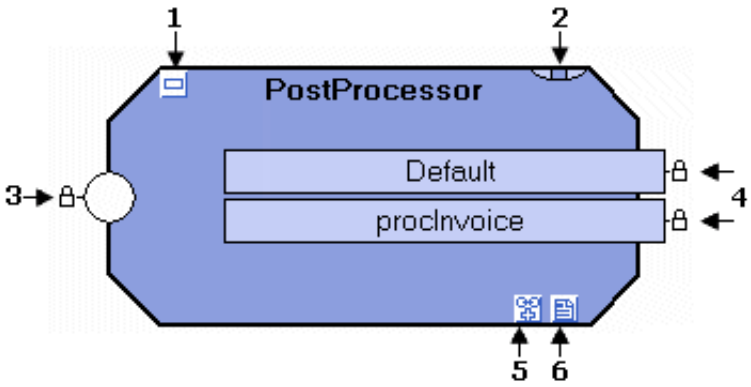


Figure 8-14: Post-processors

Resource Set

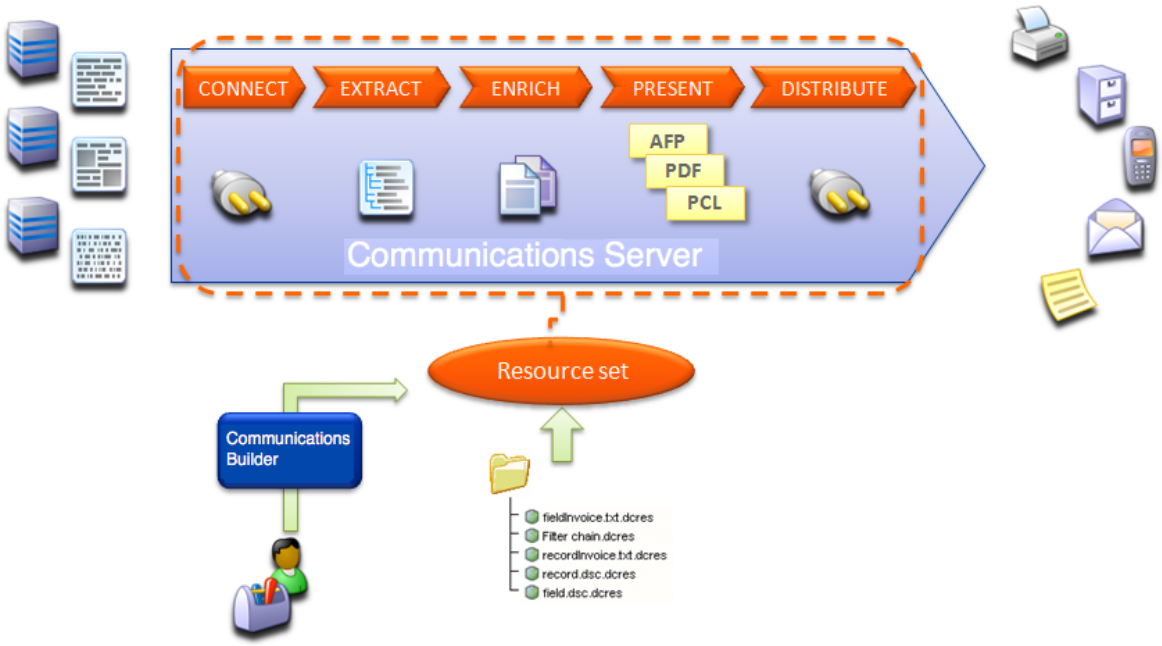


Figure 8-15: Resource Set

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
- Fragments
- Tables files
- Function files
- Sample files

Managing projects

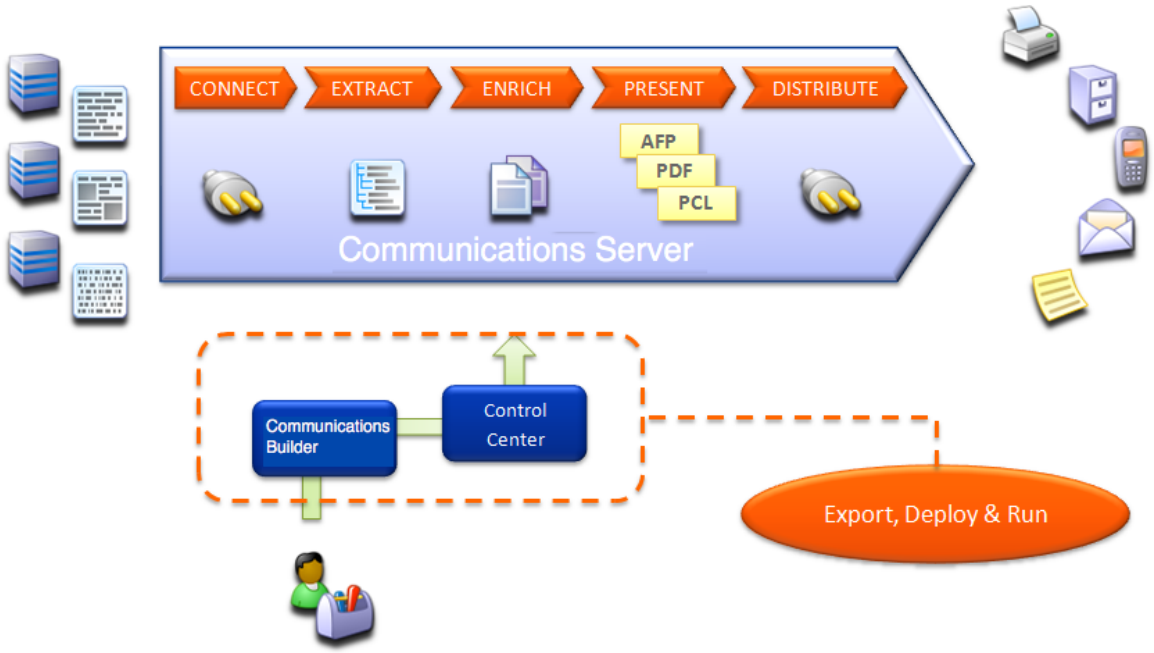


Figure 8-16: Export, Deploy, Run

You can perform the following actions with projects:

Pack A Communications Builder Project consists of several project files stored on disk or CAS. If you want to move a project to a new location, you can pack all project files into a single package file (*.dcpackage). Then you move the package file to the new location, and unpack it there.

Unpack You can unpack a Project package, and create a copy of the packed Project. When you unpack a Project, the Project files are stored in a predefined directory. You cannot change the location of this directory.

Export You can export the project to disk. The export generates a file (*.export) that contains all the platform layers in the project. Exported projects can be deployed into a Exstream Communications Server application in Control. When you deploy the project in Control Center, you must specify which physical layer to deploy.

Version When you export a project, the project is assigned a version number. Unless you update the number manually, this is always version number 1.

Check in Projects can be checked in to CAS. The Project is connected to CAS via the management gateway connection in Communications Builder. Before you can check in a Project, you must connect to the management gateway.

Create releases When you create a release, the project is exported, all project files are checked in to CAS, and the export file is added to CAS. The export generates a file (*.export) that contains all the platform layers in the project. Each time you create a release, a new version of the export file is stored in the CAS.

Create project templates You can save a project as a template. From a template you can create new projects that share the same structure, functionality and resources.

Deploy Once a project is exported, it is possible to deploy it to a Exstream Communications Server application to run it. (Deployment is covered in the Communications Server applications chapter.)

Lab: Project Management

Create a project from a project package

1. *Launch **Communications Builder**.*
2. *From the **File** menu, select **Unpack Project**.*
3. *In the **Open** window, navigate to **C:\Training\3-3730 EXS - System Administration Files**, select **AdminCourse.dspackage** and click the **Open** button.*

The Unpack Project window displays. Notice the available options to unpack the project.

4. *In the Unpack Project window, click the **OK** button.*

The Project Settings window displays. Notice the project settings available. If the project was created with a previous version of Communication Builder, as it is in this case (v11.0), then the project will be updated to the current version.

5. *In the Upgrading files to "16.6.0 GA" click the **OK** button.*
6. *In the Project Settings window leave the default values and click the **OK** button (notice that you can change the project name, code page and Resource set).*
7. *Select **Save Project** from the **File** menu.*

A new project is created in Communications Builder containing the files of the AdminCourse.dcpackage.

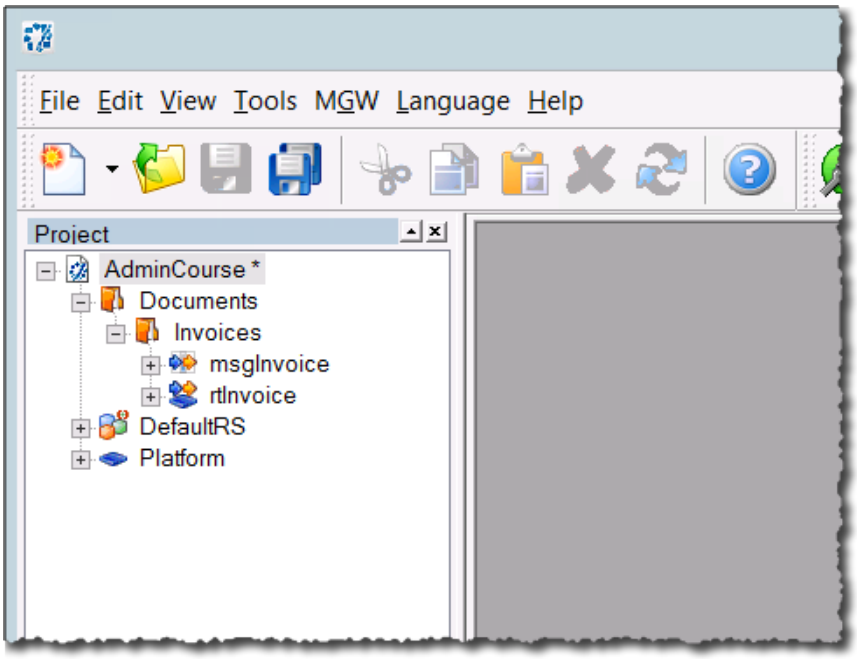


Figure 8-17:

New Project

8. *Review the different components of the project and their configurations:*
   - *Messages*
   - *Resource Set*
   - *Platforms*

Check in a project to CAS

1. *In the Communications Builder toolbar, click the* ***Check in*** *button.*

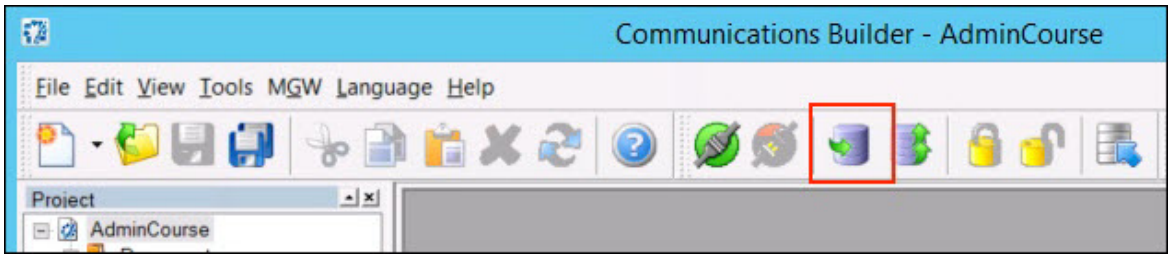


Figure 8-18: Check in

The Select Management Gateway window opens.

2. *In the Select Management Gateway window, select the **tenant1Connection** and click the **OK** button.*

The Login to Management Gateway window opens.

3. *In the Login to Management Gateway window make sure the following values are being used and click the **OK** button:*
   - *Tenant: **tenant1***
   - *User: **exadmin***
   - *Password: **opentext***

The Check in window opens.

4. *Enter **First Checkin** in the Label field and click the **OK** button.*

A message displays indicating that the check-in was successful.

Create a project release

1. *In the Communications Builder toolbar, click the **Create release** button.*

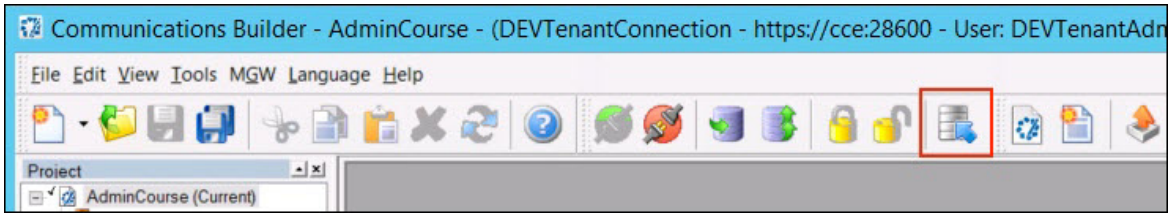


Figure 8-19: Release

The Export for release window opens. Notice the values that you can set for the release export.

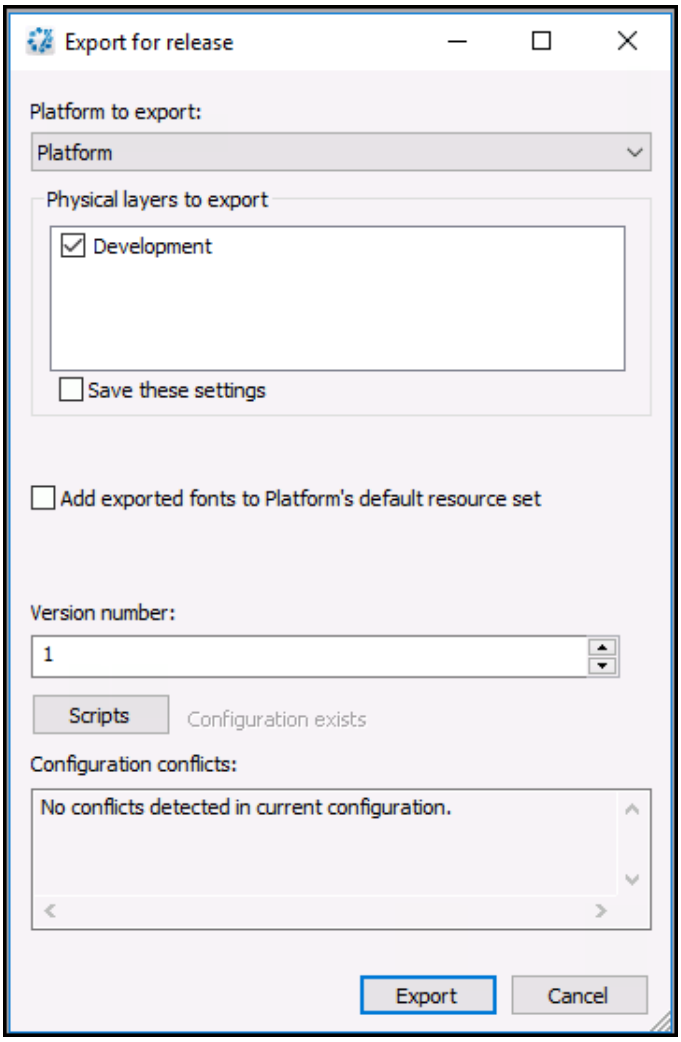


Figure 8-20:

Release export

2. *In the Export for release window, leave the default values and click the **Export** button.*

The Create release window opens.

3. *Enter **First Release** in the Label field and click the **OK** button.*

A message displays indicating that the release was created successfully.

Export a project

1. *In Communications Builder toolbar, click the* ***Export*** *button.*

Figure 8-21: Export

The Export window opens.

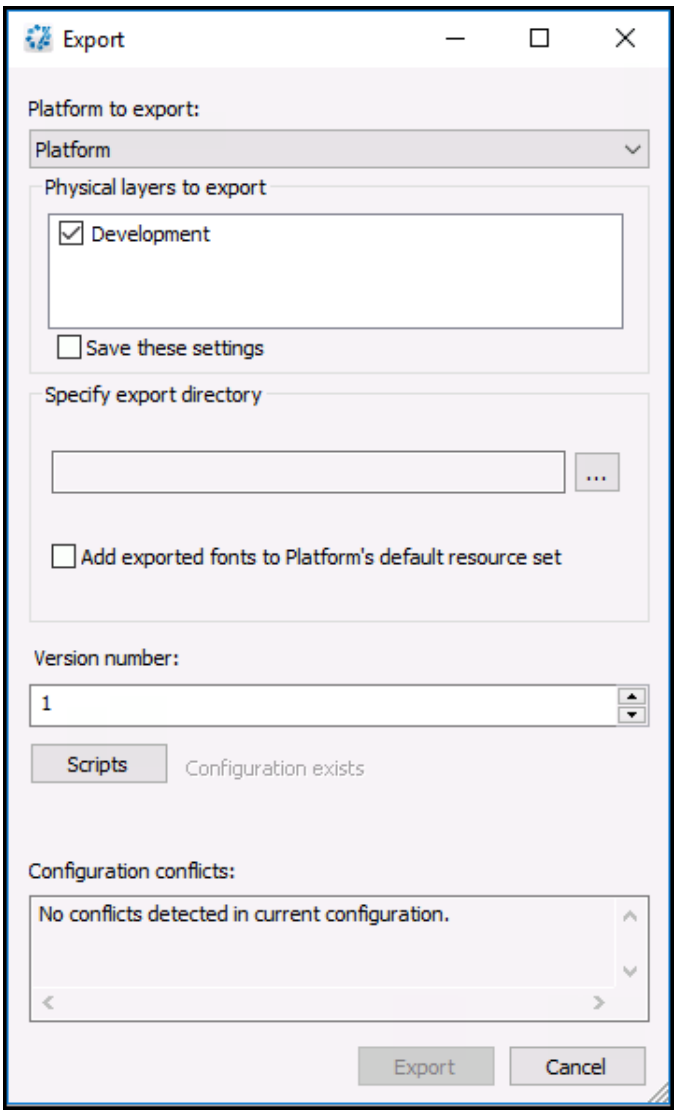


Figure 8-22:

Export

2. *In the Export window, enter **C:\Temp** in the Specify export directory field and click the **Export** button.*

The project is exported.

In the next chapter you will deploy and run the project.

9. Communications Server applications

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

- Define what Communications Server applications are
- Describe the process of creating an application
- Manage Communications Server applications
- Describe the directory structure of Communications Server applications
- Locate an application's working directory
- Deploy a project to an Communications Server application
- Test an application

About Communications Server applications

Communications Server applications run exported Communications Builder Projects. Each Project is deployed to and run by a separate Communications Server applications. You can use several Communications Server application, running different deployed Projects, in the same domain.

For failover reasons, you can deploy a Project to more than one Communications Server application and let the Communications Server applications share queues. This means that jobs can be reallocated if the Communications Server application processing the job goes down or loses connection to the repositories.

If you run several versions of a Project, each Project version requires a separate Communications Server application.

Adding Communications Server applications

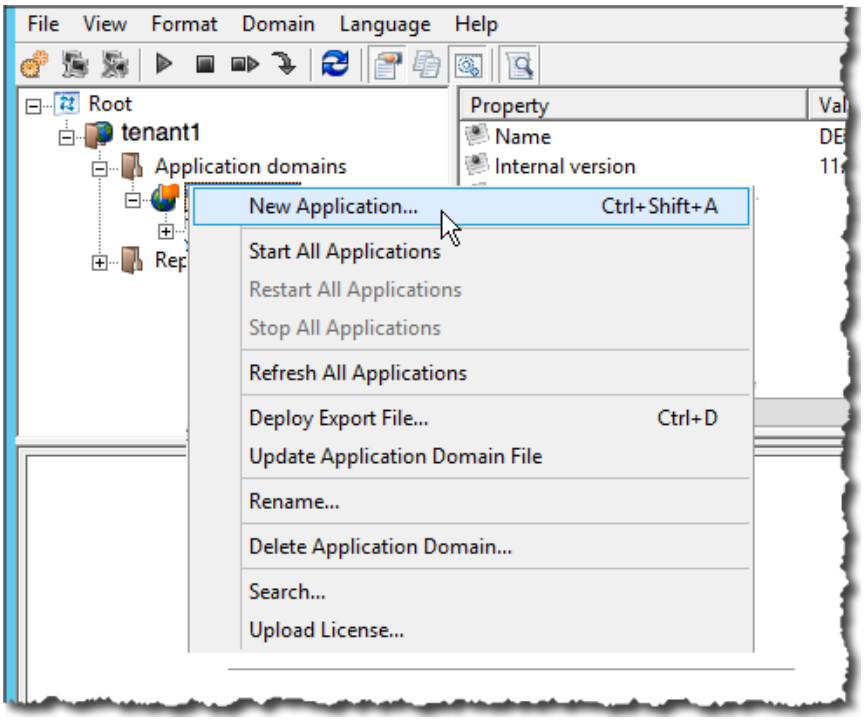


Figure 9-1:

Adding an Communications Server application

If you run several versions of a Project, each Project version requires a separate Communications Server application.

Logical names

You can assign logical names to Communications Server applications. You can, for example, use the name of the corresponding Communications Builder Project as logical name. The logical name is used as identifier when deploying a Project to a Communications Server application. For example, if a domain includes two Communications Server applications with the same logical name (for example. for load balancing), when you deploy you can select whether to deploy to both applications or to one Communications Server application only.

New Application dialog box

The New Application dialog box is used to add Communications Server applications.

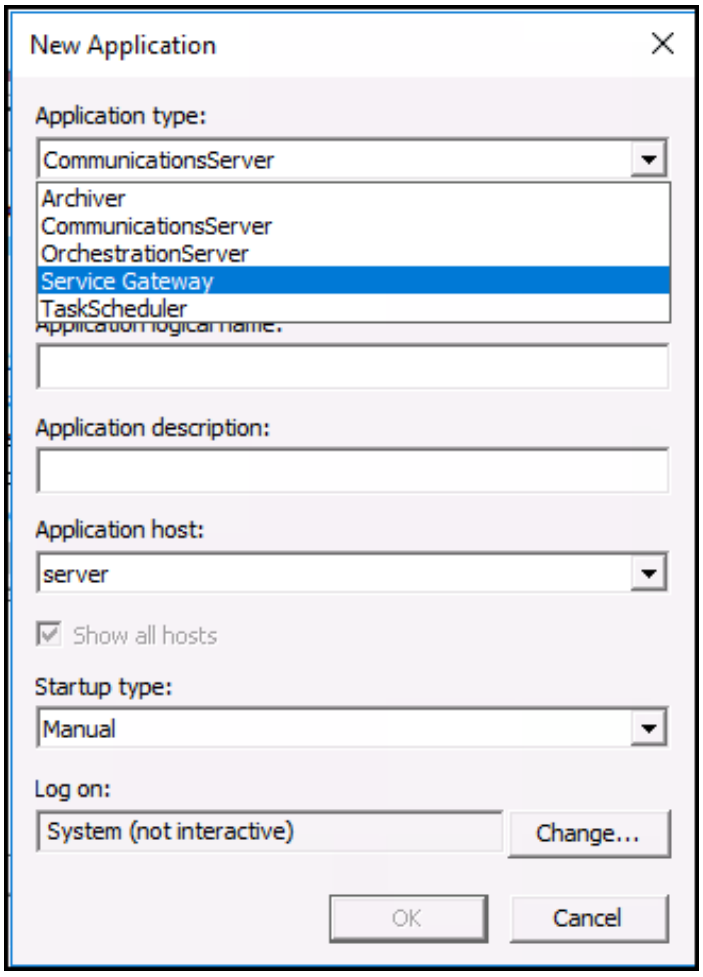


Figure 9-2:

New Application dialog box

Settings – New application dialog box

Field	Description
Application type	The type and version of the application (Archiver, Service Gateway, Communications Server, Orchestrator Server or TaskScheduler).
Application name	A logical name for the application, that can be used for development purposes.
Application logical name	A logical name associated with the application.
Application description	A description of the application.
Application host	The computer used to run the application. Select **Show all hosts** to display all the hosts that are part of the site.
Startup type	Specifies how the application is started. **Automatic** - Starts the application automatically when the system (host for the corresponding application) starts. **Manual** - Starts the application manually. **Disabled** - Disables the application.
Log on	The account used to run the application.

Managing Communications Server 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.

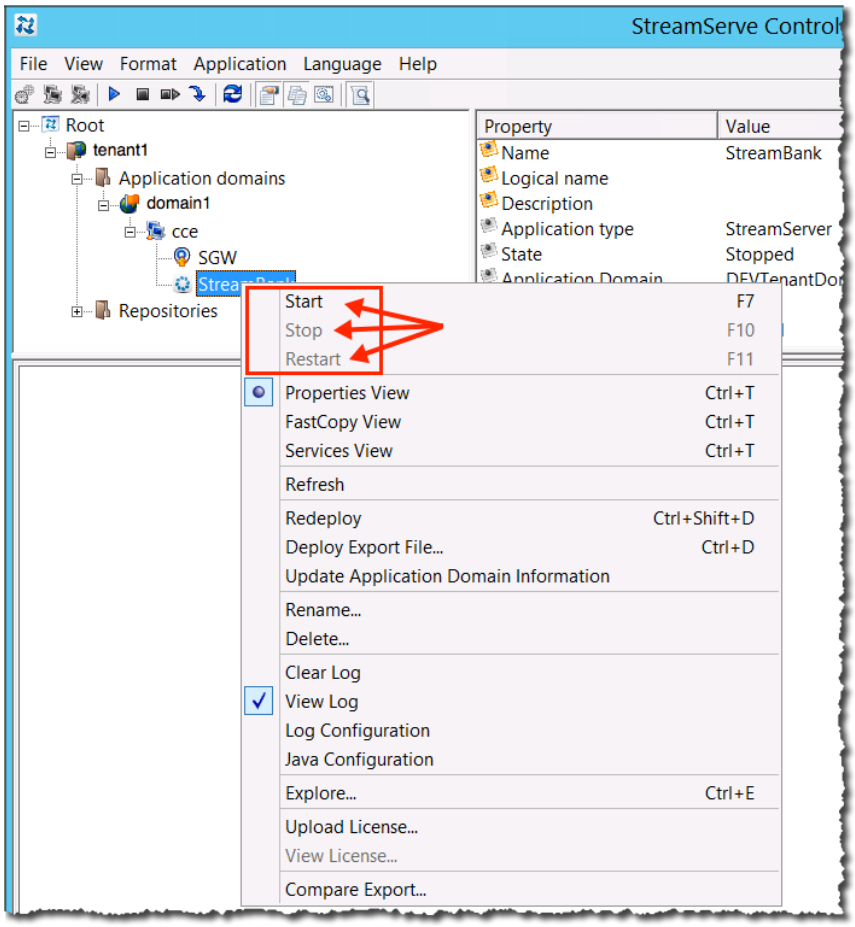


Figure 9-3:

Start, stop and restart applications

Deleting applications

You can delete Communications Server applications. This does not delete the working directory. You must manually delete this if required.

The working directory is covered in the Disk Structure later in the course.

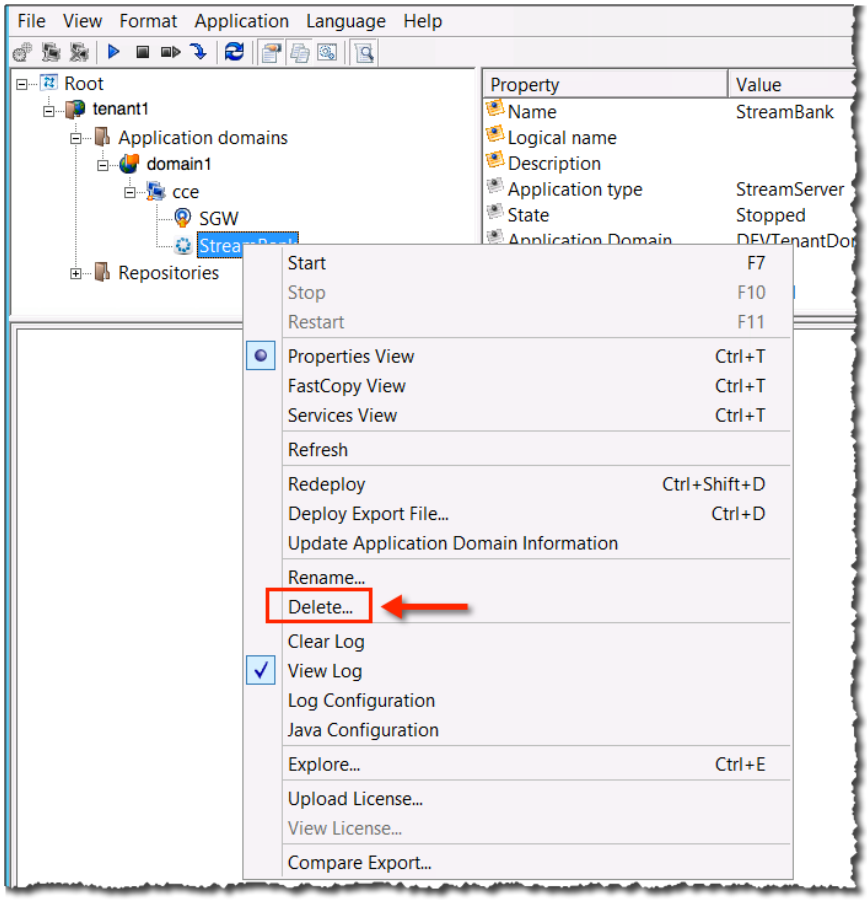


Figure 9-4:

Deleting applications

Updating application properties

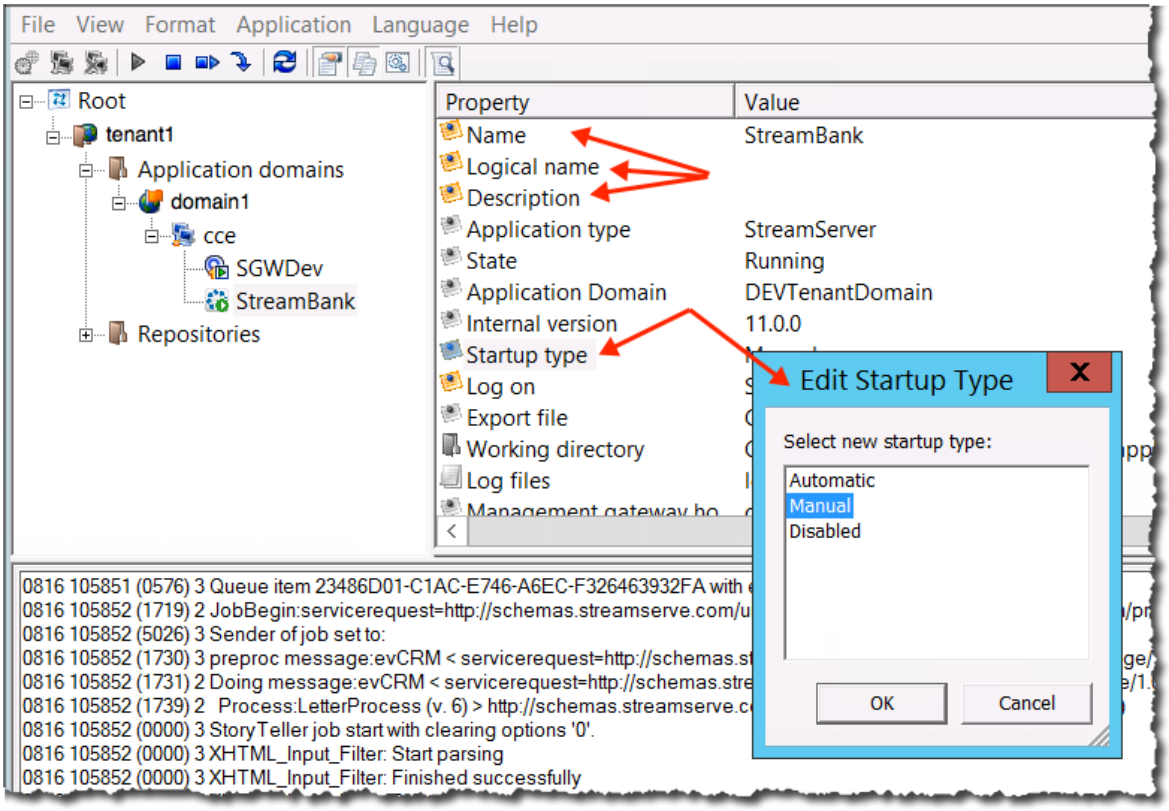


Figure 9-5: Updating application properties

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.

Adding other Communications Server applications

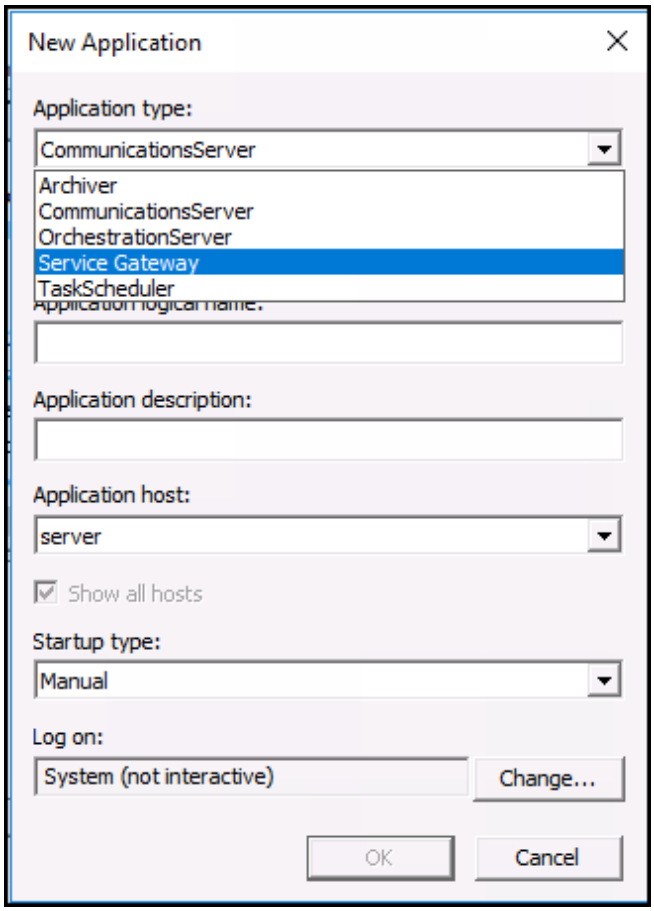


Figure 9-6:

Adding other Communications Server applications

New application dialog box

In addition to Communications Server applications, this dialog box is also used to add the following application types:

Application Type	Description
Archiver	This application type is used create applications to store documents in a Collector Archive repository.
Service gateway	To use the Exstream web applications, you must add a service gateway. For failover and load balancing reasons, you can add more service gateways. When being accessed, a web client sends a request to the management gateway, asking for a service gateway endpoint. The management gateway checks the tenant repository for available service gateways, and then returns the connection information for one of the service gateways. The web client uses the returned service gateway to access the Exstream repositories, the common asset service, and the metadata model.
Task Scheduler	This applications 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.
Orchestrator Server	Communications Orchestrator is a web application used to create flow models for customer communications management processes. A flow model starts with input channels through which source data enters the flow model, and ends with output channels through which the output is delivered. To this flow model you can also add nodes for data processing. The data flow in the model is defined by drawing connections between the flow model nodes. A connection can be static, i.e. always used, and you can also set conditions to make connections available only if the conditions are fulfilled when the flow model is run.

Administering applications on remote hosts

You can use Control Center to run and administer Communications Server applications on both Windows and UNIX hosts.

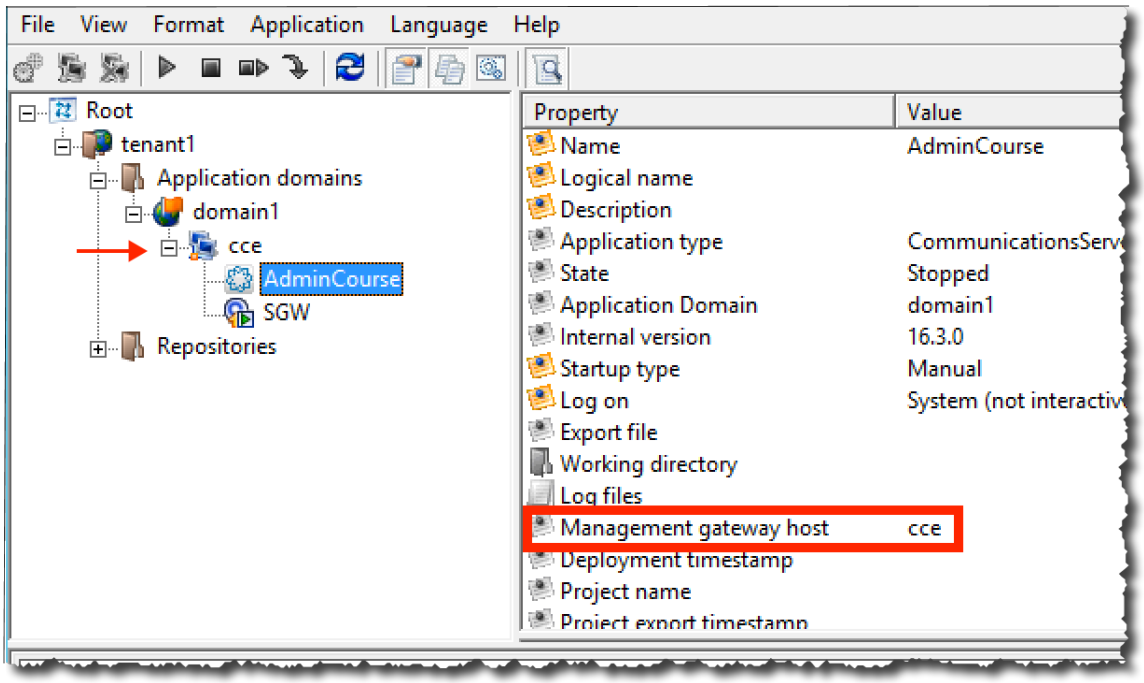


Figure 9-7: Administering applications on remote hosts

Remote hosts are displayed in the Control Center tree view after you create an application on the host.

To create an application and perform administrative tasks on the remote host, you need a user name and password to the management gateway.

Directory structure for Communications Server applications

Base directory for projects

During the installation of the Exstream software, a base directory is created for all Exstream Communications Builder projects that are deployed in Control Center.

The default path for the base directory is:

<*sysdrive:*>\ManagementGateway\16.6.0\root\applications

Working directory

Each Communications Server 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 Communications Server 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

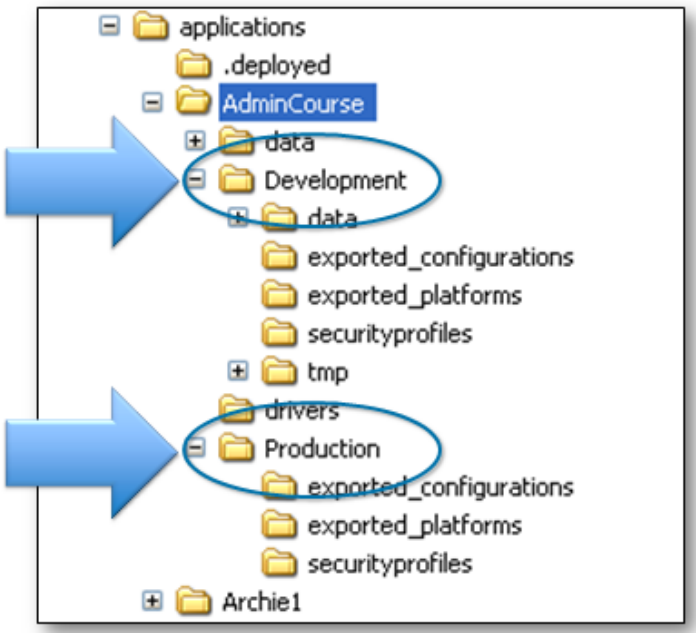


Figure 9-8:

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 Communications Server application named AdminCourse that has two physical platform layers. The layers are named Development and Production, which are the working directories.

Deploying a Communications Builder project

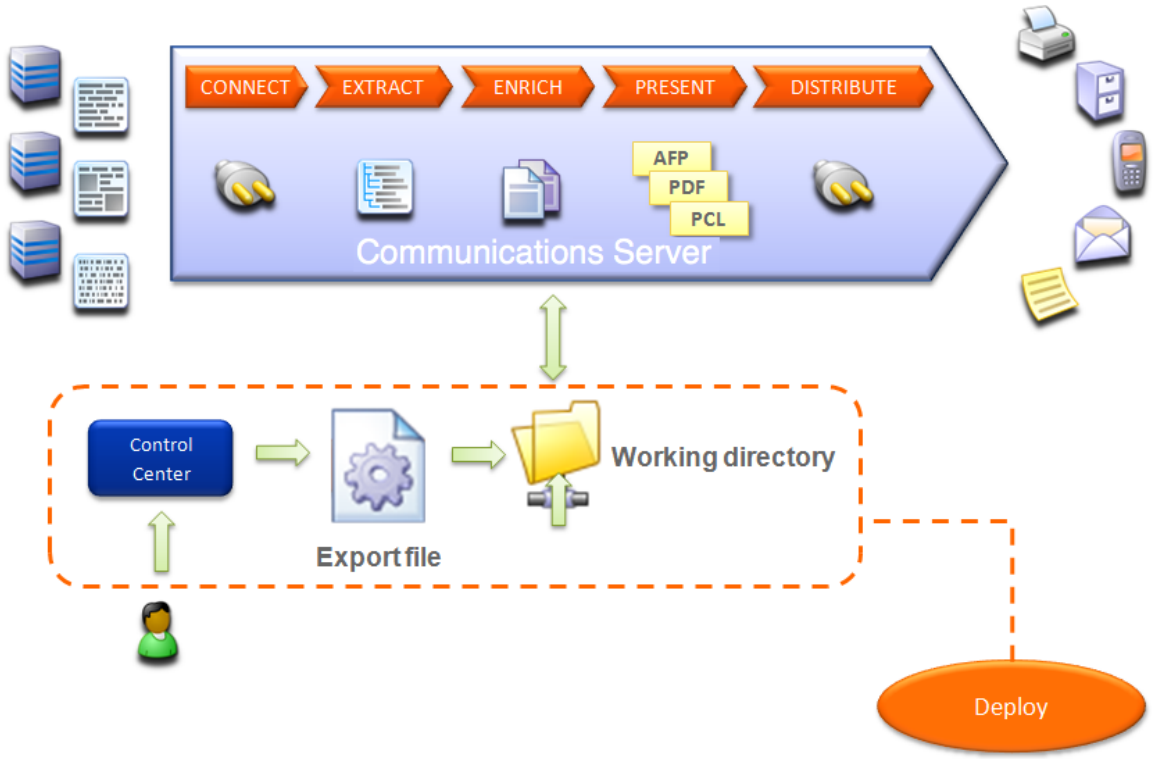


Figure 9-9: Deploying a Communications Builder project

To run Communications Server applications, you must deploy an export file for a project to an application.

Each time you export a new version of the project in Communications Builder, 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.

There are two methods to deploy export files:

- From the file system
- From CAS

Deploy wizard – Select the applications

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

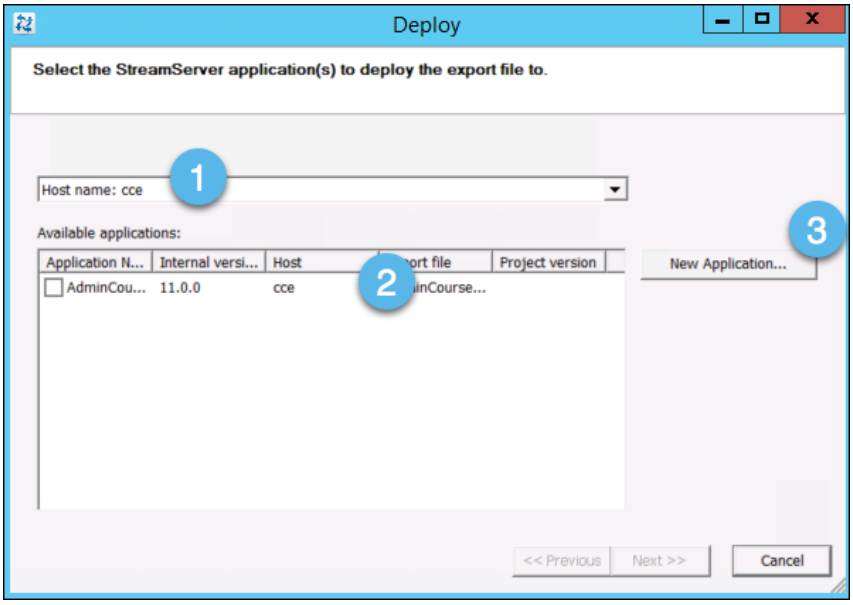


Figure 9-10:

Deploy wizard – Select the applications

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

Settings – Deploy Wizard – Select application

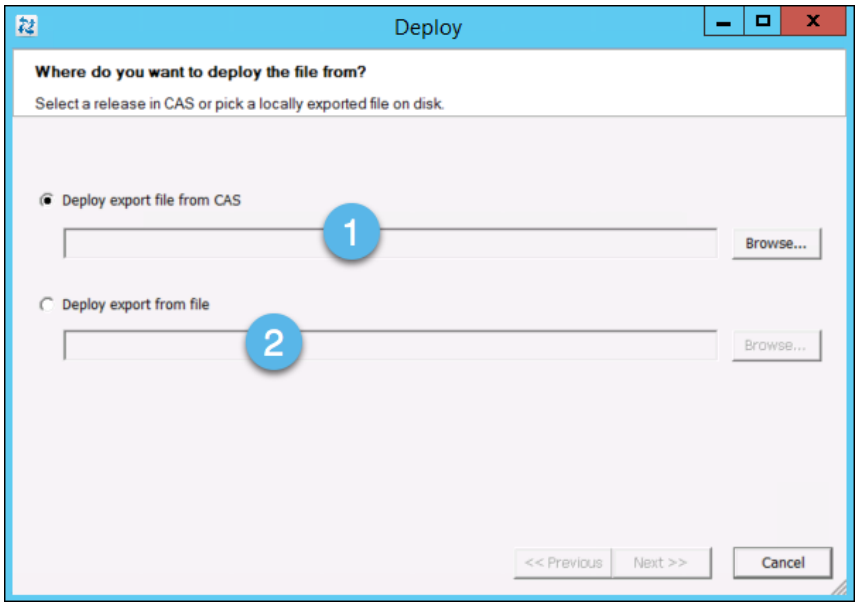
Field	Description
(1) Host drop-down list	Select the host that will be used to run the application.
(2) Available applications	Lists the existing applications at the site or for the host. **Application Name** - The name of the application. Exstream Communications Server version - The version of the application. **Host** - The host used to run the application. **Export file** - If there is already an export file deployed to the application, the name of the export file.
(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

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

Figure 9-11:

Deploy wizard – Select file



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 Communications Builder.

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

Settings – Deploy Wizard – Select export file

Field	Description
(1) Deploy an export file from CAS	Used to select an export file stored in CAS. You must select the connection profile to connect to CAS.
(2) Deploy export file	Used to select an export file from the file system. Click **Browse** to select the file.

Deploy wizard – Select Physical layer

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

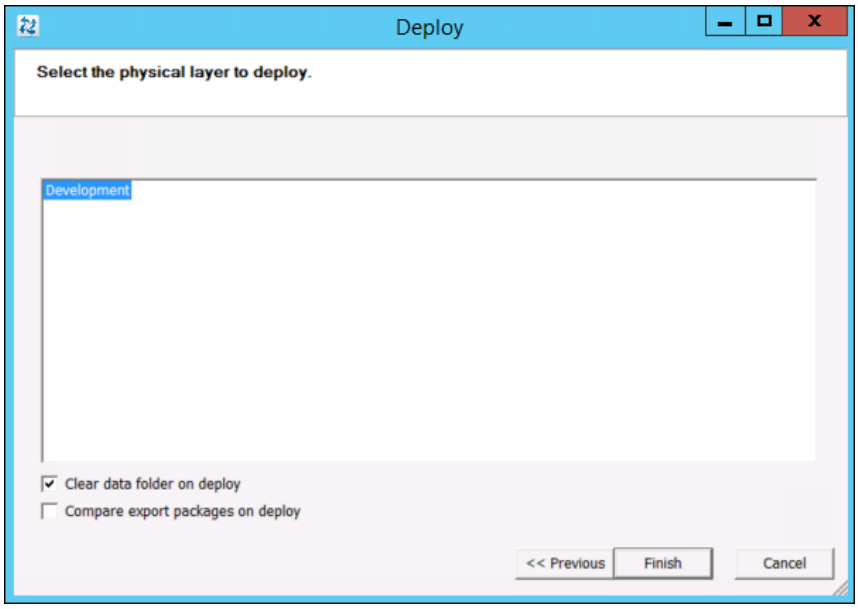


Figure 9-12:

Deploy wizard – Select Physical layer

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

Testing Communications Server applications

There are different methods to test Communications Server 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 Communications Server 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.

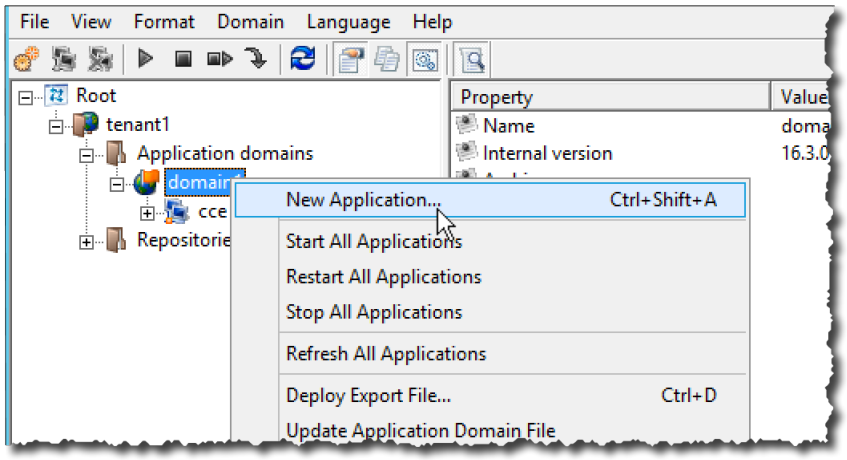
Lab: Creating an Communications Server application

Create a Communications Server application

1. *In Control Center, right-click the **domain1** application domain and select **New Application** from the pop-up menu.*

Figure 9-13:

New Application



The New Application window displays.

2. *In the New Application window enter the following information leaving the other field default values, and click the **OK** button.*
   - *Application type: **CommunicationsServer***
   - *Application name: **AdminCourse***

The new AdminCourse application is displayed in Control Center hosted by the cce host and under the domain1 application domain node.

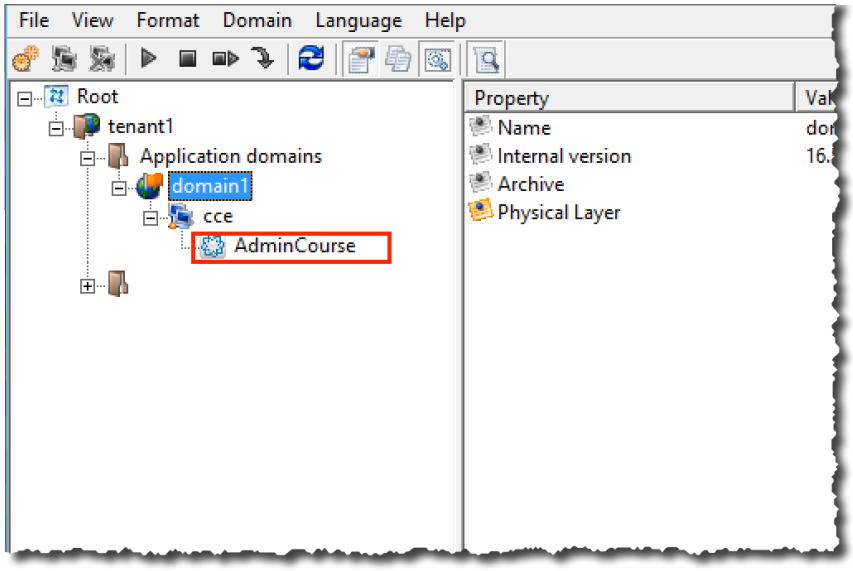


Figure 9-14:

New AdminCourse application

Typically, a developer team creates a project in Communications Builder and works with StoryTeller to create different types of documents. This projects is exported and is then deployed to a Communications Server application that will run the project.

In the next activities, you will deploy a project export file to the AdminCourse application, run it and test it.

Deploy a project export file to a Communications Server application

1. *In Control Center, right-click the **AdminCourse** application and select **Deploy Export File** from the pop-up menu.*

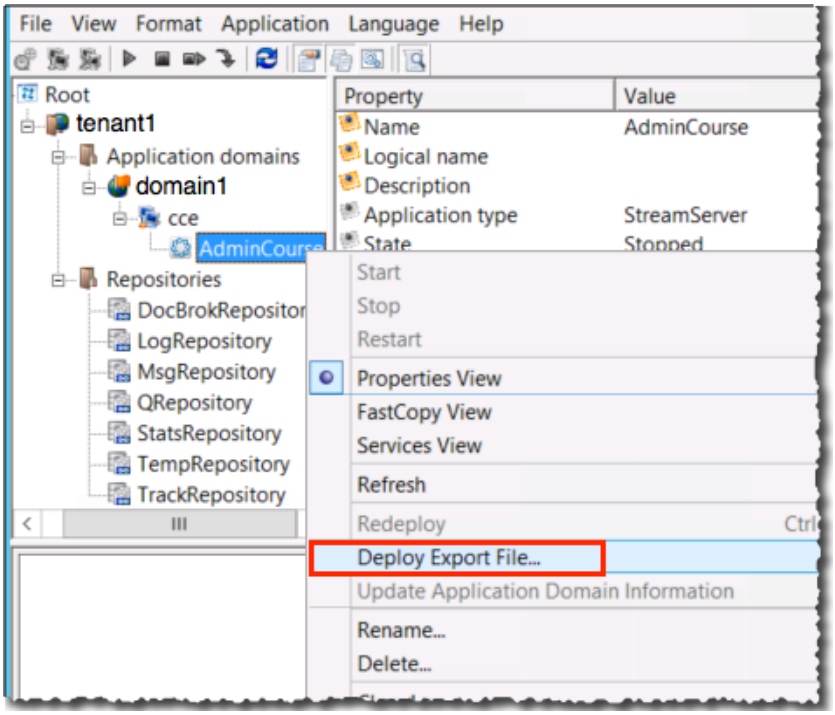


Figure 9-15:

Deploy export file

2. *In the Deploy window select **Deploy export from CAS** and click the **Browse** button.*

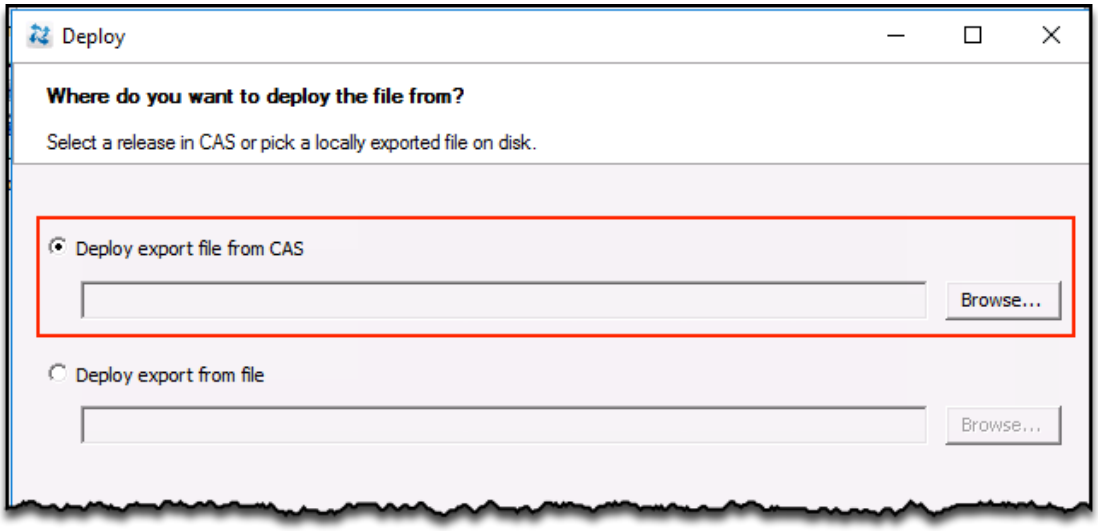


Figure 9-16: Deploy window

3. *In the Select release window select the **First Release** and click the **OK** button.*

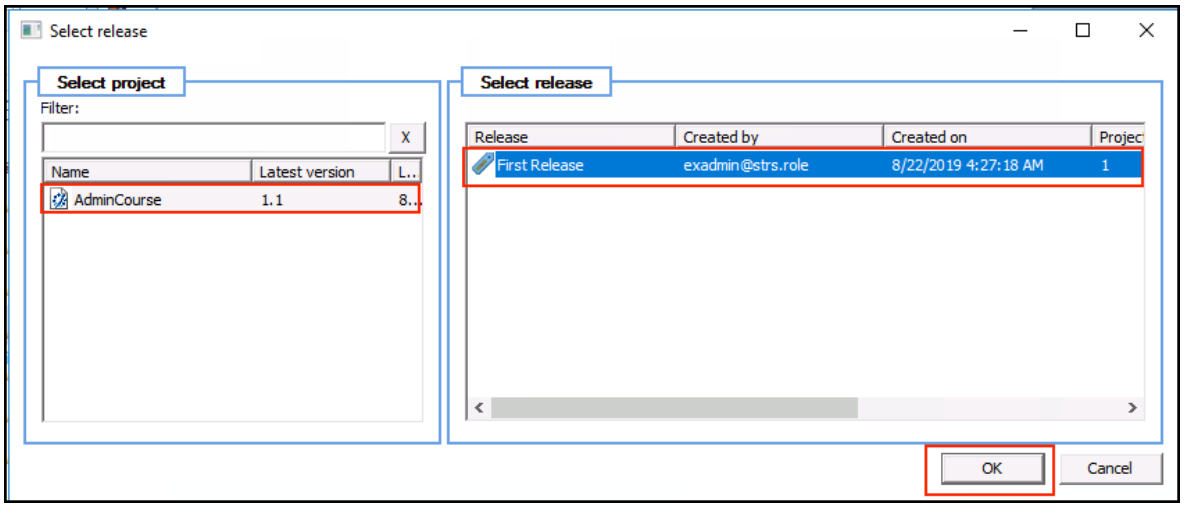


Figure 9-17: Selecting release

You are redirected back to the Deploy window.

4. *In the Deploy window click the **Next >>** button.*
5. *Select **Finish**.*

A message that the Deploy operation succeeded is displayed.

The export file is deployed to the AdminCourse application.

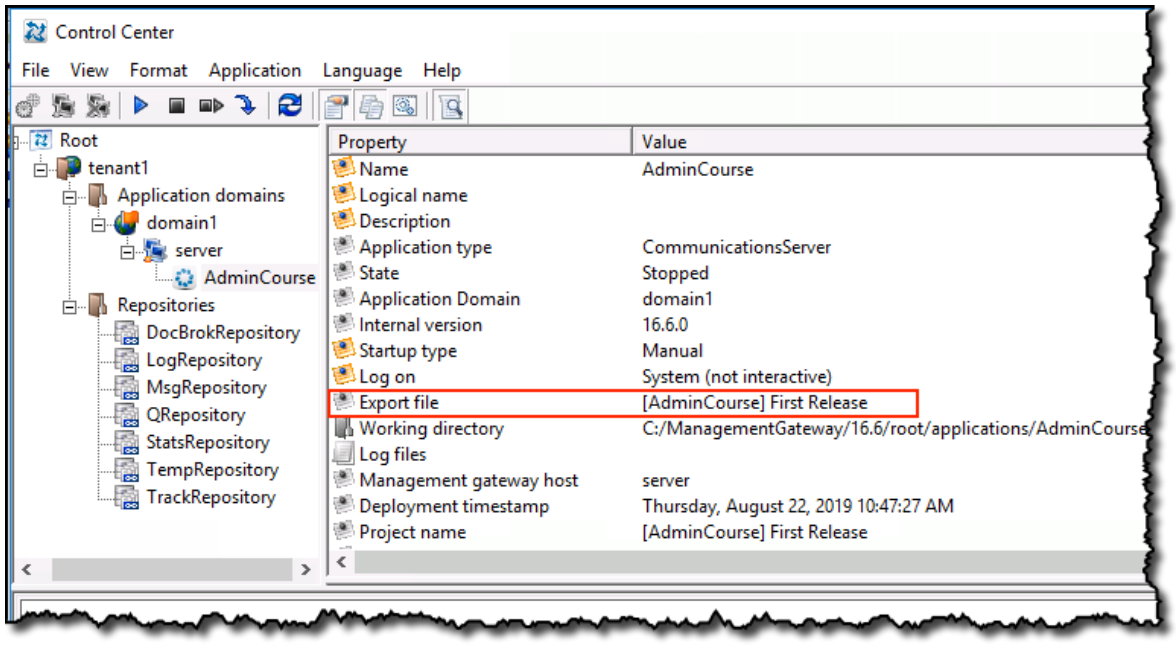


Figure 9-18: Release deployed

You can verify that the export file was deployed by checking the application working directory C:\ManagementGateway\16.6\root\applications\AdminCourse.

Start the AdminCourse application

1. *In Control Center, right-click the **AdminCourse** application and select **Start** from the pop-up menu.*

The application is started which is indicated by the green flag in the application name and also in the log panel.

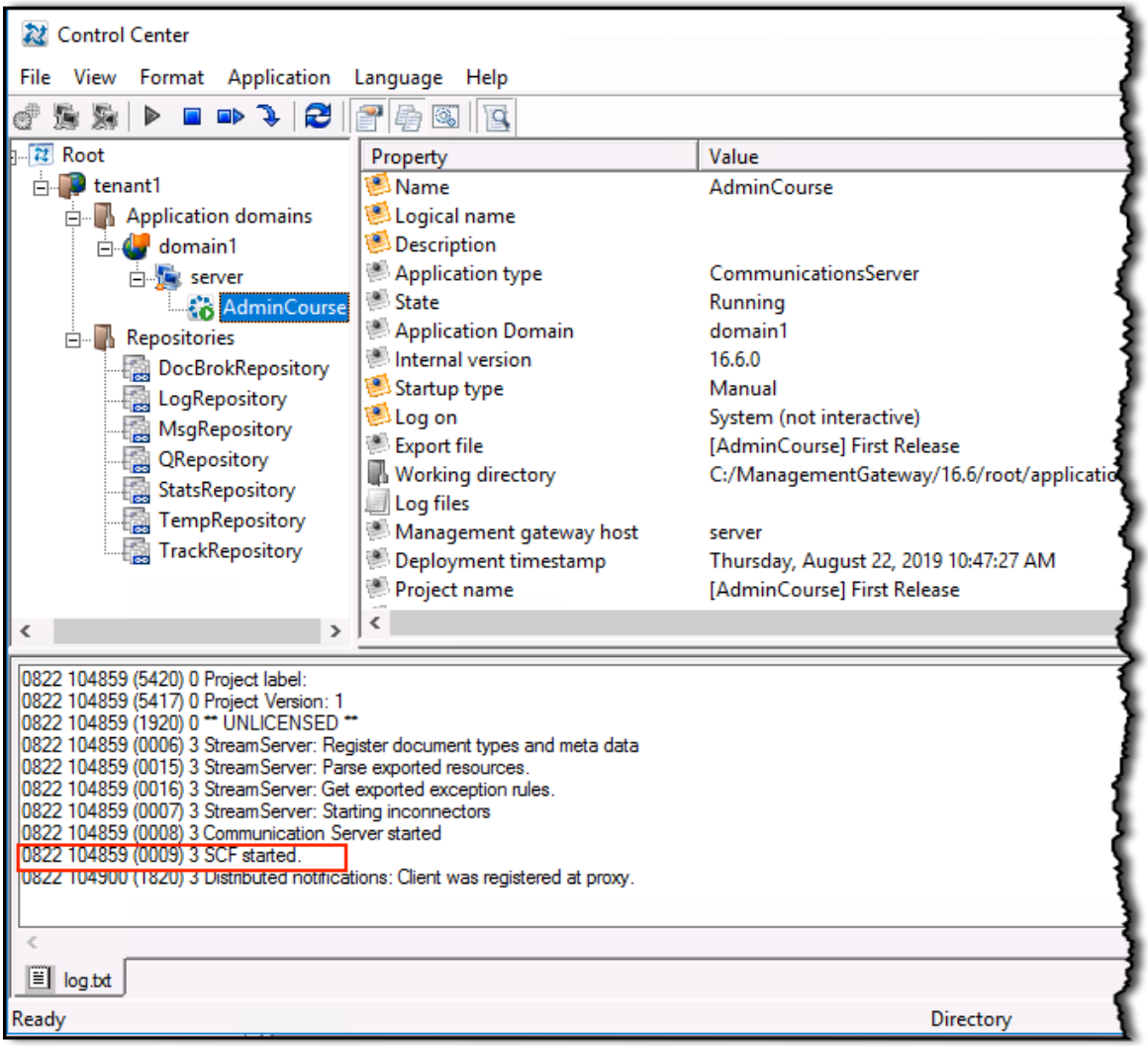


Figure 9-19: Application started

The project that was deployed to the AdminCourse application, scans the C:\DEV\IN folder for “txt” files containing invoice data and will place a formatted invoice in the C:\DEV\OUT folder.

In this next activity you will test the application.

Test the AdminCourse application

1. *Using Windows explorer copy **C:\Training\3-3730 EXS - System Administration Files\recordInvoice.txt** to the **C:\DEV\IN** folder.*

In Control Center you should see a message that the C:\DEV\IN was scanned and the found file was processed successfully and that a file called Invoice.pdf was placed in the C:\DEV\OUT folder.

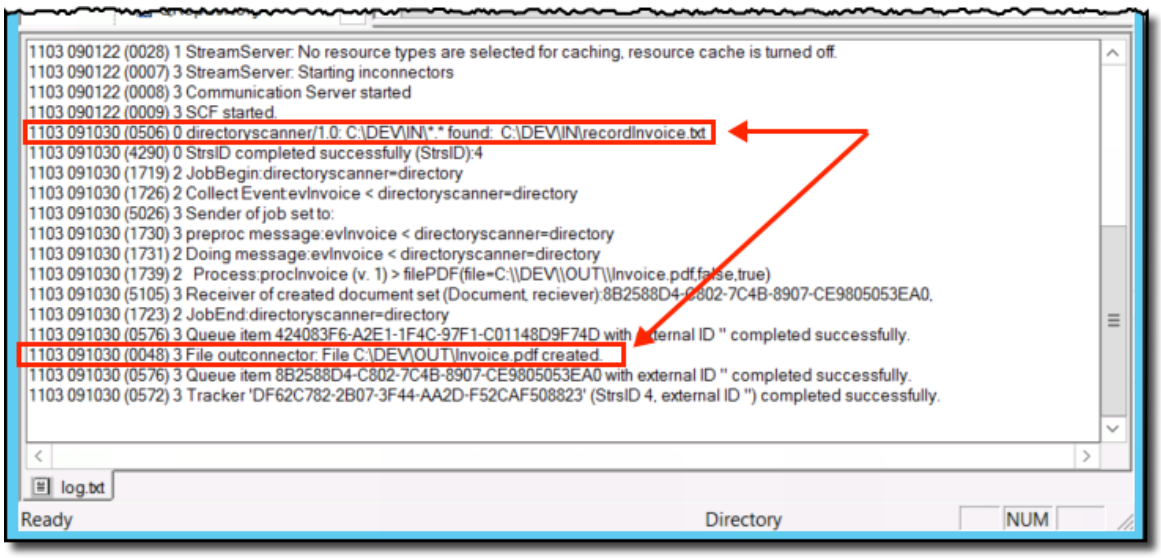


Figure 9-20: Invoice generated

2. *Open **C:\DEV\OUT\Invoice.pdf** and review its content.*
3. *Stop the **AdminCourse** application.*

Lab: Creating a Service Gateway

As indicated earlier, to use the Exstream web applications, you must add a service gateway.

A Service Gateway is an Communications Server application type and the process to create a Service Gateway instance is similar to creating any other type of Communications Server applications.

Create a Service Gateway application

1. *In Control Center, right-click the **domain1** application domain and select **New Application** from the pop-up menu.*

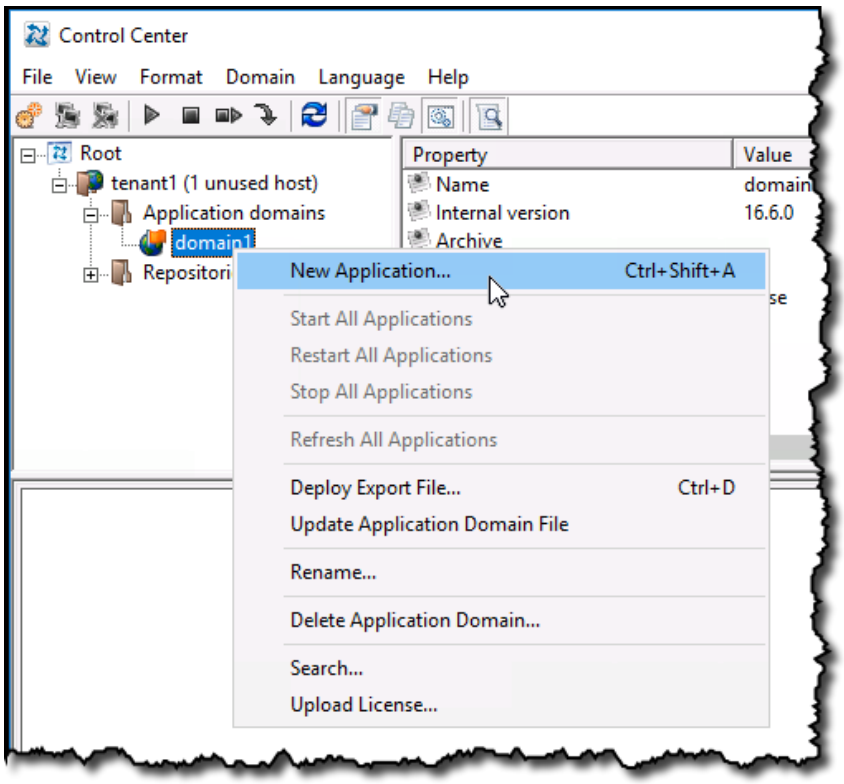


Figure 9-21:

Adding an Communications Server application

The New Application window displays.

2. *In the New Application window enter the following information leaving the other field default values, and click the **OK** button.*
   - *Application type: **Service Gateway***
   - *Application name: **SGW***
   - *Startup type: **Automatic***

Now that there is a service gateway available to the Application Domain, it is necessary to set the correct value to the "*sgw.allowedOrigins*" property so that Exstream web applications can communicate correctly with the SGW service gateway.

Later in the course you will use the Exstream web applications.

3. *Close **Control Center**.*

Enable the SGW secure communications

1. *Stop the **StreamServe Management Gateway 16.6**, and **OTDS-Tomcat** Windows services.*
2. *Copy the following file to the indicated location:*
   - *From: **C:\Training\Resources\Tomcat\training.cer***
   - *To: **C:\OpenText\Exstream\16.6\Platform\Core\16.6\bin\security\certificatestore\trusted\authorities***
3. *Copy the following file to the indicated location:*
   - *From: **C:\Training\Resources\Tomcat\training.cer***
   - *To: **C:\OpenText\Exstream\16.6\Server\global\security\certificatestore\trusted\authorities***
4. *Copy the following file to the indicated location:*
   - *From: **C:\Training\Resources\Tomcat\keystore.pfx***
   - *To: **C:\OpenText\Exstream\16.6\Server\global\security\keystore\private***
5. *Copy the following file to the indicated location:*
   - *From: **C:\Training\Resources\Tomcat\keystore.jks***
   - *To: **C:\ManagementGateway\16.6\root\applications\SGW\wd***
6. *In a text editor open **application.properties** from **C:\ManagementGateway\16.6\root\applications\SGW\wd** and add the following code at the end of the file:*

   server.ssl.key-store: keystore.jks
   server.ssl.key-store-password: opentext
   server.ssl.keyStoreType: JKS
   server.ssl.keyAlias: training

Where:

- **server.ssl.key-store** references the keystore that you placed in C:\ManagementGateway\16.6\root\applications\SGW\wd (earlier in this activity).
- **server.ssl.key-store-password**: is the keystore password.
- **server.ssl.keyStoreType**: type of keystore.
- **server.ssl.keyAlias***:* alias used for the certificate.

7. *Save and close the file.*
8. *In a text editor open **mgmgateway.xml** from **C:\OpenText\Exstream\16.6\Server\solutions\management** and add **<virtualhost name= "thecompany.com"/>** after the line containing <implementation module="mgmgateway"> (which is line 21).*

The resulting code will look like:

```
<applications>
    <application type="http://schemas.streamserve.com/uid/application/mgmgatewayapp/1.0" nam
        <configuration>
            <managementgateway xmlns="http://schemas.streamserve.com/uid/application/mgmgat
                <general tempdir="tmp"/>

                <implementation module="mgmgateway">
                <virtualhost name= "thecompany.com"/>
        <export type="application/x-application.com-resource">
          <filters>
            <filter relation="domain"      apptype="STRSCI" type="application/x-streamserve.
            <filter relation="domain"      apptype="STRSSG" type="application/x-streamserve
            <filter relation="domain"      apptype="STRSCS" type="application/x-streamserve
            <filter relation="domain"      apptype="STRSCI" type="application/x-streamserve.
            <filter relation="domain"      apptype="STRSCS" type="application/x-streamserve.
            <filter relation="domain"      apptype="STRSSG" type="application/x-streamserve.
            <filter relation="domain"      apptype="STRSCI" type="application/x-streamserve.
```

Figure 9-22: Code

9. *Save and close the file.*
10. *In a text editor open **mgw-trustedcommunicationchannel.xml** from **C:\OpenText\Exstream\16.6\Server\solutions\management** and modify line 34 as indicated below (you need to change the keystore file name and password):*

```
<file href="../../global/security/keystore/private/keystore.pfx"
password="opentext" />
```

11. *Save and close the file.*
12. *Start the **StreamServe Management Gateway 16.6** and **OTDS-Tomcat** Windows service.*
13. *Open **Control Center**.*

14. *In Control Center connect to tenant1, and select the **SGW** application and in the properties panel double-click the **Service gateway secure mode** node.*

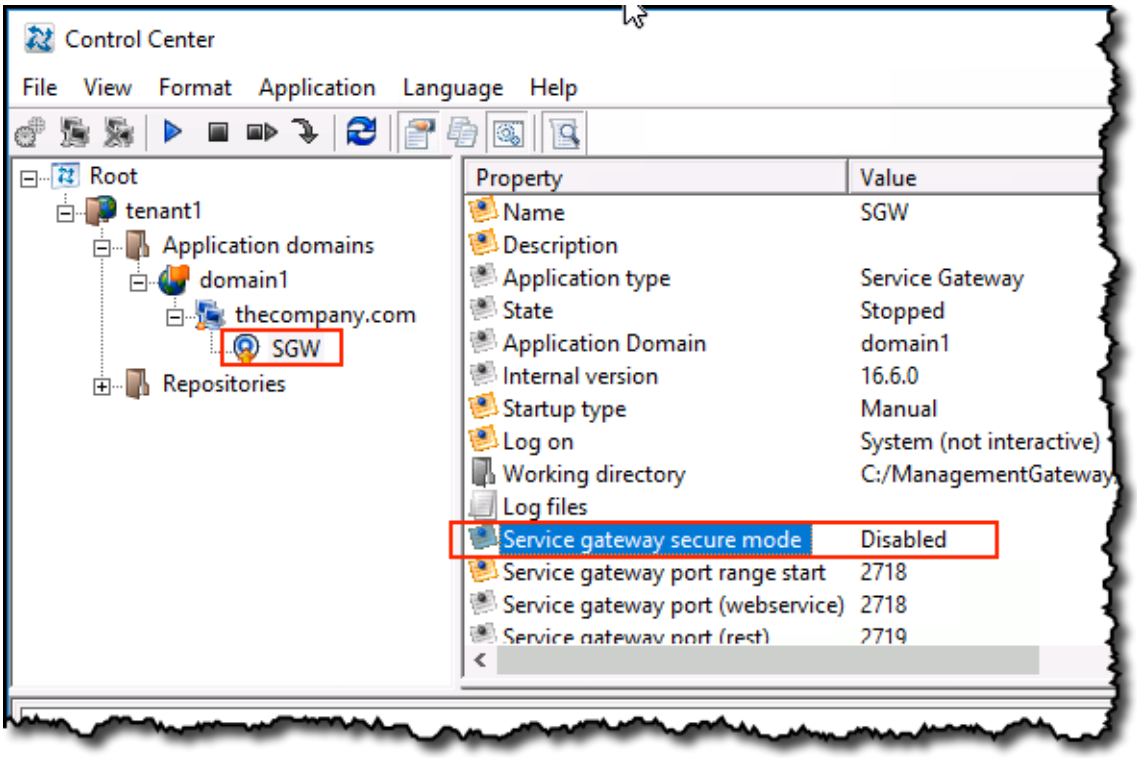


Figure 9-23: Service gateway

15. *Select **Enabled** and click **OK**.*

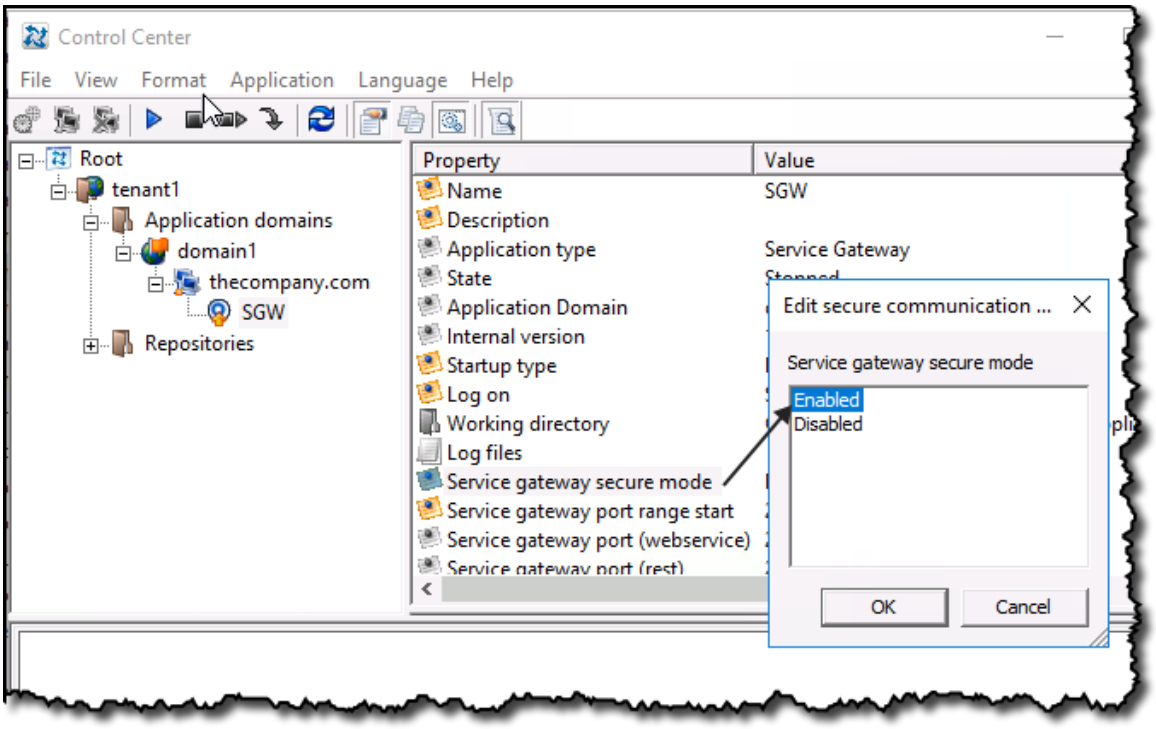


Figure 9-24: Enable the service gateway secure mode

16. *In Control Center select the **SGW** node and click the **Start** button in the toolbar so that the changes in the application.properties file are picked up.*

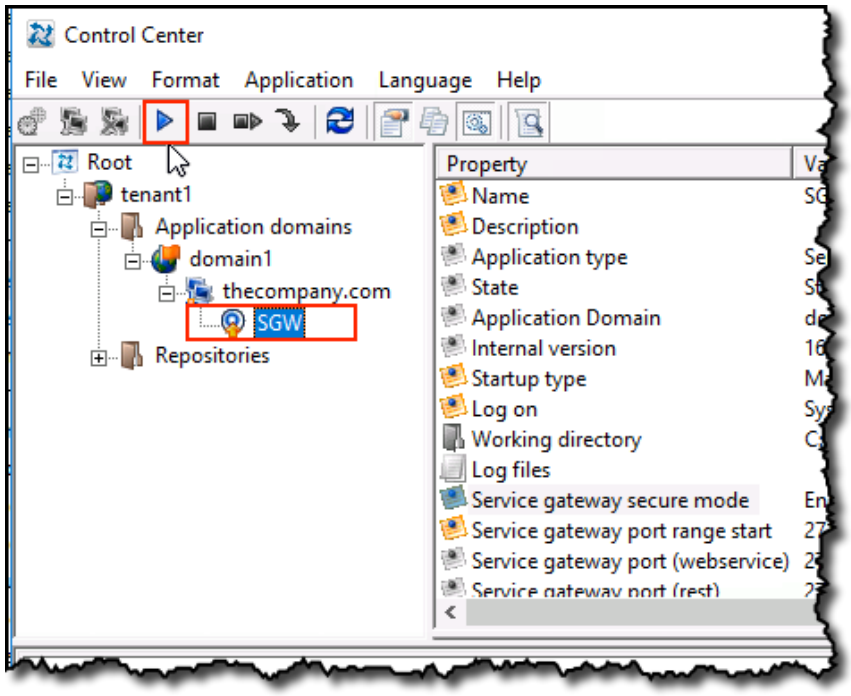


Figure 9-25:

Start service gateway

17. *Make sure no error is displayed in the logs:*

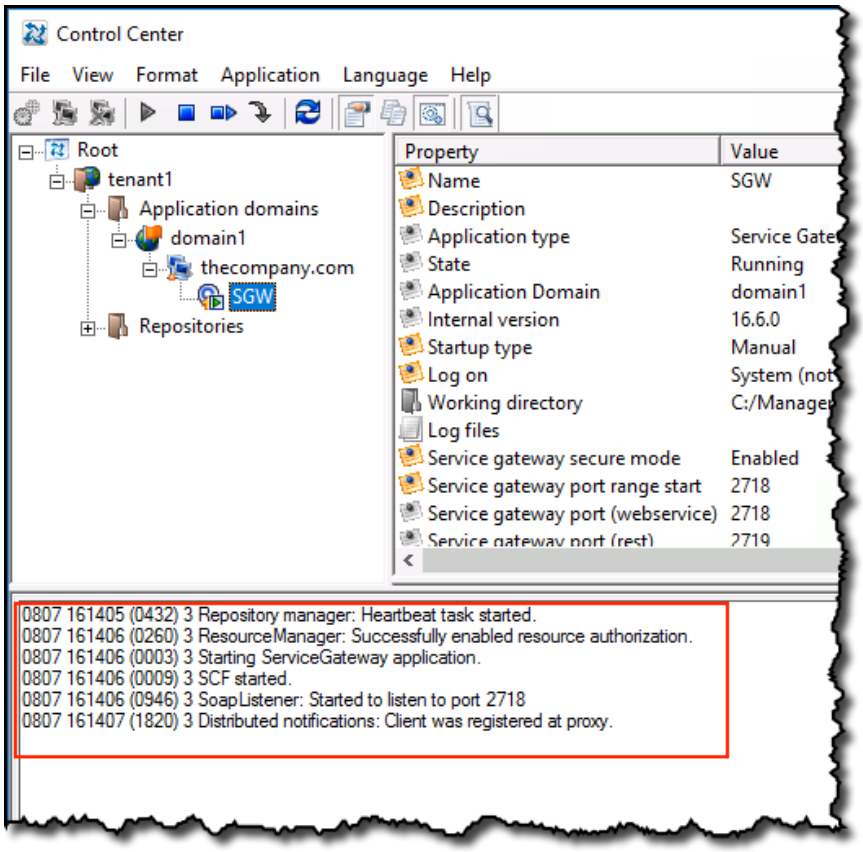


Figure 9-26:

Log

10. Exstream web applications

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

- Enumerate the Exstream web applications
- Identify the use and the URL of the different Exstream web applications
- Describe the recommendations to run the Exstream web applications
- Configure the Exstream web applications

Overview

The Exstream platform includes browser-based web applications that provide users with easy- to-use design and authoring environments that can be accessed without requiring complex installation and configuration steps on individual workstations.

In this section you will learn how to administer the web applications to:

Support for hosted mode or standalone mode

Each of the web applications included in the Exstream Runtime installer and the web application views are available as standalone applications or can be hosted in an existing business application.

The Runtime installer includes the following web applications: WorkShop, Supervisor, Communications Orchestrator, Control, StoryBoard, ReTouch, Rule Editor, Writer, and CAS Browser.

Role-based access control

Roles control which permissions people have in the web applications. For example which views, items, and tasks people can perform in the web applications. Exstream comes with a set of preconfigured roles, which are called the default roles. Administrators can change access rights and permissions for the default roles in Supervisor. It is also possible for administrators to create new roles in Supervisor.

Exstream web applications

The following is a list of the web applications that are included with the Design and Production Runtime installer:

Workshop

Workshop provides a graphical interface for managing and interacting with resources that are stored in the common asset service (CAS). The CAS is a central shared repository that provides access to and storage for the resources used in Exstream solutions. This includes image assets, Design and Production application package files, StoryBoard templates, Communications Builder projects, and PowerDocs templates.

StoryBoard

In StoryBoard, business users can enhance StoryTeller templates and personalize communications by adding text, images, and rules. Users can also use the device preview capabilities of StoryBoard to see how communications look in print and email format, and on different devices.

ReTouch Editor ReTouch is a lightweight web application that lets business users interactively edit documents generated from StoryBoard templates and also halted in reviewer.

Supervisor Supervisor is a web application that lets system administrators track and manage jobs and documents as they move through the Exstream repositories and queues during their lifecycle. The application also provides a basic statistics view where administrators can monitor job processing statistics.

Control Control provides a browser-based way for operations and system administration users to perform many common job deployment and monitoring functions. An extension to the desktop Control Center product, this thin client interface provides an easy-to-understand dashboard with status information for all applications within a domain. It also allows users to start and stop Control Center applications, as well as redeploy Communications Builder projects to existing Control Center applications configurations. Control is supported on touch mobile devices.

Migration tool You can use the Exstream Migration Tools to migrate StreamServe 5.6.2 content such as design resources and document definitions to Exstream 16.x or to transport and migrate release packages between tenants in Exstream 16.x.

Orchestrator Communications Orchestrator is a web application used to create flow models for customer communications management processes. A flow model starts with input channels through which source data enters the flow model, and ends with output channels through which the output is delivered. To this flow model you can also add nodes for data processing.

The data flow in the model is defined by drawing connections between the flow model nodes. A connection can be static, i.e. always used, and you can also set conditions to make connections available only if the conditions are fulfilled when the flow model is run.

CAS Browser CAS browser is a tool used to browse the content of CAS.

Rule Editor Rule editor is a tool used to create/edit rules that can be used with other resources to drive their behavior in a communication.

Writer Writer is a tool used to write/edit text resources or STL fragments that can be used in your communications.

The following is a list of the web applications that are installed separately:

Communications Designer Provides an intuitive web-based design environment for designers to create communications that can leverage resources that are set up in Design Manager—such as fonts, styles, data files, variables, and output queues. You can then fulfill the resulting personalized customer communications using the engine orchestration features in Communications Server.

Content Author Lets business users add content to Design and Production designs by creating and modifying communications based on Design and Production applications that have been uploaded to CAS. When you publish the modified communication, the content is included in the next engine run without requiring you to re-package your application.

Empower With Exstream Empower, organizations enable users on the front lines of customer interactions to form strong relationships with customers and prospects, providing them with a pure, thin-client interface that is easy to use and ensures compliance for critical correspondence, including letters, notices, emails, proposals, quotes and invoices.

Using a standard web browser, without any additional download or installation, Exstream Empower Editor lets users personalize communications in a controlled environment. It responds to user input by automatically populating communications with approved content. Since the editor is part of Exstream, it can access business, legal and marketing data and content from multiple sources to deliver communications to virtually any print or electronic channel.

Web applications communication

The web applications access the Exstream repositories (common asset service, messagestore repository, tracking repository, etc.) using either the REpresentational State Transfer Services API (REST API) or the Web Service API (WS API). The REST services and the web services are hosted by a service gateway. The service gateway is linked to the repositories via the domain.

When being accessed, a web client sends a request to the management gateway, asking for a service gateway endpoint. The request includes information about the tenant and the domain to be used. The management gateway checks the tenant repository for service gateways that fulfills the request, and returns the connection information for the service gateway to be used. The web client then uses the provided information to connect to the service gateway.

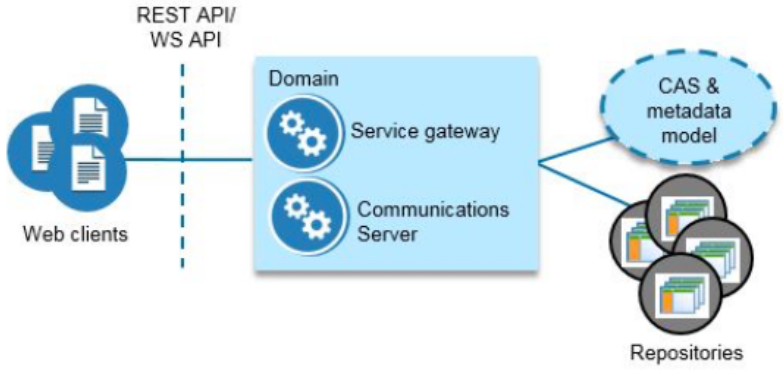


Figure 10-1:

Exstream web applications communication

Web application requirements

Before you deploy and configure the web applications, you need to review the following requirements:

Supported software

Exstream supports running WorkShop, Supervisor, Communications Orchestrator, Control, StoryBoard, ReTouch, Rule Editor, Writer, and CAS Browser on a Java application server or web server.

Host name requirements

As part of the web application configuration process, you need to specify the URLs to the web applications in the configuration files listed below. Exstream requires that you specify the host names in the URLS in the same format in all the files, and then access the web applications using the same format. For example, if you specify the host names as fully qualified domain names (FQDNs) in the configuration files, you also need to access the web applications using FQDNs.

HTTPS or HTTP requirements

The Exstream web applications use the ports listed below for communication. You need to configure all ports to use either HTTP communication or HTTPS communication.

Memory recommendations (Applies to Java application servers)

If you plan to run the web applications on a Java application server, there must be enough memory allocated for the web applications on the JVM (Java Virtual Machine).

The required amount of memory depends on the installation environment, number of deployed web applications, and the load on the Java application server. As a rule of thumb, OpenText recommends the following JVM memory setting when deploying a single web application to a dedicated Java application server: -Xmx1024m.

Accessing WorkShop

WorkShop can be run as a standalone application or embedded in a hosting application.

URL http://<host>:<port>/workshop[/hosted][/view]/tenant/<tenantKey>]/domain/<domainKey>

Where:

<host> The host name or IP address of the computer that runs the Java application server.

<port> The port the Java application server listens to. Default is 8080.

workshop The WorkShop application name. Default is workshop.

hosted Runs WorkShop in hosted mode (embedded in a hosting applications).

<view> Access one of the following views directly:

- resources – Access the Resources view.
- templates – Access the Templates view.
- services – Access the Services view.

Query strings Additional query strings can be applied to a tenant and domain information URL:

? [header=true|false][&navigation=true|false][&actions=true|false]&OTDSTicket=<ticket>][&language=<langCode>]

Where:

header=true|false Show/hide the WorkShop header bar (with the application branding and top level actions) when accessing the view directly. N/A for hosted mode.

navigation=true|false Show/hide the WorkShop navigation bar (with the view tabs) when accessing the view directly. N/A for hosted mode.

actions=true|false Show/hide the WorkShop function toolbar (with the actions for a selected resource) when accessing the view directly. N/A for hosted mode.

OTDSTicket=<ticket> Authenticates the user, if already logged in. If not specified, the user is redirected to the OTDS log in page.

language=<langCode> Optional language code (two or four letter code) that determines which locale to apply. If not set, the language code set in the browser is used. You can use either underscore (_) or dash (-) as separator in four letter codes. For example, you can use either de_DE or de-DE.

Accessing Supervisor

Supervisor can be run as a standalone application or embedded in a hosting application.

URL http://<host>:<port>/supervisor[/hosted][/view]/tenant/<tenantKey>]/domain/<domainKey>

Where:

<host> The host name or IP address of the computer that runs the Java application server.

<port> The port the Java application server listens to. Default is 8080.

supervisor The Supervisor application name. Default is supervisor.

hosted Runs Supervisor in hosted mode (embedded in a hosting applications).

<view> Access one of the following views directly:

- messages – Access the Review view.
- jobs – Access the Track view.
- produce – Access the Produce view.

Query strings Additional query strings can be applied to a tenant and domain information URL:

? [header=true|false][&navigation=true|false][&actions=true|false]&OTDSTicket=<ticket>][&language=<langCode>]

Where:

header=true|false Show/hide the Supervisor header bar (with the application branding and top level actions) when accessing the view directly. N/A for hosted mode.

navigation=true|false Show/hide the Supervisor navigation bar (with the view tabs) when accessing the view directly. N/A for hosted mode.

actions=true|false Show/hide the Supervisor function toolbar (with the actions for a selected resource) when accessing the view directly. N/A for hosted mode.

OTDSTicket=<ticket> Authenticates the user, if already logged in. If not specified, the user is redirected to the OTDS log in page.

language=<langCode> Optional language code (two or four letter code) that determines which locale to apply. If not set, the language code set in the browser is used. You can use either underscore (_) or dash (-) as separator in four letter codes. For example, you can use either de_DE or de-DE.

Accessing StoryBoard

StoryBoard can be launched as a standalone application or run embedded in a hosting application.

URL http://<host>:<port>/storyboard[/hosted]/tenant/<tenantKey>/domain/<domainKey>/<appMode>.html/<viewMode>/<themeId>/<previewFormat>/<simulationId>/language=<langCode>

Where:

<host> The host name or IP address of the computer that runs the Java application server.

<port> The port the Java application server listens to. Default is 8080.

storyboard The StoryBoard application name. Default is supervisor.

hosted Runs StoryBoard in hosted mode (embedded in a hosting application).

tenant Leave as is.

<tenantKey> The tenant name or tenant ID.

domain Leave as is.

<domainKey> The domain name or domain ID.

<appMode> Index for running StoryBoard standalone and hosted for running StoryBoard embedded in a hosting application.

<viewMode> Preview for view-only and edit for editing themes.

<themeId> The theme ID returned from the REST API.

<previewFormat> The preview format (WEB for unpaginated preview or PRINT for paginated preview).

<simulationId> The simulation ID returned from the REST API.

language=<langCode> Optional language code (two or four letter code) that determines which locale to apply. If not set, the language code set in the browser is used. You can use either underscore (_) or dash (-) as separator in four letter codes. For example, you can use either de_DE or de-DE.

Accessing ReTouch

ReTouch can be launched as a standalone application or run embedded in a hosting application.

URL http://<host>:<port>/retouch[/hosted]/tenant/<tenantKey>/domain/<domainKey>/<appMode>.html/<viewMode>/<themeId>/<previewFormat>/<simulationId>/language=<langCode>

Where:

<host> The host name or IP address of the computer that runs the Java application server.

<port> The port the Java application server listens to. Default is 8080.

retouch The Retouch application name. Default is retouch.

hosted Runs Retouch in hosted mode (embedded in a hosting applications).

tenant Leave as is.

<tenantKey> The tenant name or tenant ID.

domain Leave as is.

<domainKey> The domain name or domain ID.

<appMode> Index for running Retouch standalone and hosted for running Retouch embedded in a hosting application.

<documentTypeId> The document type ID in the Message repository.

<documentId> The document ID in the Message repository.

<previewFormat> The preview format (WEB for unpaginated preview or PRINT for paginated preview).

<simulationId> The simulation ID returned from the REST API.

language=<langCode> Optional language code (two or four letter code) that determines which locale to apply. If not set, the language code set in the browser is used. You can use either underscore (_) or dash (-) as separator in four letter codes. For example, you can use either de_DE or de-DE.

Accessing Writer

Writer can be launched as a standalone application or run embedded in a hosting application. You can use the following URL to access Writer:

URL http://<host>:<port>/writer/[hosted]/tenant/<tenantKey>/domain/<domainKey>/resource/<resourceID>?language=<langCode>

Where:

<host> The host name or IP address of the computer that runs the Java application server.

<port> The port the Java application server listens to. Default is 8080.

writer The Writer application name. Default is supervisor.

hosted Runs Writer in hosted mode (embedded in a hosting applications).

tenant Leave as is.

<tenantKey> The tenant name or tenant ID.

domain Leave as is.

<domainKey> The domain name or domain ID.

resource Leave as is.

<resourceID> The resource ID returned from the REST API.

language=<langCode> Optional language code (two or four letter code) that determines which locale to apply. If not set, the language code set in the browser is used. You can use either underscore (_) or dash (-) as separator in four letter codes. For example, you can use either de_DE or de-DE.

Accessing Rule Editor

Rule Editor can be launched as a standalone application or run embedded in a hosting application. You can use the following URL to access Rule Editor:

URL http://<host>:<port>/ruleeditor/[hosted]/tenant/<tenantKey>/domain/<domainKey>/resource/<resourceID>?language=<langCode>

Where:

<host> The host name or IP address of the computer that runs the Java application server.

<port> The port the Java application server listens to. Default is 8080.

ruleeditor The Rule Editor application name. Default is supervisor.

hosted Run Rule Editor in hosted mode (embedded in a hosting applications).

tenant Leave as is.

<tenantKey> The tenant name or tenant ID.

domain Leave as is.

<domainKey> The domain name or domain ID.

resource Leave as is.

<resourceID> The resource ID returned from the REST API.

language=<langCode> Optional language code (two or four letter code) that determines which locale to apply. If not set, the language code set in the browser is used. You can use either underscore (_) or dash (-) as separator in four letter codes. For example, you can use either de_DE or de-DE.

Configuring connection properties

To allow users to access and navigate between the WorkShop, Supervisor, StoryBoard, ReTouch, Rule Editor, Writer, and Content Author web applications, you need to configure a connection properties file with the URL to each web application.

If you want to add other web applications, like Package, CAS Browser, or Communications Analysis iHub Dashboard to the application switcher in WorkShop and Supervisor, you also need to configure URLs to those applications in the connection properties file.

To configure the connection properties file:

Create the connection properties file

You can use the cc-webapp-config.xml template when you create the configuration file. After updating the file, you can use the schema cc-webapp- config.xsd to validate the configuration file. These files are available in the following directory:
<Installation directory>\<version>\Server\solutions\management\web.

Save the connection properties file

You need to save the cc-webapp-config.xml file in a directory called config in the directory where the web applications are deployed.

Tomcat example:

C:\Program Files\Apache Software Foundation\Tomcat<version>\apache-tomcat-<version>\webapps\ROOT\config\cc-webapp-config.xml.

Service gateway properties

You must specify properties for the allowed origins for each service gateway application. To run in secure mode (using HTTPS communication), you also need to specify secure mode properties. You also have the option to specify properties to enable debug level in the service gateway log and to enable thumbnails for image resources in StoryBoard and ReTouch.

These properties are specified in the file application.properties located in the working directory of the service gateway application:

<base>\<version>\root\applications\<SGWname>\wd

where <base> is specified during the installation (default is C:\ManagementGateway) and <SGWname> is the name of the service gateway application.

The template for application.properties is located in the following directory relative to the Exstream installation directory:

solutions\management\config\<version>\STRSSG

This template is copied to the working directory when you create a new service gateway application. If you edit this file, the changes will be applied to all new service gateway applications you create, but existing service gateway applications are not affected by the changes.

Allowed origins for service gateway

You must edit the sgw.allowedOrigins property. This property specifies the origins (protocol, host and port) allowed to interact with service gateway. You must add the origins (comma separated) for the web applications, management gateway, and OTDS as values to this property. Note that you must use a colon to separate the property and value, and the values should not be within quotes.

sgw.allowedOrigins=<protocol>://<webAppHost>:<port>,...,
<protocol>://<webAppHost>

Secure mode properties

If secure mode is enabled for a service gateway, you must add secure mode properties to the applications.properties file.

server.ssl.key-store: keystore.p12
server.ssl.key-store-password: myPassword
server.ssl.keyStoreType: PKCS12
server.ssl.keyAlias: myAlias

or

server.ssl.key-store:keystore.jks
server.ssl.key-store-password:myPassword
server.ssl.keyStoreType:JKS
server.ssl.keyAlias:myAlias

Running web applications in iframes

A web application in hosted mode can be run in an iframe in a hosting application. To allow the web application to run in the hosting application, you must make sure you have the value of the xFrame.options.setHeader property set correctly in the applications.properties file:

- xFrame.options.setHeader=false – the default (if clickjacking security is not enabled).
- xFrame.options.setHeader=true – If the Java application server/web server is configured to secure against clickjacking.

Service gateway secure mode

In Control Center, you can enable secure mode for a service gateway. If secure mode is enabled, HTTPS is activated for communication via the REST API, and the web clients will use HTTPS when calling the service gateway. If secure mode is disabled, the web clients will use HTTP.

Keep in mind that if you enable secure mode for the service gateway, you also need to secure the communication for OTDS, the management gateway REST port, and for the web server or the Java application server.

The service gateway includes an integrated Java application server from Apache Tomcat, which is used internally for business logic. To configure secure mode properties for the service gateway you edit the file application.properties in the working directory of the service gateway.

1. In Control Center, select the service gateway.
2. In the Properties view, enable the Service gateway secure mode.
3. Open application.properties in a text editor.
4. Add the properties needed to enable a secure channel. For detailed information, see the Apache Tomcat or web server user documentation.
5. Save and close the file.
6. Restart the service gateway for the changes to take effect.

If you plan to use several service gateways, you can modify the application.properties file in <Exstream installation>\<version>\Server\solutions\management\config\<version>\STRSSG.

The application.properties file is copied from this location when creating new service gateways in the application domain.

Securing management gateways

This section describes how to secure management gateway SOAP communication on port 28800. This port is used for communication between the Exstream applications, OTDS, repositories, command line utilities, and desktop tools such as Communications Builder, Control Center, Describer, and Design Manager.

To secure management gateway SOAP communication, you must use a keystore file in PKCS12 format. If you have multiple management gateways in your environment, all the management gateways must use the same keystore file.

Follow these steps to secure an environment with a single management gateway:

1. Generate a keystore file on the management gateway and then configure the management gateway to use the keystore file.
2. Configure the certificates on the clients connecting to the management gateway.

About the demonstration Exstream keystore

The Exstream installation includes a demonstration keystore file named demo.pfx. This is 1024–bit RSA key file, so it cannot be used for Design Manager, which requires a higher level of encryption.

If you run the bootstrap, it automatically configures the management gateway to use one of the following keystore files:

- bootstrap.pfx – This is a self-signed 2048–bit keystore file.
- demo.pfx – This is the keystore file included in the Exstream installers. When installed with the bootstrap, it configures the following options:

"CN=#VIRTUALHOST,OU=CCEdemo,O=CCEdemo1,L=CCedemo,S=CCedemo,C=SE"," subjectaltnames": "dns:#VIRTUALHOST,dns:localhost";

You can check which keystore file you are using in the following file:

<Exstream_Runtime_ Installation_directory>\<version>\Server\solutions\management\mgw-trustedcommunicationchannel.xml

The keystore files included in the bootstrap and Exstream installers are not suitable for use in production environments.

Setting up a secure channel for the management gateway

To set up or change a secure channel for the management gateway, you must generate a private keystore file (*.pfx) that is used by the management gateway, and a file containing the public certificate (*.crt or *.cer) that is used by the applications connecting to the management gateway.

After you have generated the private keystore and certificate files, you must update the configuration file mgw-trustedcommunicationchannel.xml with the new private keystore file and corresponding password.

Private keystore location <Exstream_Runtime_ Installation_directory>\global\security \keystore\private.

Public certificate location <Exstream_Runtime_ Installation_directory>\<version>\Server \global\security\certificatestore\trusted\authorities.

Configuration file location <Exstream_Runtime_ Installation_directory>\<version>\Server \solutions\management\mgw-trustedcommunicationchannel.xml.

In the <file> tag (child of the <keystores> tag) edit the attributes:

- href – change demo.pfx to the name you specified when you generated the private keystore file.
- password – change the password to the password you specified when you generated the private keystore file.

After this, you will have to copy the public certificate file to each client running Communications Builder, Describer, Control Center, and the command line utilities.

Securing Control Center, Communications Builder, and Describer

To run these tools in secure mode, the management gateway must run in secure mode using a keystore file.

Copy the public certificate to the clients connecting to the management gateway:

- For the command line utilities, which are included in the Runtime installer, copy the public certificate to: <Exstream_Runtime_Installation_directory>\<version>\Server \global\security\certificatestore\trusted\authorities
- For Communications Builder, Describer, Control Center, and other design tools included in the Design Tools installer, copy the public certificate to: <Exstream_Design_Tools_Installation_directory>\<version> \Platform\Core\<version>\bin\security\certificatestore\trusted \authorities

Management gateway properties

You must specify the allowed origins property for each management gateway. To run in secure mode (using HTTPS communication), you also need to specify secure mode properties. You also have the option to specify properties to allow Control to run in iframes and to configure debug level in the management gateway log.

These properties are specified in the file application.properties located in the working directory of the management gateway application:

<base>\<version>\root

where <base> is specified during the installation (default is C: \ManagementGateway).

Allowed origins for management gateway

You must edit the sgw.allowedOrigins property. This property specifies the origins (protocol, host and port) allowed to interact with the management gateway. You must add the origins (comma separated) for the web applications and OTDS as values to this property:

sgw.allowedOrigins=<protocol>://<webAppHost>:<port>,...,
<protocol>://<webAppHost>

Secure mode properties

If secure mode is enabled for the Management gateway, you must add secure mode properties to the applications.properties file. Note that you must use a colon to separate the property and value, and the values should not be within quotes.

server.ssl.key-store: keystore.p12
server.ssl.key-store-password: myPassword
server.ssl.keyStoreType: PKCS12
server.ssl.keyAlias: myAlias

or

server.ssl.key-store:keystore.jks
server.ssl.key-store-password:myPassword
server.ssl.keyStoreType:JKS
server.ssl.keyAlias:myAlias

Running Control in iframes

When run in hosted mode, Control can be run in an iframe in a hosting application. To allow Control to run in the hosting application, you must make sure you have the value of the xFrame.options.setHeader property set correctly in the applications.properties file:

- xFrame.options.setHeader=false – the default (if clickjacking security is not enabled).
- xFrame.options.setHeader=true – If the Java application server/web server is configured to secure against clickjacking.

Lab: Deploying the Exstream web applications

Deploy the web applications

1. *Stop the **OTDS-Tomcat** Windows Service.*
2. *In Windows Explorer navigate to **C:\OpenText\Exstream\16.6\Server\solutions\management\web**.*
3. *Extract from each ZIP file the corresponding exploded war file to **C:\Tomcats\OTDS\webapps**.*

For example, casbrowser.zip will contain a casbrowser folder, so simply drag-and-drop it over the C:\Tomcats\OTDS\webapps in Explorer.

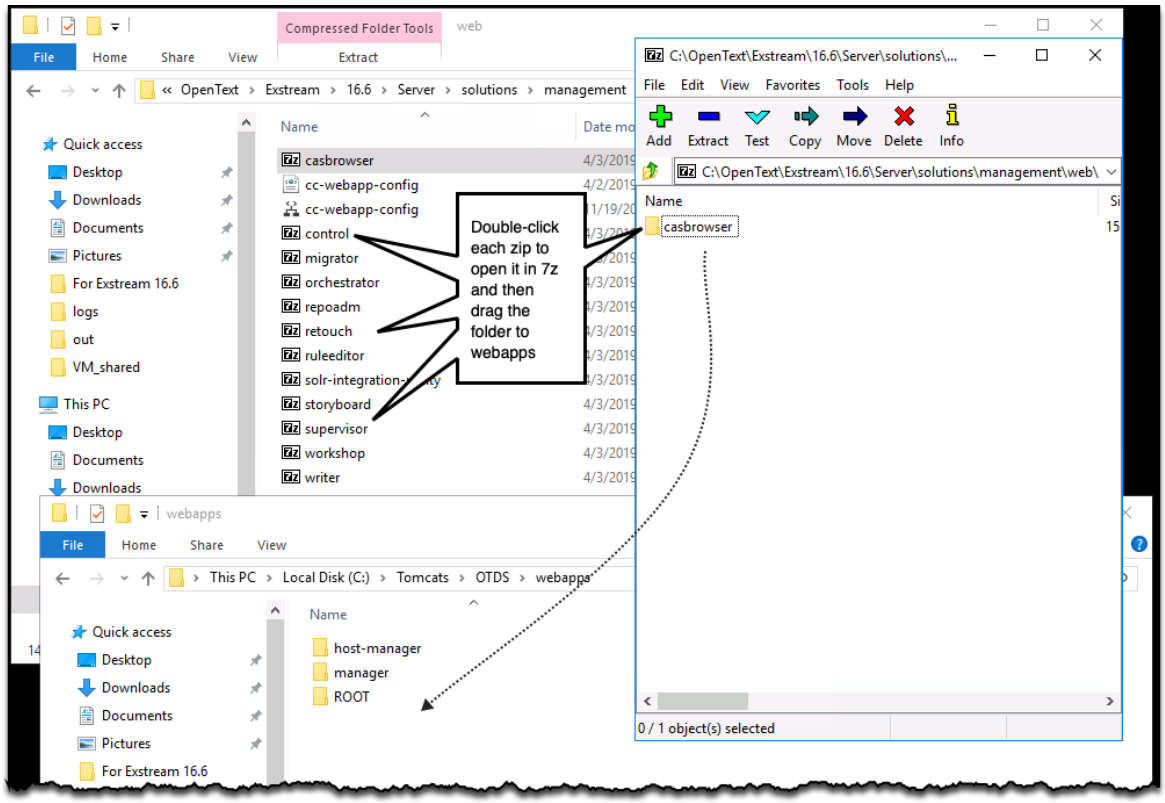


Figure 10-1: Deploying web apps

4. *Follow the previous step for the remaining zip files:*
   - ***control.zip***
   - ***migrator.zip***
   - ***orchestrator.zip***
   - ***repoadm.zip***
   - ***retouch.zip***
   - ***ruleeditor.zip***
   - ***storyboard.zip***
   - ***supervisor.zip***
   - ***workshop.zip***
   - ***writer.zip***

All of these zip files (each corresponding to a web application) were placed in C:\OpenText\Exstream\16.6\Server\solutions\management\web when installing the Exstream Runtime.

5. *Extract the content of* ***C:\Training\16.6\Software\exstream-16.6.zip*** *to* ***C:\Tomcats\OTDS\webapps****.*

C:\Training\16.6\Software\exstream-16.6.zip has the expanded war folder for Content Author and Communications Designer.

Lab: Configuring the Management and Service Gateways

Secure the Management Gateway

1. *Stop the **StreamServe Management Gateway 16.6** Windows service.*
2. *Copy the **keystore.jks** file to the indicated location:*
   - *From: **C:\Training\Resources\Tomcat***
   - *To: **C:\ManagementGateway\16.6\root***
3. *In a text editor open **application.properties** from **C:\ManagementGateway\16.6\root** and add the following code at the end of the file:*

   server.ssl.key-store: keystore.jks
   server.ssl.key-store-password: opentext
   server.ssl.keyStoreType: JKS
   server.ssl.keyAlias: training

   Where:

   - **server.ssl.key-store** references the keystore that you placed in C:\ManagementGateway\16.6\root (earlier in this activity).
   - **server.ssl.key-store-password**: is the keystore password.
   - **server.ssl.keyStoreType**: type of keystore.
   - **server.ssl.keyAlias***:* alias used for the certificate.
4. *Set the **sgw.allowedOrigins** property value to **https://thecompany.com:8443**.*
5. *Set the **xFrame.options.allowFrom** property value to **https://thecompany.com:8443**.*
6. *Save and close the file.*
7. *In a text editor open **mgmgateway.xml** from **C:\OpenText\Exstream\16.6\Server\solutions\management** and make sure that **<virtualhost name= "thecompany.com"/>** displays after the line containing <implementation module="mgmgateway"> (which is line 21).*

The resulting code will look like:

Figure 10-2: Code

8. *Save and close the file.*
9. *In a text editor open* ***mgw-trustedcommunicationchannel.xml*** *from* ***C:\OpenText\Exstream\16.6\Server\solutions\management*** *and make sure that line 34 looks as indicated below:*

```
<file href="../../global/security/keystore/private/keystore.pfx"
password="opentext" />
```

10. *Save and close the file.*

Secure the SGW

1. *In a text editor open* ***application.properties*** *from* ***C:\ManagementGateway\16.6\root\applications\SGW\wd*** *and set the* ***sgw.allowedOrigins*** *property value to* ***https://thecompany.com:8443****.*
2. *Set the* ***xFrame.options.allowFrom*** *property value to* ***https://thecompany.com:8443****.*
3. *Save and close the file.*
4. *Start the* ***StreamServe Management Gateway 16.6****, and* ***OTDS-Tomcat*** *Windows services.*

5. *In Control Center select the **SGW** node and click the **Start** button in the toolbar so that the changes in the application.properties file are picked up.*

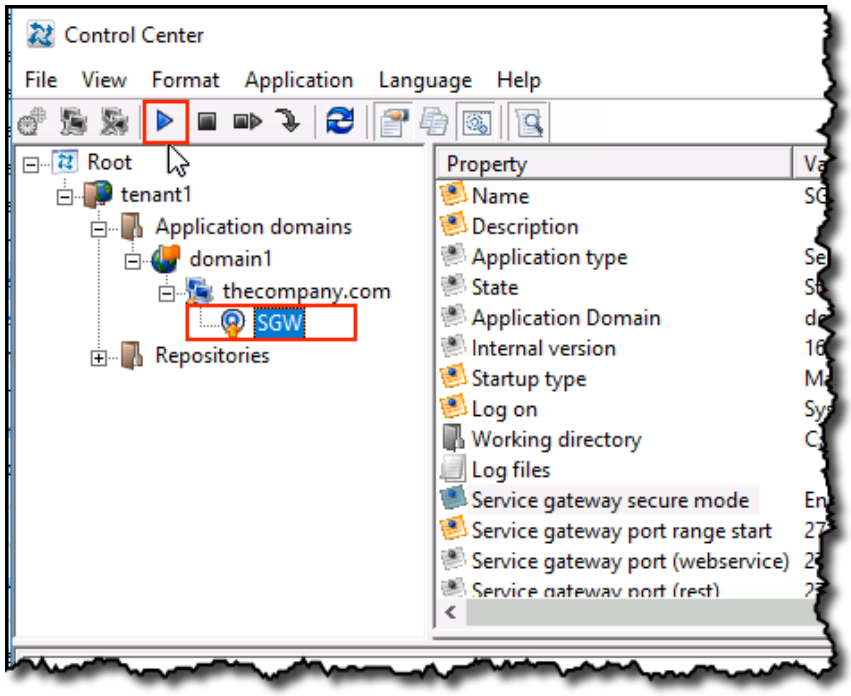


Figure 10-3:

Start service gateway

Make sure no error is displayed in the logs:

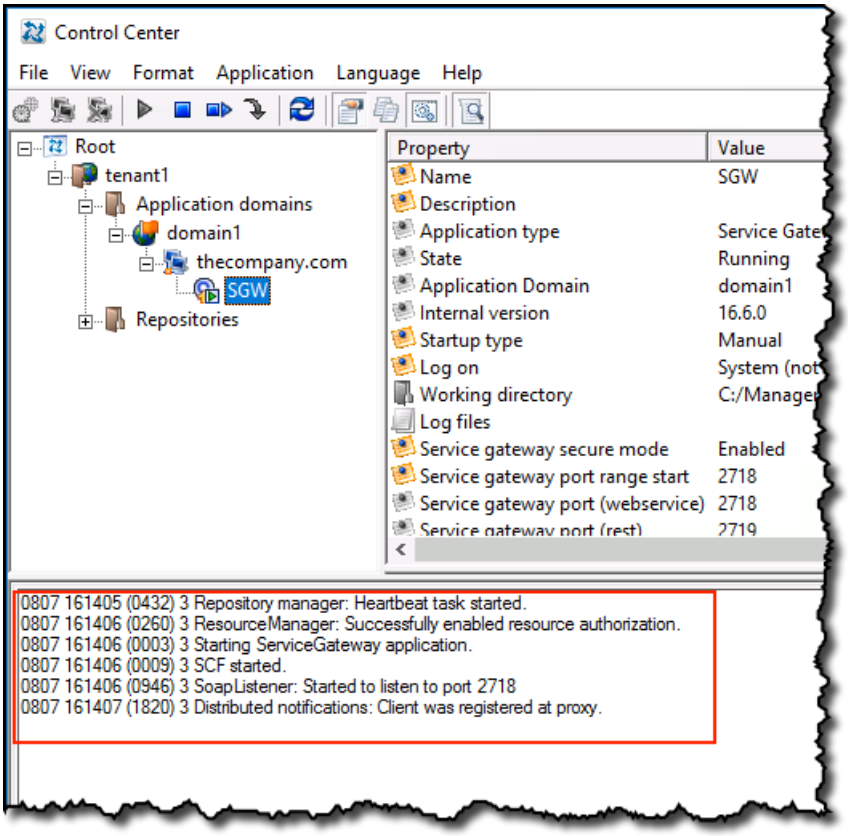


Figure 10-4:

Log

6. *In Windows Explorer navigate to **C:\OpenText\Exstream\16.6\Server\solutions\management\web**, copy **cc-webapp-config.xml** and **cc-webapp-config.xsd** and paste both files into the **C:\Tomcats\OTDS\conf** folder.*
7. *In a text editor open **C:\Tomcats\OTDS\conf\cc-webapp-config.xml**.*
8. *Replace all the **http://localhost:8080** and **https://localhost:8080** occurrences with **https://thecompany.com:8443**.*
9. *Replace all the **http://localhost:28801** with **https://thecompany.com:28801**.*
10. *Click **OK**.*

If you will be using Content Author or Communications Designer it is necessary to disable cascading approvals to make sure that approval workflow for your communications works correctly.

Disable cascading approvals

1. *In a text editor open **mgmgateway.xml** from **C:\OpenText\Exstream\16.6\Server\solutions\management**.*
2. *Set the cascadeapprovestate to false in line 119:*

   <cascadeapprovestate defaultEnabled="false">

3. *Save the file and close the text editor.*
4. *Start the **OTDS-Tomcat** and **StreamServer Management Gateway 16.6** Windows services.*

Test access to the web applications

1. *Make sure that the **SGW** application is running in Control Center.*
2. *In Chrome navigate to WorkShop using the following information:*
   - *URL: **https://thecompany.com:8443/workshop/#/tenant/tenant1/domain/domain1***
   - *User name: **exadmin***
   - *Password: **opentext***

You are logged in to WorkShop.

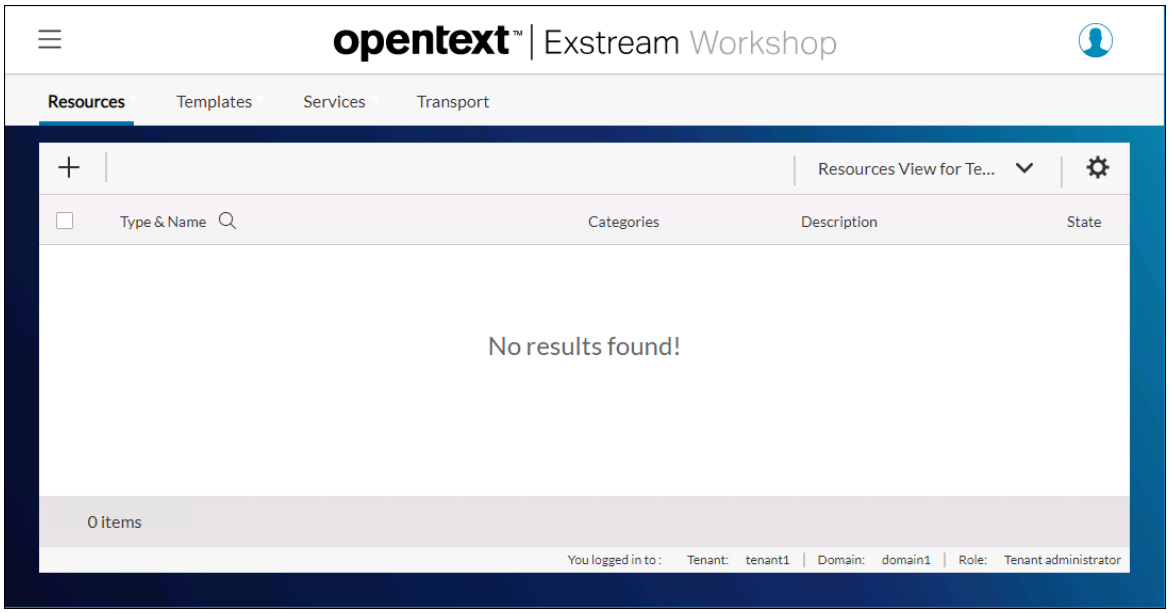


Figure 10-5: WorkShop

3. *In Chrome navigate to Content Author using the following information:*
   - *URL:* ***https://thecompany.com:8443/exstream/#/tenant/tenant1/domain/domain1***
   - *User name:* ***exadmin***
   - *Password:* ***opentext***

You are logged in to Content Author.

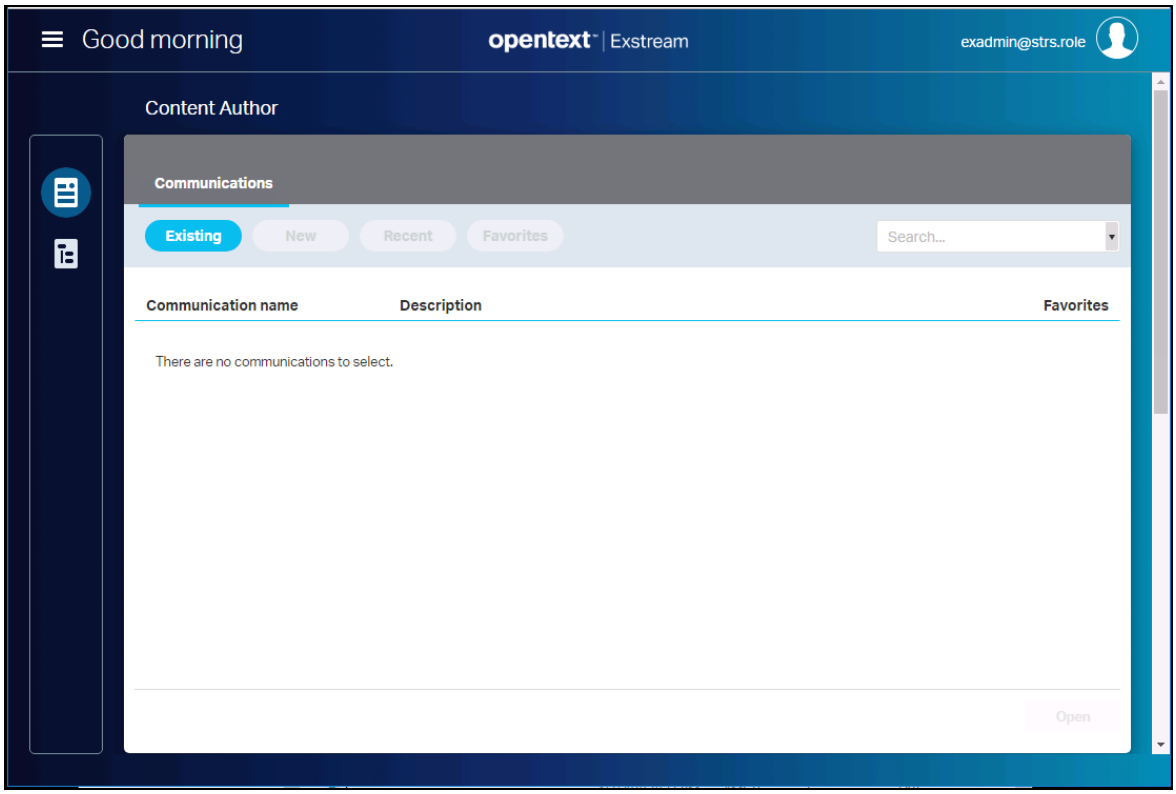


Figure 10-6: Content Author

11. Profiling

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

- Define profiling
- Identify the processes that can be monitored by Profiler
- Enable the Profiler
- Configure the Profiler
- Define and apply event filters
- Analyze profiler output

About profiling

When profiling, you investigate how the Exstream 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 Communications Builder 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.

Exstream Profiler

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 Exstream Profiler for the Exstream related parts. Profiler 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 Exstream repositories.
- Web service requests to and from the service gateway.
- Cache service operations (mainly for ADEP Designer Processes).

Enabling Profiler

When you start an application (Exstream Communications Server, Task Scheduler, or service gateway), the Profiler service is loaded but no performance data is collected. To start collecting performance data, you must enable Profiler in Control Center or via the command line.

By default performance data is created in a file called profiler.data in the working directory.

Using Profiler to collect performance data affects Exstream processing performance. In a production environment, you should only enable Profiler and collect performance data when required.

Profiler can be enabled:

- In Control Center
- Via command line

Configuring Profiler

You can change the default configuration for Profiler 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 the type of information the Profiler provider logs for each event.
- Configure filters that control which Exstream components the Profiler logs events for.

Common or application-specific file

You can either:

- Update the common profilerservice.xml file, applicable for all profile enabled applications. On Windows, this file is found under <InstallationDirectory>\Platform\Core\<version>\bin.
- Copy the file to the working directory of a specific application and apply updates to this application only (recommended).

Configuring a Profile provider

The Profile provider provides an interface between the Profiler service and the tool in which you intend to analyze the profile data.

By changing properties in the profilerservice.xml, you can configure the behavior of the provider, such as where the output file is generated and the type of information that is logged for each event.

Profile provider properties

Property	Description
source	The name and location of the file where the performance data is saved. By default, a profiler.data file is generated in the working directory of each profiler-enabled application.
codepage	The format of the profile data, which must be UTF–8.
delete	Determines whether Profiler overwrites any existing profile data, for example in profiler.data, when Profiler is enabled. • TRUE – The file and any existing data is deleted and a new file is created. • FALSE – The new profile data is appended to the existing file.
buffersize	The size in KBs of the front and back buffers of the Profile provider.
messageformat	The event properties that Profiler logs. The more events that are logged, the more Profiler affects the performance of the Exstream Communications Server application.

Specifying which event properties are logged

By default, Profiler only logs some of the event properties and uses # as the delimiter. You can use the event UID (%1) to look up the other event properties when you import the profile data to the database.

You can include other event properties in the profile data and specify the delimiter which separates the events by configuring the messageformat attribute. The delimiter must match the tool in which you will analyze the profile output.

It is also possible to only log the event UID and look up all other properties when you import the data to the database. Logging only one event reduces the size of the profile output and improves Profiler performance.

Event properties

Property	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).
%9	The sequence number of the event, which shows the order the events are logged in.
%10	The ISO timestamp of the event.

Profile provider configuration example

In this example, a profiler.data file is generated in the working directory. Only one event property is logged, which is the event ID (%1).

```
<provider value="http://schemas.Communications Server.com/uid/component/
fileprofilerprovider/2.0">
   <configuration>
         <fileprofilerprovider xmlns="http://
         schemas.Communications Server.com/uid/component/
fileprofilerprovider/
         2.0">
               <provider source="profiler.data" codepage="utf8"
               delete="yes" messageformat="%1"> </provider>
         </fileprofilerprovider>
   </configuration>
</provider>
```

Applying event filters

By default all measured profile events are presented in the profile data, which results in comprehensive output. 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

Filter configuration example The filter below excludes information in the profile output about detailed events relating to StoryTeller processing.

```
<filters>
   <filter type="http://schemas.Communications Server.com/uid/resource/
   profilerservice/namespacefilter/1.0">
         <configuration>
               <namespacefilters xmlns=
               "http://schemas.Communications Server.com/
               uid/resource/profilerservice/namespacefilter/1.0">
                  <namespacefilter type="exclude">
                     Communications
Server.notification.storyteller.profiler.
                     profilerevent.timer.detailed</namespacefilter>
                  <namespacefilter type="exclude">
                     Communications
Server.notification.storyteller.profiler.
                     profilerevent.value</namespacefilter>
                  </namespacefilters>
         </configuration>
   </filter>
</filters>
```

Flushing Profiler data to the output file

By default profile data is flushed or written into the output file (for example, profiler.data) every 5 minutes or when the memory buffer of the Profile provider is full.

You can change the interval at which the profile data is flushed in the flushschedule attribute of the profilerservice element in profilerservice.xml. If you leave this string blank "", profile data is only flushed into the output file when the memory buffer is full.

Tips

- Flushing too often affects the Profiler performance.
- Because the default Profile provider has a large memory buffer, we recommend you use a flushing interval to ensure data is written to the output file regularly.

Scheduling the Profiler counter events

The counter events are reported every 60 seconds by default. You can change this interval in the resportschedule attribute of the counters element in profilerservice.xml. If you do not want to report the counter events, leave this string empty "". If you comment out this section, the counter events are reported periodically.

Analyzing profiler output

You must stop the Exstream Communications Server application before you begin analyzing the profiler output data. If you did not change the default settings, the performance data is created in a file called profiler.data in the working directory.

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.

Lab: Profiler Services

Enable the Profiler output

1. *In Control Center, click the **AdminCourse** application node.*
2. *In the Properties panel scroll down and double-click the **Profiler** property.*

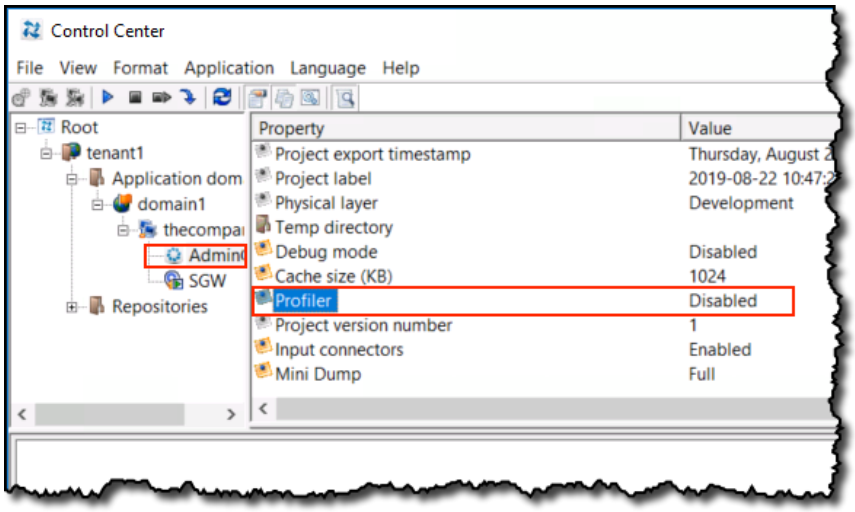


Figure 11-1:

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.*

Upon startup a profiler.data file is added to the application working directory (C:\ManagementGateway\16.6\root\applications\AdminCourse\Development).

5. *Open **profiler.data** with **TexPad** (make sure to open TextPad as an administrator user).*
6. *Review the content of the file, note the information that is logged.*

You may need to reload Control Center for the profiler.data tab to display.

Next you will configure the format of the messages logged by Profiler. (Do not close TextPad).

Set the Message Format for the Profile Provider

1. *Open windows explorer and navigate to* ***C:\OpenText\Exstream\16.6\Platform\Core\16.6\bin****.*
2. *Copy* ***profilerservice.xml*** *to* ***C:\ManagementGateway\16.6\root\applications\AdminCourse\Development****.*
3. *Open* ***profilerservice.xml*** *with TextPad.*
4. *Locate the line* ***<provider source="profiler.data" codepage="utf8" delete="yes">*** *(around line 38).*
5. *Add* ***messageformat="%2#%4#%5#%6#%7#%8"*** *at the end of the tag right before the closing ">":* ***<provider source="profiler.data" codepage="utf8" delete="yes" messageformat="%2#%4#%5#%6#%7#%8">****.*
6. *Save the* ***profilerservice.xml*** *file and do not close the file in TextPad.*
7. *Stop the* ***AdminCourse*** *application by clicking the* ***Stop*** *button in the Control Center toolbar.*
8. *Once the* ***AdminCourse*** *application is stopped, go back to TextPad and make sure the* ***profiler.data*** *tab is selected. (If prompted to reload the file, click* ***Yes****.)*
9. *Select all the content of the* ***profiler.data file****, delete it and save the file. (This way, next time the file is reloaded, you will only see the latest (Profiler logged messages).*

The content of the profiler.data tab is cleared.

10. *In Control Center, making sure the* ***AdminCourse*** *application is selected, click the* ***Start*** *button.*
11. *Back in TextPad, make sure the* ***profiler.data*** *tab is selected and select* ***Yes*** *when prompted to reload the file.*

Notice that the messages are being displayed using the format indicated (messageformat="%2#%4#%5#%6#%7#%8". See “Event properties” on page 4 for the description of each event property).

Apply Filters to the Profile Output

1. *In TextPad select the tab for the* ***profilerservice.xml*** *file.*

2. *Locate the following block (it is line 53 in the file):*

```
<!--<filters>
<filter type="http://schemas.streamserve.com/uid/resource/profilerservi
         ce/namespacefilter/1.0">
<configuration>
<namespacefilters xmlns="http://schemas.streamserve.com/uid/resource/
profilerservice/namespacefilter/1.0">
<namespacefilter type="include">streamserve.notification.profiler.
profilerevent.cacheservice</namespacefilter>
              <namespacefilter type="exclude">*</namespacefilter>
                           </namespacefilters>
                         </configuration>
                       </filter>
                     </filters>
```

3. *Uncomment the block by removing the "**<!--**" and "**-->**" in lines 53 and 62 respectively to activate this filter.*

This filter configures the Profiler output to display only information about the cache connection provider.

4. *Save the **profilerservice.xml** file.*
5. *Stop the **AdminCourse** application by clicking the **Stop** button in the Control Center toolbar.*
6. *Once the **AdminCourse** application is stopped, go back to TextPad and make sure the **profiler.data** tab is selected. (when prompted to reload the file, click **Yes**).*
7. *Select all the content of the **profiler.data file**, delete it and save the file. (This way, next time the file is reloaded, you will only see the latest Profiler logged messages.)*

The content of the profiler.data tab is cleared.

8. *Making sure the **AdminCourse** application is selected, click the **Start** button in the Control Center toolbar.*
9. *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.

Lab: 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 **Programs > Microsoft SQL Server Tools 17 > SQL Server 2017 Management Studio**.*

Microsoft SQL Server Management Studio opens presenting the Connect to Server window opens.

2. *In the Connect to Server window make sure the following values are set and click the **Connect** button.*
   - *Service type: **Database Engine***
   - *Server name: **server***
   - *Authentication: **SQL Authentication***
   - *Login: **sa***
   - *Password: **opentext***
   - *Remember password: **selected***
3. *Right-click the **Databases** node and select **New Database** in the pop-up menu.*

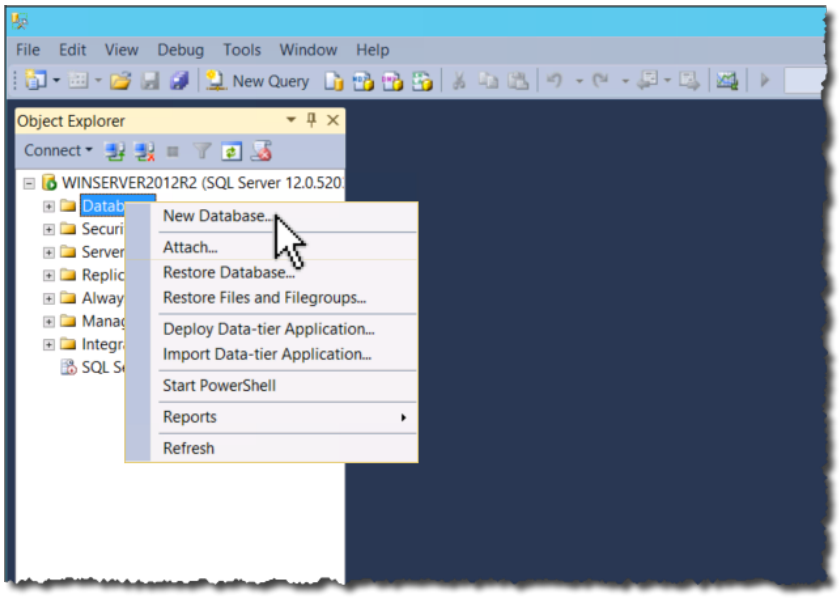


Figure 11-2:

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.

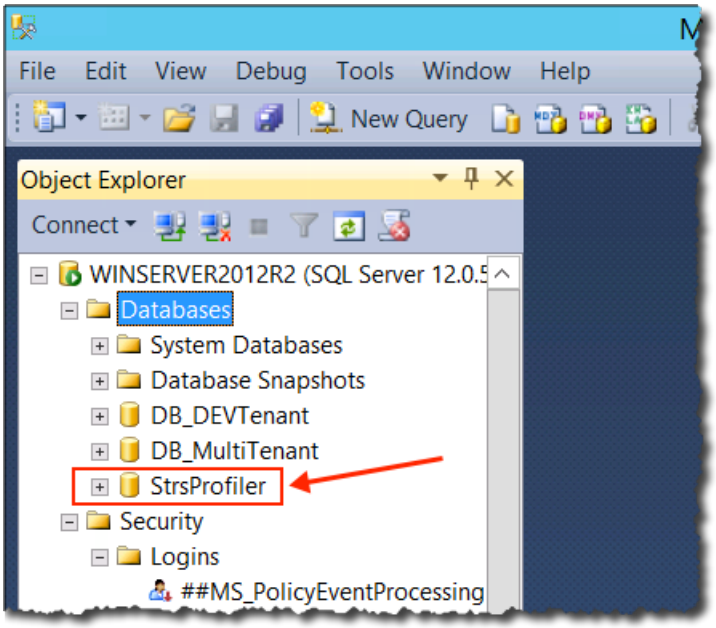


Figure 11-3:

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:\Training\3-3730 EXS - System Administration Files\Scripts**, select **createTables.sql** and click the **Open** button.*

The createTables.sql script opens in the right panel:

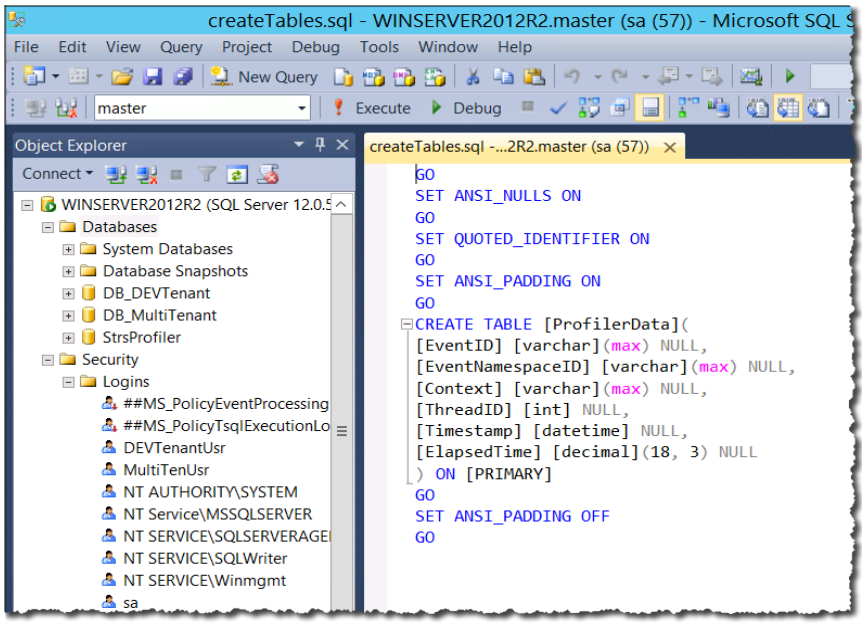


Figure 11-4:

Script loaded

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.*

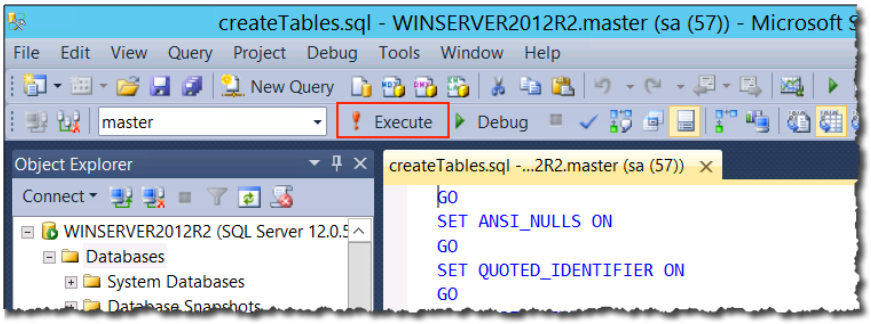


Figure 11-5:

Execute Script

9. *Make sure 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:\Training\3-3730 EXS - System Administration Files\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 Control Center stop the **AdminCourse** application.*
6. *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:\Training\3-3730 EXS - System Administration Files\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.

12. Monitoring

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

- Describe the Control Center logs
- Identify the different types of logs
- Identify the location of the logs
- Describe the structure of logged messages
- Identify the log levels
- Describe how to debug applications
- Describe database logging
- Configure Java Notifications
- Define task scheduler
- Describe the settings to run system command tasks
- Describe the configuration of the task scheduler application
- Delete expired content
- Identify the recommendations for scheduling the deletion tasks

About Control Center logs

From Control Center you can view the log for each application and change the log levels. You can also enable database logging for each application and store the log information in a logging repository.

Log files and paths

Each Exstream Communications Server, service gateway, and Task Scheduler application generates log files, which are found in the working directory of the application.

- Boot log – which contains early startup messages. File name <application wd>/strsboot.log
- Application log – which contains runtime log messages. File name <application wd>/log.txt

In addition to the boot and application logs, the service gateway has a Java log file (servicegateway_rest.log).

Control Center log view

The log view in Control Center contains the same information as the corresponding platform.txt log file for Communications Server, task scheduler, and service gateway applications. If you want to, you can clear the information displayed for the application (right-click the log view and select Clear). This action clears the log displayed in Control Center, but does not affect the corresponding log file.

Exstream Communications Server application log files

Communications Server applications generate log files with information about start up, run and stop. Each physical layer has a separate log file, located in the corresponding working directory. The default name of the log file is log.txt. The file name can be changed by project developers in Communications Builder. The location of the log files is:

<InstallDrive>\ManagementGateway\16.6.0\root\applications\<application_name>\<platform_name>.

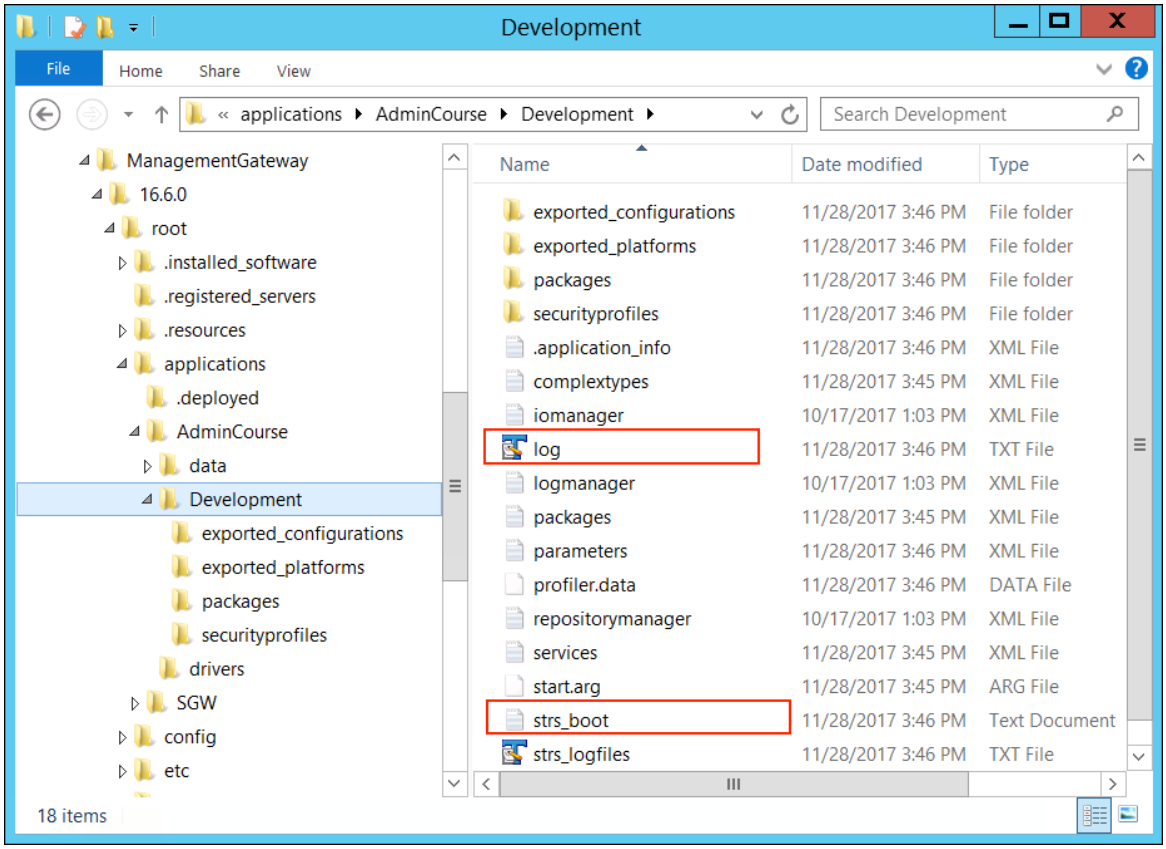


Figure 12-1: Application log files

Management Gateway log file

The management gateway generates log files with information about the connection to Control Center. The default location is <InstallDrive>\ManagementGateway\16.6.0\root.

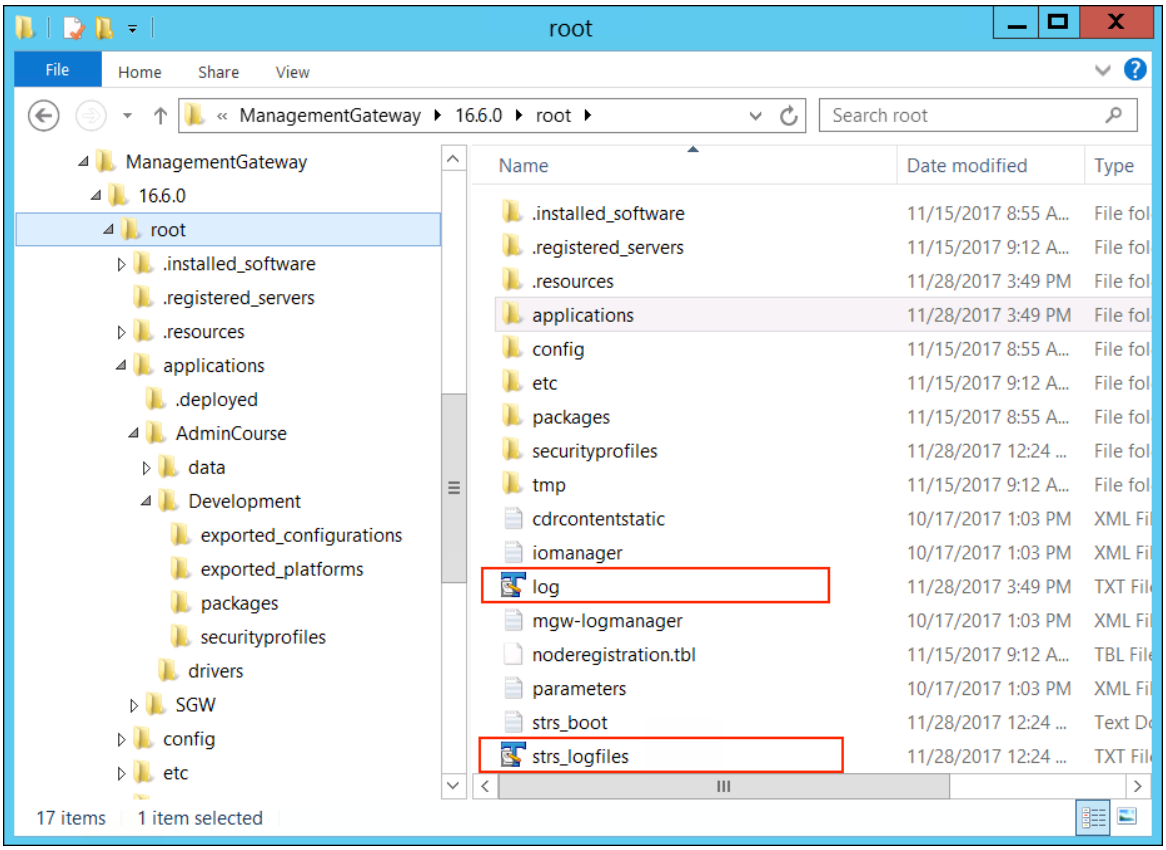


Figure 12-2: Management Gateway log files

Service Gateway application log files

The service gateway generates log messages with information about the connection to the Exstream web applications. In addition to the boot and application logs, the service gateway has a Java log file (servicegateway_rest.log).

The location of the service gateway is <InstallDrive>\ManagementGateway\16.6.0\root\applications\SGW\wd.

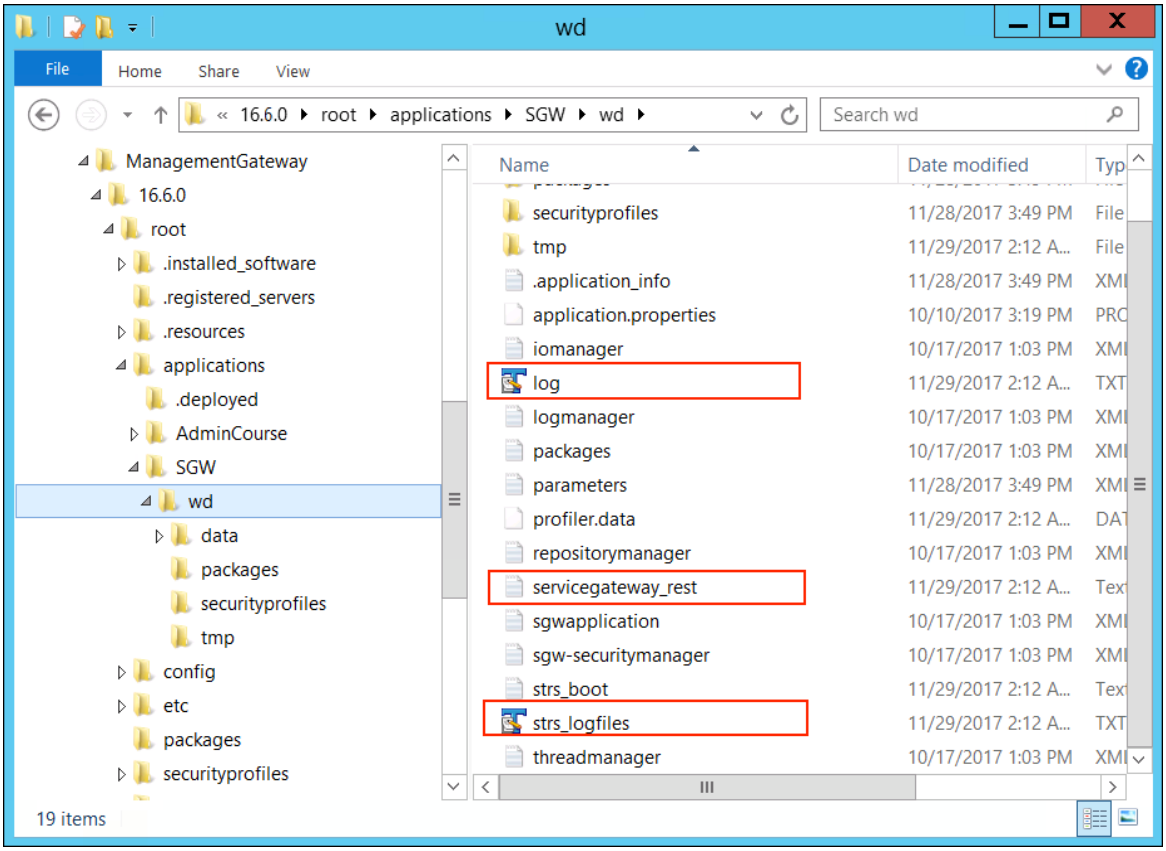


Figure 12-3: Service Gateway log files

Task Scheduler application log files

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

The location of the Task Scheduler log files is the same as the one for the application location for which the scheduler was configured.

Log messages

Log message parts

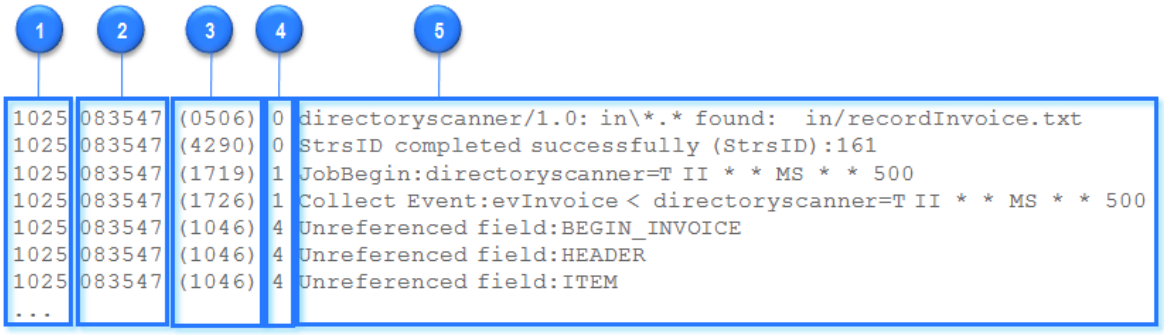


Figure 12-4: Log message parts

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 applications:

- Exstream Communications Server
- Management Gateway
- Service Gateway
- Task Scheduler

Log messages included

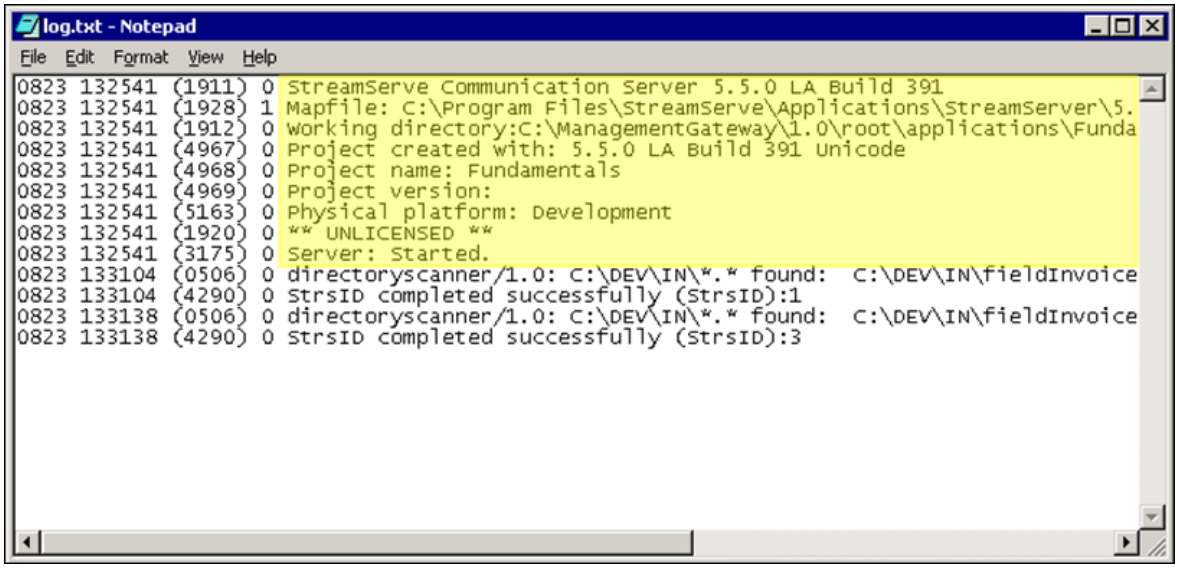


Figure 12-5: Log message always included

Some messages 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

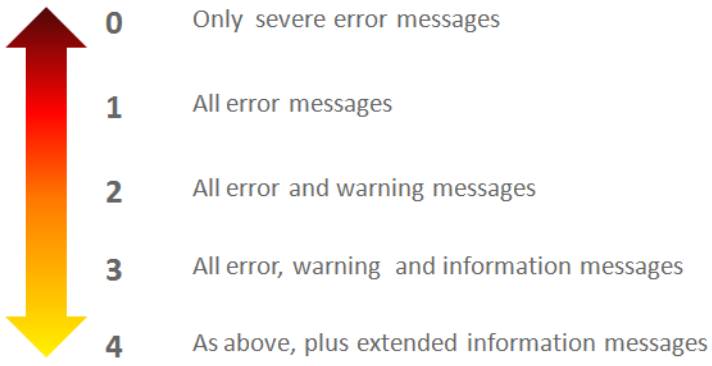


Figure 12-6: Log levels

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.

```
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
```

Setting log levels

Log levels may be set at the following levels:

Platform log levels

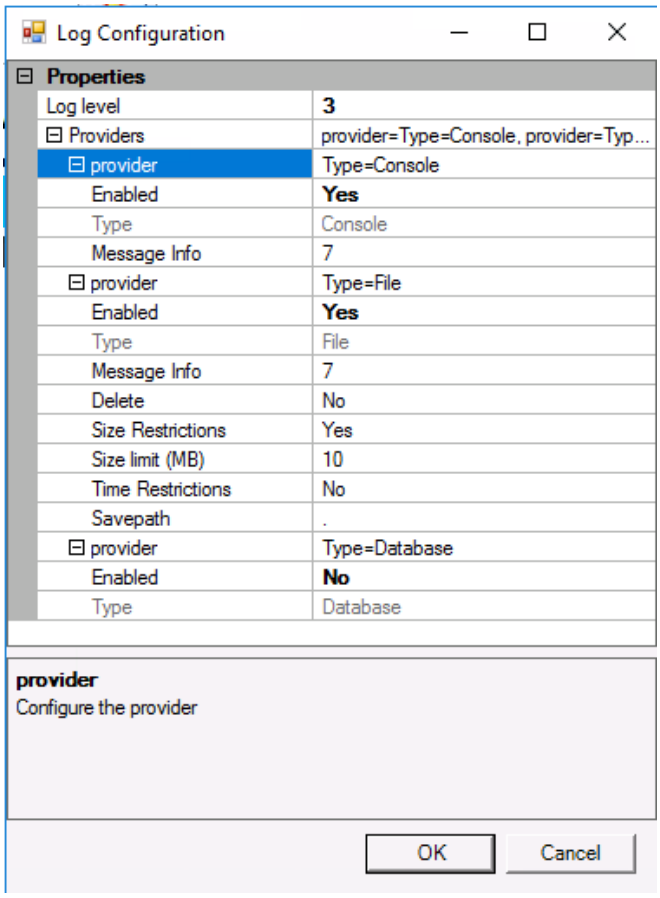


Figure 12-7:

Log Configuration dialog

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

- Exstream Communications Server
- Task Scheduler
- Service gateway

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 enable the corresponding log (Console, Database or File with its corresponding parameters) and click OK.
4. Start the application.

Setting Exstream Communications Server application log level

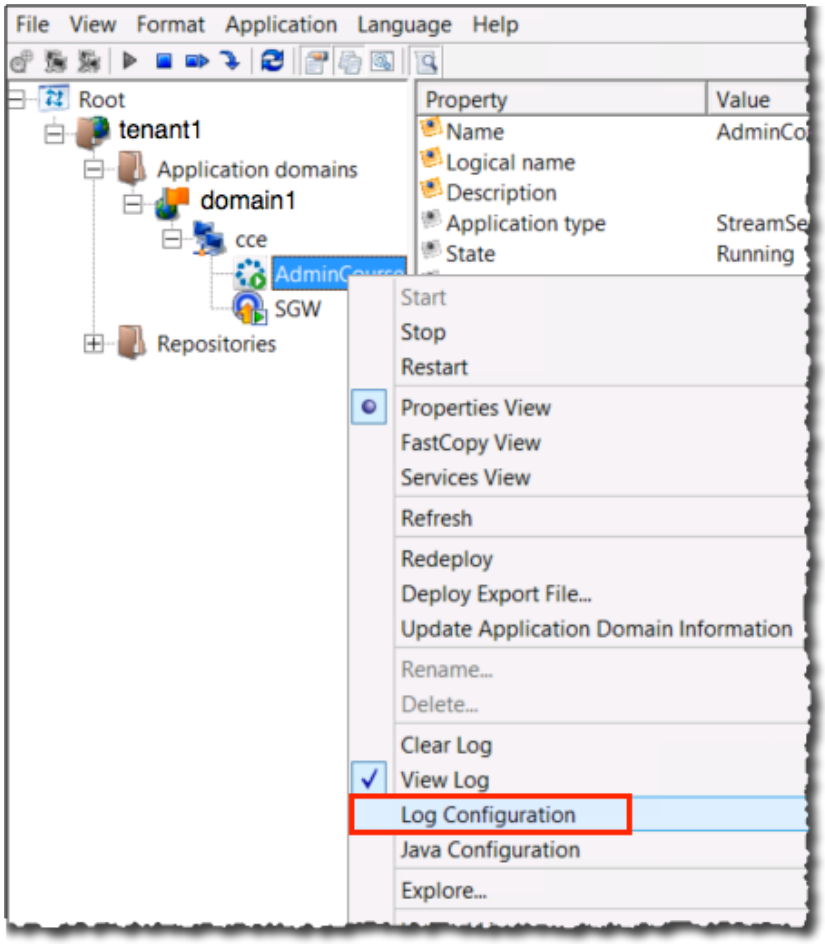


Figure 12-8:

Exstream Communications Server application log level

The log level for Exstream Communications Server application logs is specified in Communications Builder. Each time you start a Exstream Communications Server application, the log level is set to the same value as specified in Communications Builder.

You can change the log level for a running Exstream Communications Server application from Control Center, without having to change the log level in Communications Builder and export/redeploy to the Exstream Communications Server application.

To change the log level for a running Exstream Communications Server application:

1. Right-click the Exstream Communications Server application node in the tree view.
2. Set the new level in the Log Configuration window.

Debugging applications

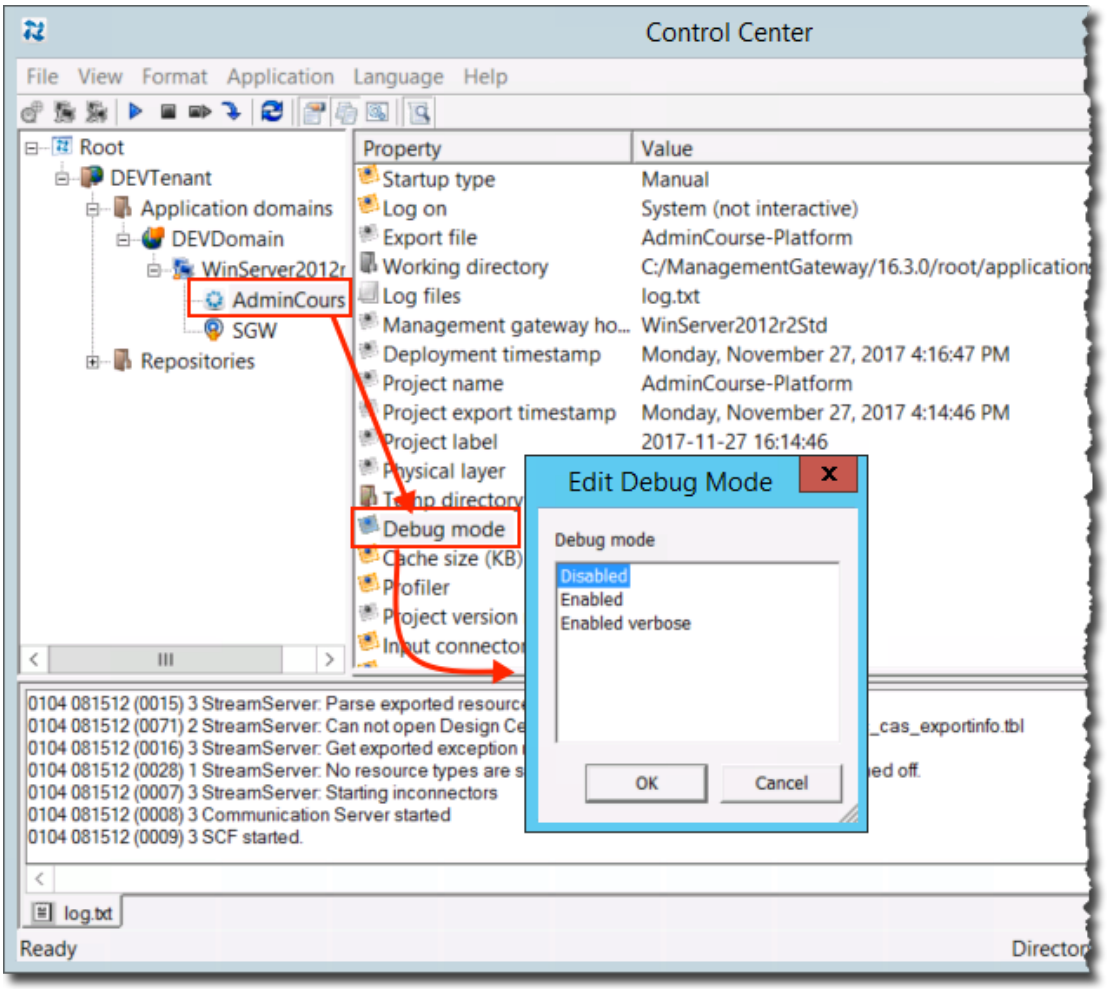


Figure 12-9: Debugging applications

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

You can enable database logging for each application, which means log messages are stored in the logging repository. By default all the applications of a given domain share the same repositories.

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 logging 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.

Exstream notifications overview

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

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

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

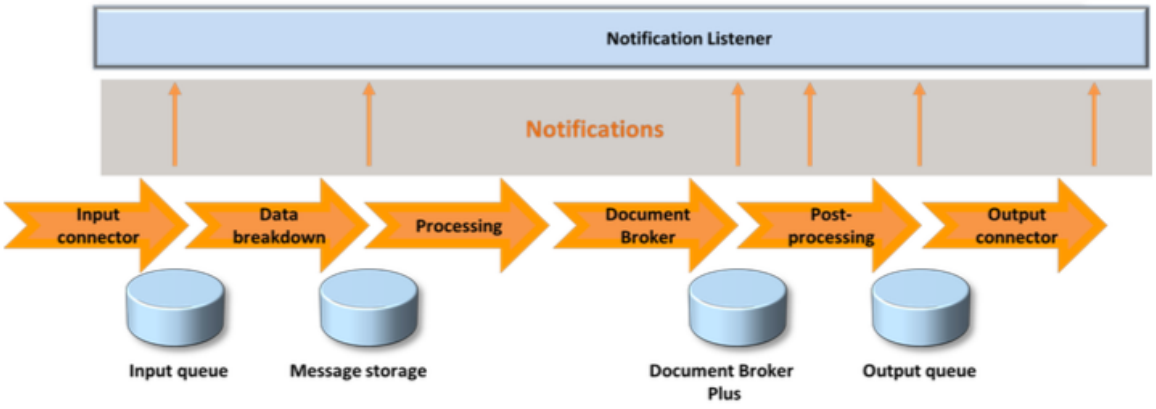


Figure 12-10: Notifications published by Communications Server applications during job processing

Exstream Communications Server notification types

Job related notifications For example, the Exstream Communications Server publishes a notification when it starts processing a top job and another notification when it finishes processing the top job.

Document broker notifications 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.

Sheet layout notifications For example, Exstream Communications Server publishes a notification each time a new sheet is started.

Other notifications For example, Exstream Communications Server publishes a notification each time a StoryTeller Process uses a resource.

Preparing the Communications Server application to use notifications

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

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

Enable Java

You must enable Java for each Communications Server application that should subscribe to notifications using a Java notification listener.

To enable Java for a Communications Server 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 Communications Server applications. Add the Java files to the resource set in Communications Builder. When you deploy the export file, Java files are placed in Data/Java folder in the working directory of your Exstream Communications Server application.
- Option 2: In the working directory of your Communications Server 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 Communications Server application.

Subscribe to notifications In order to subscribe a notification in Communications Server, 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 12-11: Subscription XML file

Task Scheduler overview

You can add a Task Scheduler application to a domain and schedule repository maintenance tasks such as deleting expired content from the Exstream repositories each night at 1:00 a.m.

Each Task Scheduler can run several tasks. To ensure that the tasks are executed even if the Task Scheduler application goes down, you can add several Task Scheduler applications.

The following tasks are available in the Task Scheduler application:

Task	Description
Run system command	Create a system command task to be executed according to the specified schedule.
Run database maintenance	Runs index maintenance (SQL Server) or coalesce indexes (Oracle) on the repositories
Delete expired content	Deletes expired top jobs, queue items, resources, Document Broker documents, stored Messages, log messages, and temporary data from the repositories.
Delete expired tracking information	Deletes tracker jobs from the Tracking repository.
Expire content	Expires queue items from Queue repository, messages from Message and Logging repositories, document broker items from the Document Broker Repository and temporary data from the Temporary repository.
Retrieve EasyLink reports	Generates reports with status information for jobs that can be sent via EasyLink.

Settings to run system command tasks

Logical task name The logical name of the system command, to be used in the log file. This parameter is optional.

Command The system command, including parameters, to be executed by the application. The command must follow the syntax of the operating system where the application is run. For example, on Windows, use backslash (\) as a directory separator and quotes ("") for arguments containing spaces. For example, the following command runs a batch file that triggers reports according to the specified schedule:

"c:\My Reports\TriggerReport.bat -type ReportType -SortOrder Alpabethical"

Working directory The name and path of the working directory for the system command task. For example, for the command above you can specify the following working directory for the reports:

c:\My Maintenance Reports

If you leave this option empty, the working directory for the application is used:

<Base directory>\<Version>\root\applications\<Task Scheduler name>\wd

Where:

- <Base directory>– Is the path specified for Exstream Projects during the Framework and Exstream Communications Server installation. For example: C:\ManagementGateway.
- <Task Scheduler name> – Is the name of the Task Scheduler application.

Configuring a Task Scheduler application

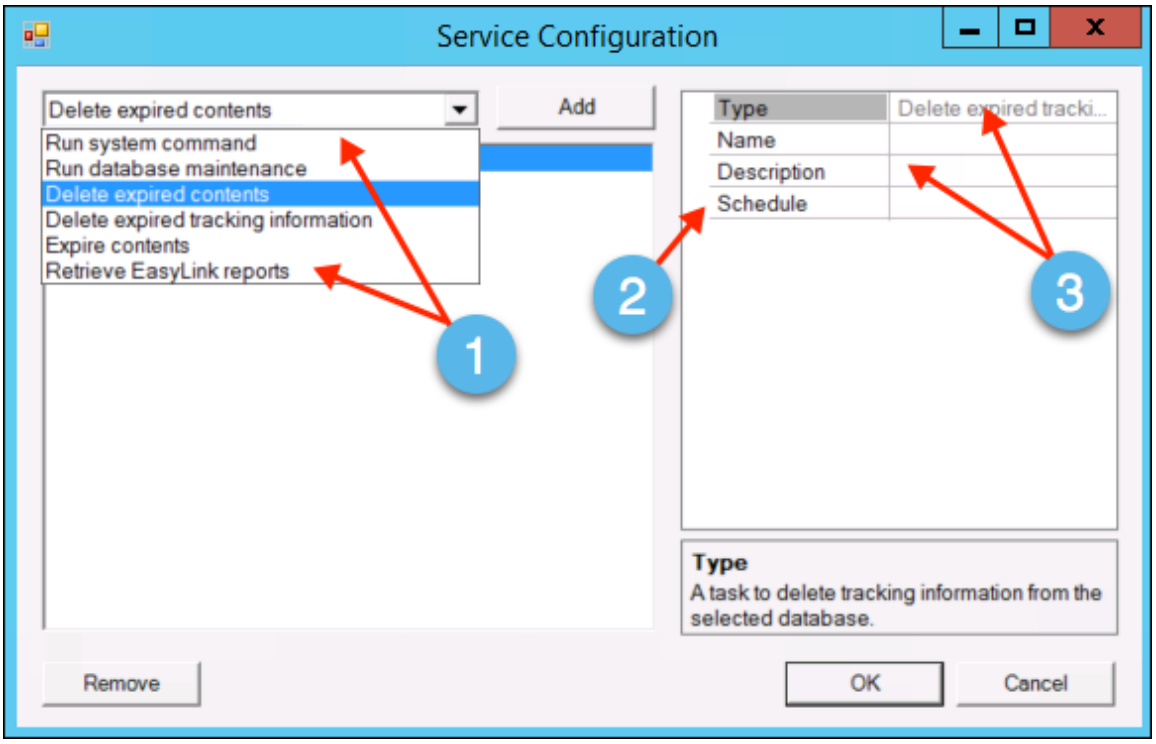


Figure 12-12: Task Scheduler

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.

Service Configuration dialog box settings

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.
(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

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.

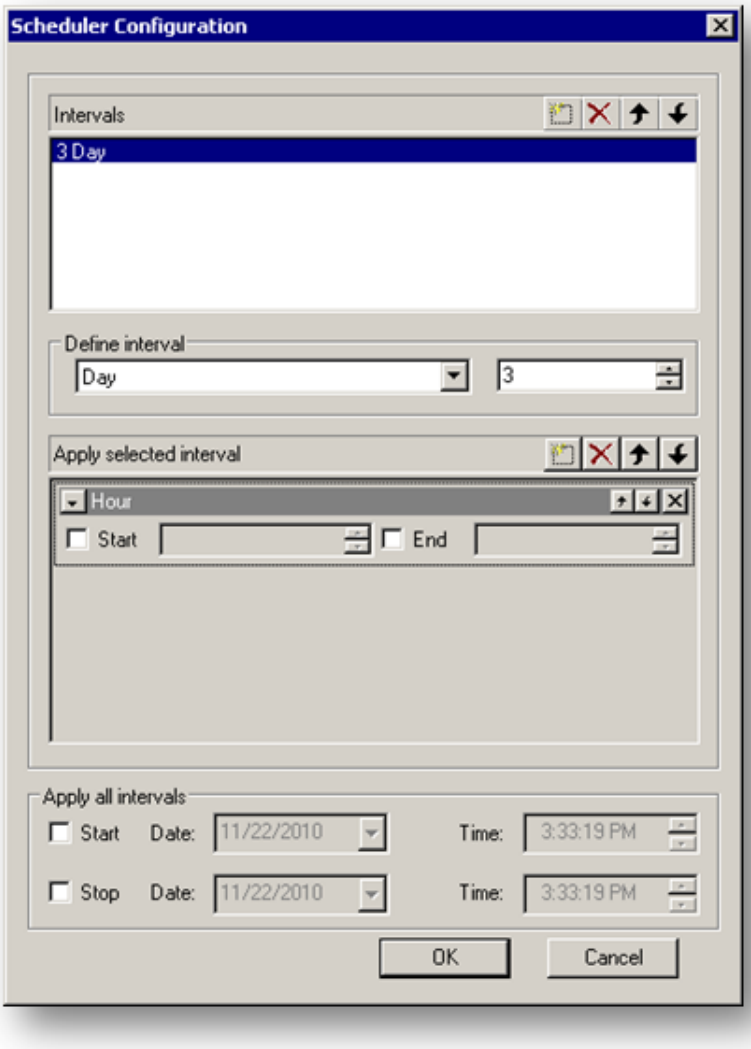


Figure 12-13:

Scheduler configuration

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.

Deleting expired content

Content in the tracking repository, Message storage, and queue repository is expired according to settings in Communications Builder. You can also manually expire this content by using Task Scheduler.

Content in the Document Broker repository must be manually expired with Task Scheduler, a process_and_delete PPQ, or a delete PPQ before it can be deleted.

Temporary data in the temporary data repository is automatically expired and no manual configuration is required. You can also manually expire this content by using Task Scheduler.

Recommended Communications Builder platform settings

- To delete successful jobs, the deletion process must be allowed to delete successful jobs. (Delete successful jobs is enabled in the Configure Platform dialog box.) We recommend a short expiry time for successfully processed jobs. Keep the number of top jobs marked for deletion to a minimum.
- To delete failed jobs, the job deletion process must be allowed to delete failed jobs. (Delete failed jobs is enabled in the Configure Platform dialog box.)

Task scheduler deletion and expiry events

Task Scheduler includes the following tasks for expiring and deleting content:

- Delete expired tracking information: for deleting tracker jobs from the tracking repository.
- Expire contents: for expiring the following types the content:
  - Queue items from the queue repository.
  - Messages from the Message repository.
  - Resources from the common asset service (which are stored in the tenant repository), which includes resources created with the Resource filter or the Resource output connector.
  - Document Broker documents from a Document Broker repository.
  - Messages from the logging repository.
  - Temporary data from the temporary data repository.
- Delete expired contents: for deleting expired content.

Recommendations for scheduling the deletion tasks

It is recommended that you schedule the Delete expired tracking information and the Delete expired contents tasks in the following way:

- Primarily, you should run these tasks at a time period when the Communications Server applications are idle or the job throughput is low. For example, after scheduled batch jobs or when the average CPU usage for the database falls below a specified value and remains below this level for a specified time period. You must make sure that the available time period is longer than the time interval required to complete the deletion after a peak load.
- If the available time periods are too short or if the workload is continuous, you should start the deletion tasks at an available time period and then schedule continuous deletion with a high frequency.
- Microsoft SQL Server - If a large number of jobs are deleted in each deletion, any defragmentation or re-indexing should run after the deletion tasks in order to avoid index fragmentation.

Scheduling a Task Scheduler to delete or expire content

When using Task Scheduler applications, you must set up separate tasks for deleting expired tracking information, deleting expired contents, and expiring contents. You must also use one task for each repository that you want to delete or expire contents from. To ensure the task runs even if the Task Scheduler goes down, you can add several Task Schedulers.

The Expired contents task requires an XML file with the DBQ to select the items that you want to expire, for example the Messages, Document Broker documents, and/or queue items.

Lab: Changing 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.

Change the log level of an application at runtime

1. *In Control Center, make sure that the **AdminCourse** application is started.*

In the log.txt panel (lower panel in Control Center) notice that you have messages with log levels 0,1,2 and 3. 3 is the default logging level.

2. *Right-click the **AdminCourse** application and select **Log Configuration** from the pop-up menu.*
3. *In the Log Configuration window, set the **Log level** to 4 and click the **OK** button.*

In the log.txt panel (lower panel in Control Center) notice that you have messages with log levels 0,1,2, 3 and 4.

Lab: 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 the applications 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.

Configure Java notifications

1. *In the Control Center right-click the **AdminCourse** application and select **Java/Server Configuration**.*

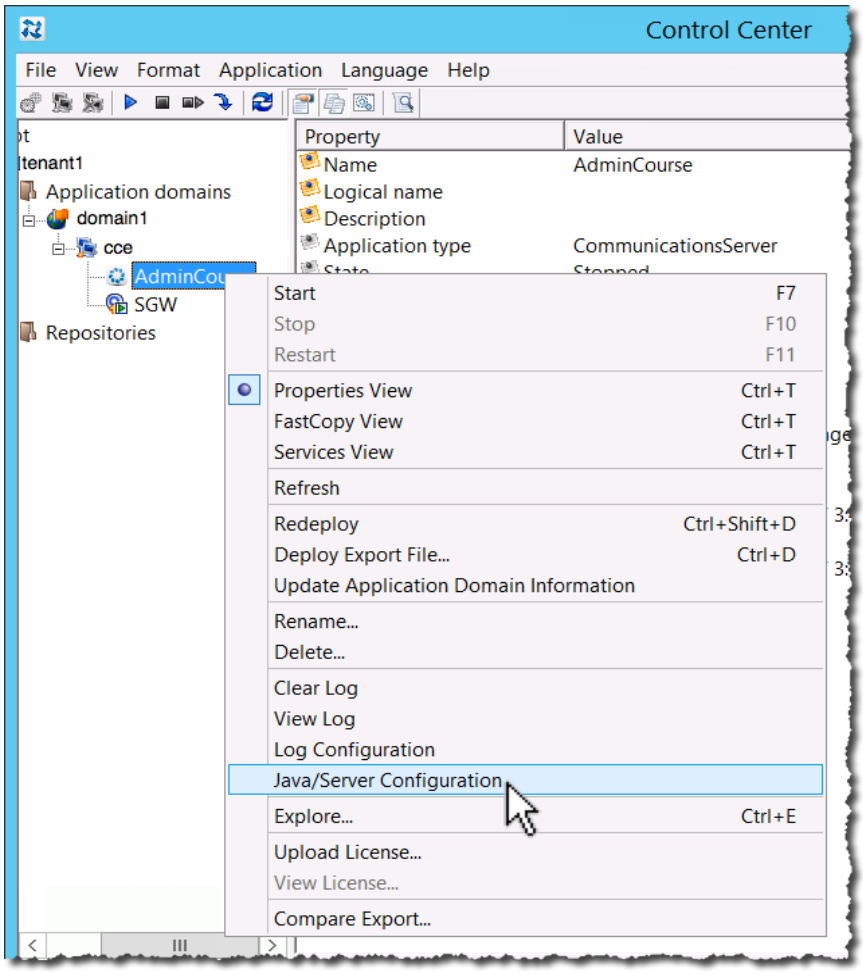


Figure 12-14:

Java configuration

The Java Configuration window opens.

2. *In the Java Configuration window, enable Java by setting the Value for the Enable Java parameter to* ***Yes*** *and click the* ***OK*** *button.*

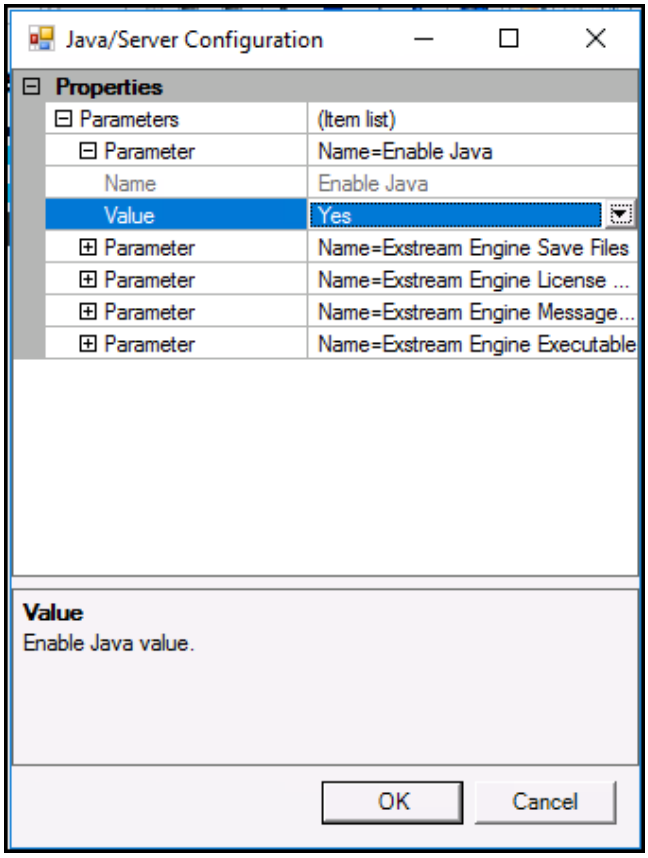


Figure 12-15:

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\16.6\root\applications\AdminCourse\data***.
4. *Copy* ***jstrscs.jar*** *from and to the locations indicated below:*
   - *From:* ***C:\OpenText\Exstream\16.6\Server\lib***
   - *To:* ***C:\ManagementGateway\16.6\root\applications\AdminCourse\data\java***
5. *Copy* ***notifications.xml*** *from and to the locations indicated below:*
   - *From:* ***C:\Training\3-3730 EXS - System Administration Files***
   - *To:* ***C:\ManagementGateway\16.6\root\applications\AdminCourse\Development\packages***

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

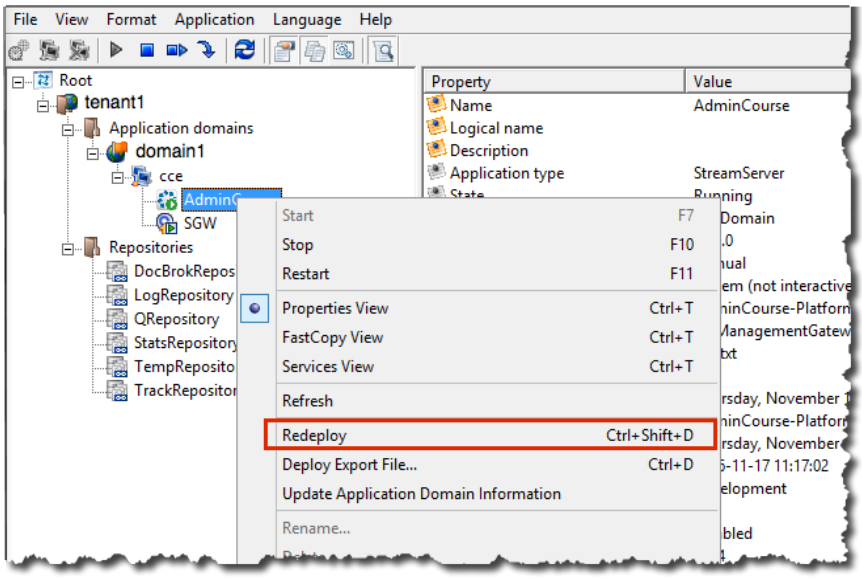


Figure 12-16:

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:\Training\3-3730 EXS - System Administration Files***
   - *To: **C:\|DEV\IN***

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

After a few seconds you will find one notification XML file in C:\ManagementGateway\16.6.0\root\applications\AdminCourse\Development\notifications for each batch processed.

8. *Navigate to **C:\ManagementGateway\16.6\root\applications\AdminCourse\Development\notifications**, right click any of the notifications (xml files) and select **Open with >XmlPad**.*
9. *Make sure the Grid View is selected.*

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

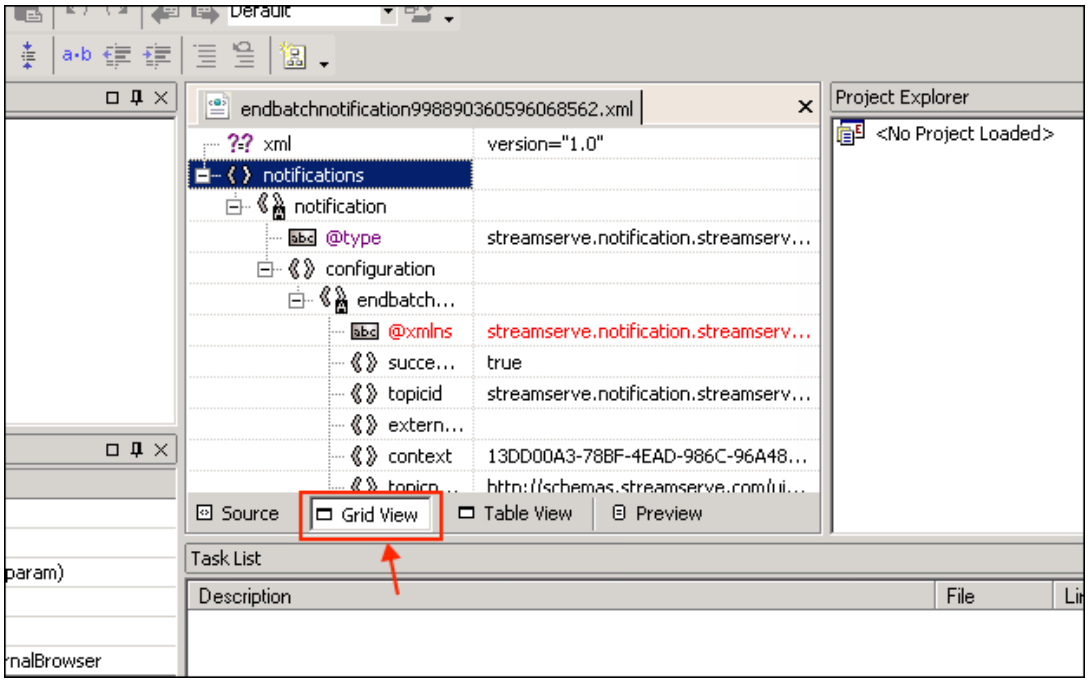


Figure 12-17: Notification XML

Lab: Creating a Task Scheduler application

In this exercise, you will create a Task Scheduler application that is used to delete expired tracking informatin from the TrackingRepository. The following steps are required for this:

- Configure the expiry time for documents.
- Disable the default schedule for deleting expired tracking information in the TrackingRepository.
- 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 tracking information from the TrackingRepository, we will set a 1 minute expiry time for successfully processed documents. This means that after a job is successfully processed, its tracking information remains in the TrackingRepository for 1 minute before being handled by the deletion process.

Configure the expiry time for documents

1. *Open the **AdminCourse** Project in Communications Builder.*
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 Allow deletion after field, make sure that **1** is selected.*
5. *In the Time units field, make sure that **Minutes** is selected.*

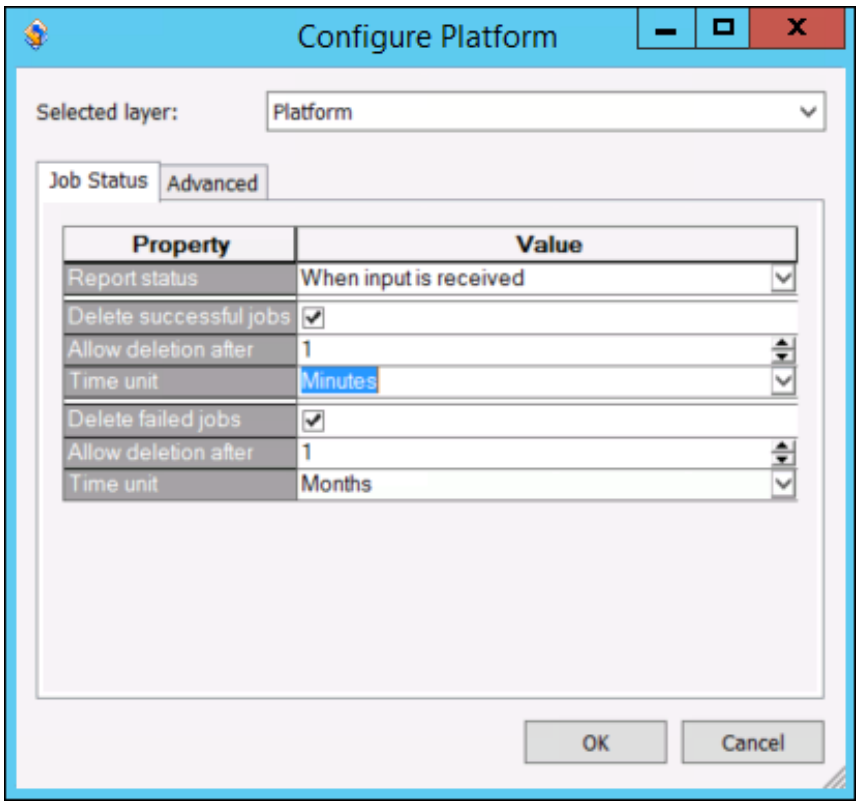


Figure 12-18:

Configure Platform

6. *Click **OK**.*
7. *Save the Project, export and deploy the application using the newly generated export file.*
8. *Start the **AdminCourse** application.*

Create the task scheduler application

1. *In Control Center, right-click the application domain and select* ***New Application****.*

The New Application dialog box opens.

2. *Configure the application properties for the new Task Scheduler.*
   - *Application type:* ***TaskScheduler***
   - *Name:* ***TS_DeleteJobs***
3. *Click* ***OK****.*

The Configuration dialog box opens.

4. *Click anywhere in the* ***Tasks*** *line and then click the* ***...*** *button in the right column.*

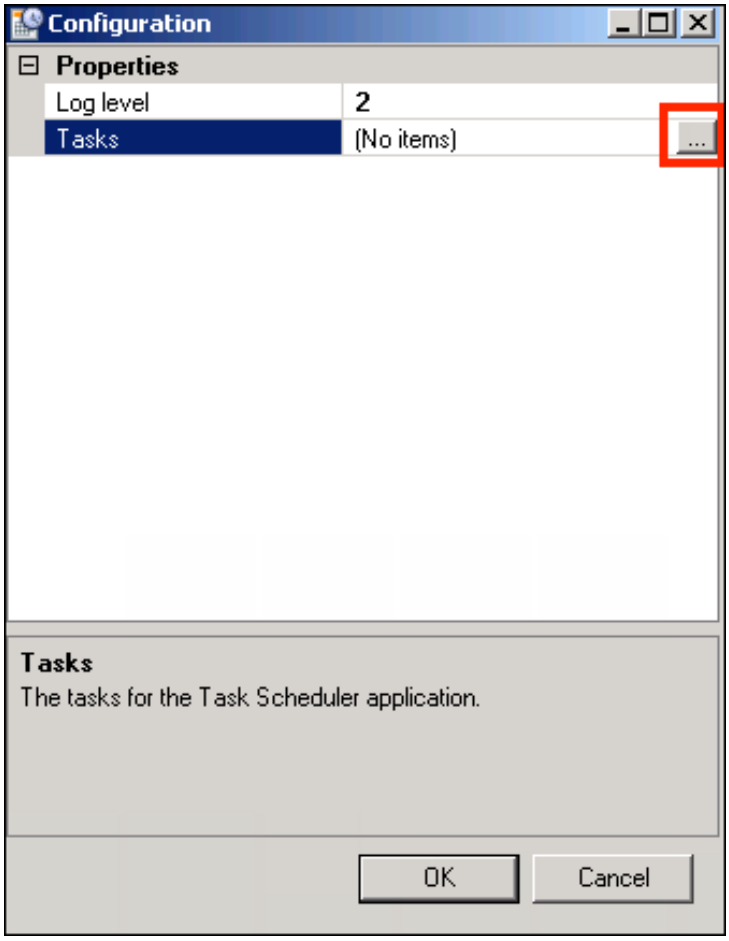


Figure 12-19:

Configuration window

The Service Configuration dialog box opens.

5. *Select the **Delete expired tracking information** task and click **Add**.*

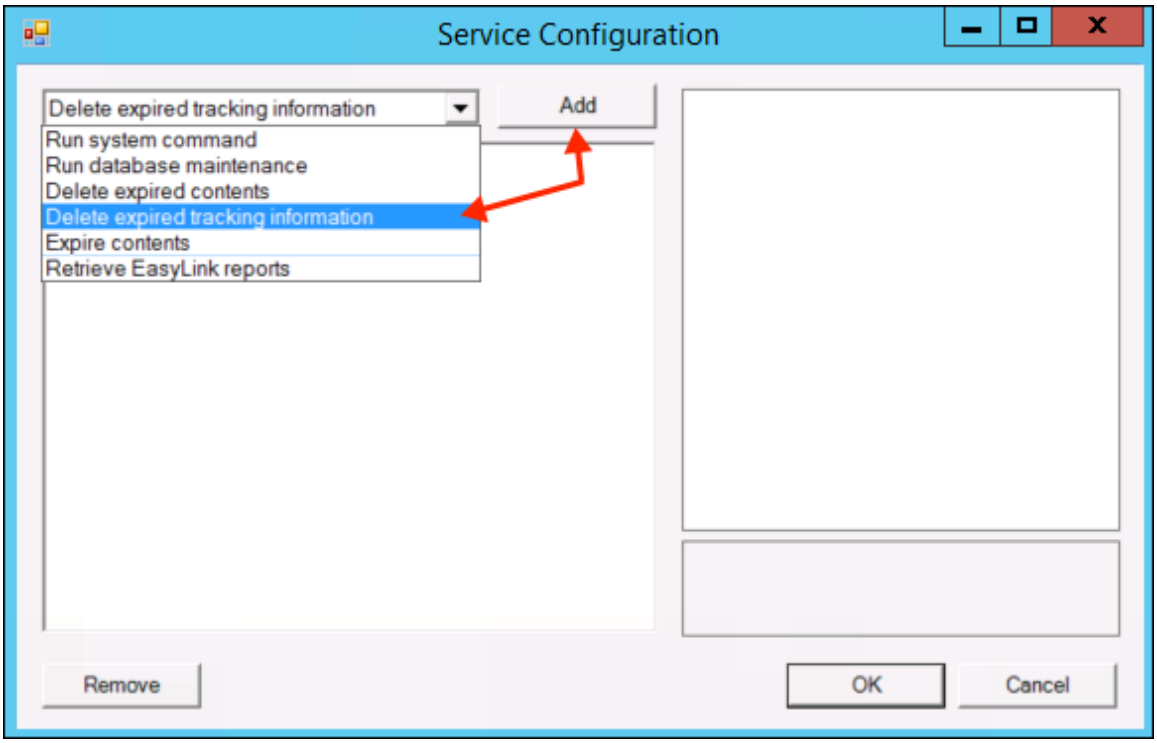


Figure 12-20: Service Configuration window

6. *Configure the properties as indicated below:*
   - *Name: **DeleteJobsTest***

7. *In the Schedule field, click the ... button.*

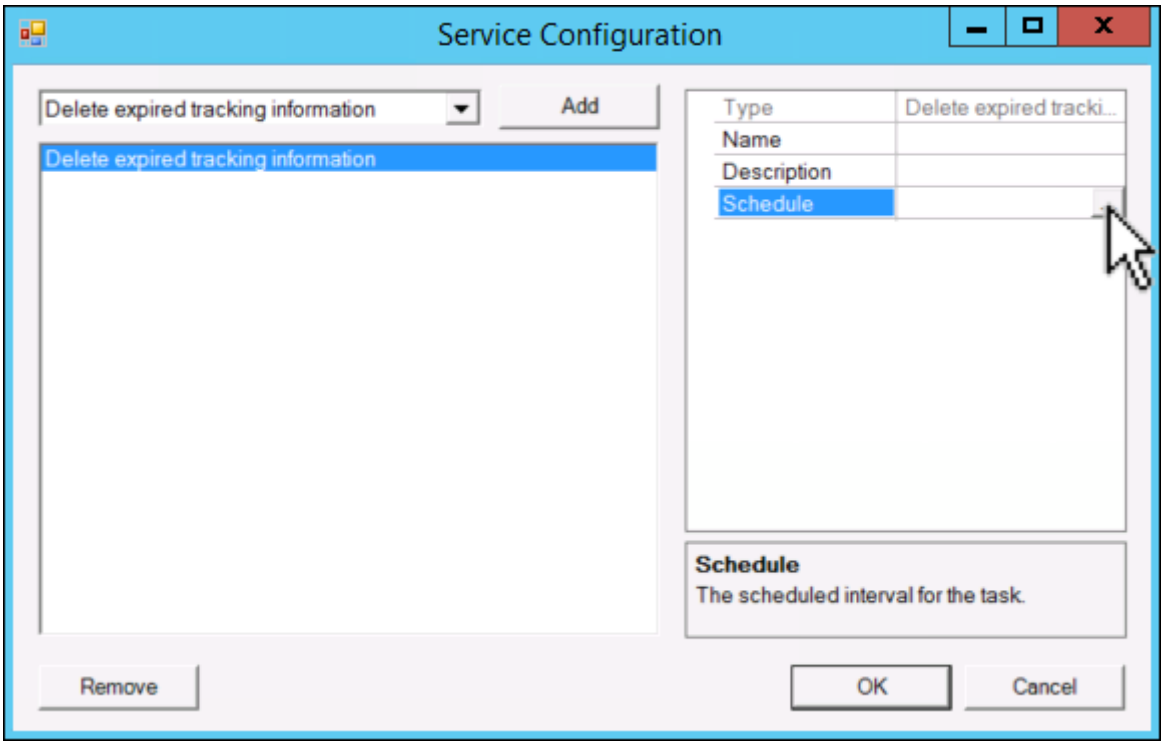


Figure 12-21: Schedule Configuration

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.*

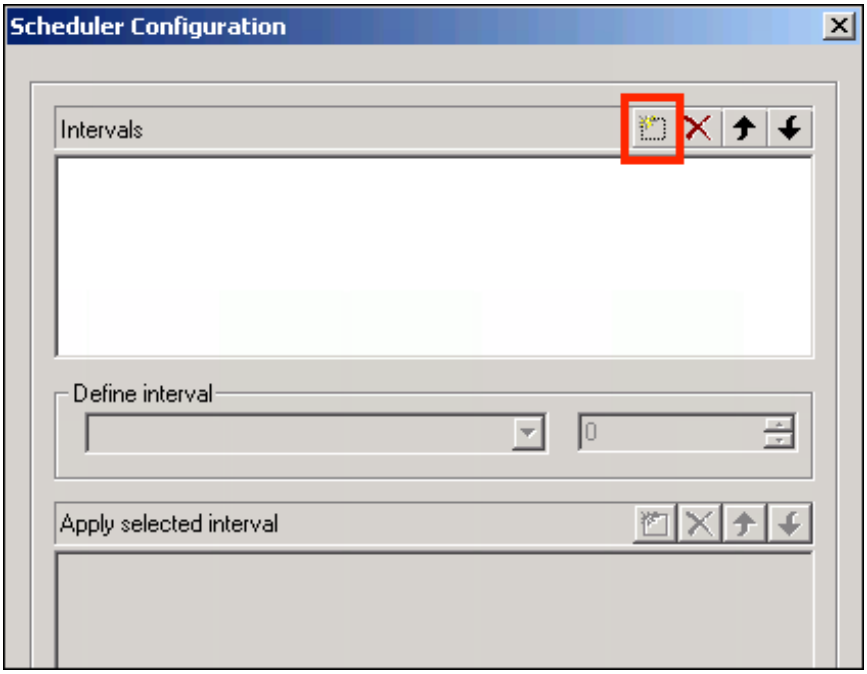


Figure 12-22:

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**.*

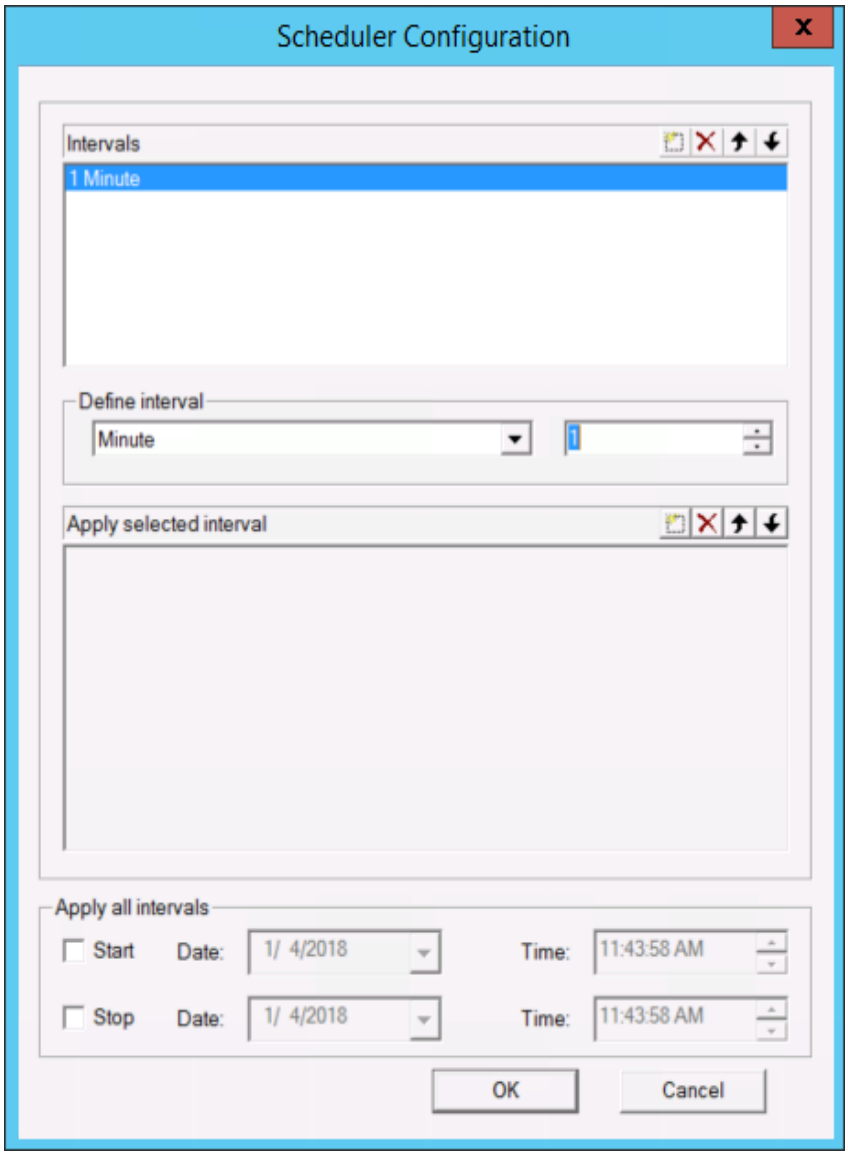


Figure 12-23:

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 information available in the Trackrepository.

View the jobs information logged in the Track repository

1. *In Microsoft SQL Server 2017 Management Studio, run the following query:*

   ```
   SELECT * FROM [DB_DEVTenant].[dbo].[Tracker] order
   by SequenceNumber
   ```

2. *View the number of entries in the results.*

Results | Messages

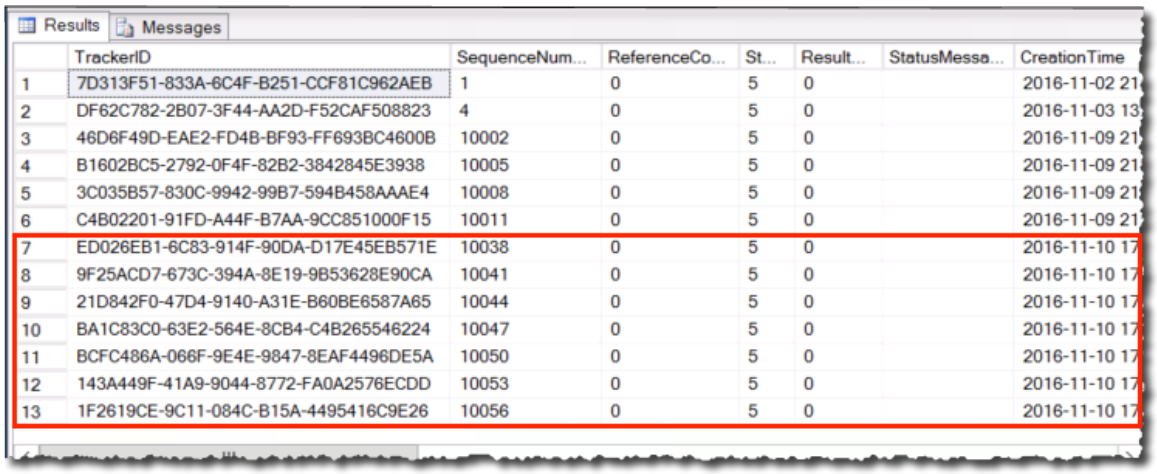
	TrackerID	SequenceNum...	ReferenceCo...	St...	Result...	StatusMessa...	CreationTime
1	7D313F51-833A-6C4F-B251-CCF81C962AEB	1	0	5	0		2016-11-02 21
2	DF62C782-2B07-3F44-AA2D-F52CAF508823	4	0	5	0		2016-11-03 13
3	46D6F49D-EAE2-FD4B-BF93-FF693BC4600B	10002	0	5	0		2016-11-09 21
4	B1602BC5-2792-0F4F-82B2-3842845E3938	10005	0	5	0		2016-11-09 21
5	3C035B57-830C-9942-99B7-594B458AAAE4	10008	0	5	0		2016-11-09 21
6	C4B02201-91FD-A44F-B7AA-9CC851000F15	10011	0	5	0		2016-11-09 21

Figure 12-24: Track repository content

3. *In Windows Explorer, navigate to **C:\Training\3-3730 EXS - System Administration Files** and copy the **recordInvoice.txt** file.*
4. *Also in Windows Explorer, navigate to **C:\DEV\IN** and paste the copied file a few times (it is important to wait a few seconds before each pasting try to me sure the file is deleted from the C:\DEV\IN folder, which is an indication that it has been picked up by Exstream Communications Server and processed).*
5. *In Control Center you can verify that the file has been processed several times (as many times as you pasted the file in the **C:\DEV\IN** folder) by reviewing the Log panel.*
6. *Back in In Microsoft SQL Server 2017 Management Studio, run the query again:*

   ```
   SELECT * FROM [DB_DEVTenant].[dbo].[Tracker] order
   by SequenceNumber
   ```

7. *View the number of entries in the results.*



Results | Messages

	TrackerID	SequenceNum...	ReferenceCo...	St...	Result...	StatusMessa...	CreationTime
1	7D313F51-833A-6C4F-B251-CCF81C962AEB	1	0	5	0		2016-11-02 21
2	DF62C782-2B07-3F44-AA2D-F52CAF508823	4	0	5	0		2016-11-03 13
3	46D6F49D-EAE2-FD4B-BF93-FF693BC4600B	10002	0	5	0		2016-11-09 21
4	B1602BC5-2792-0F4F-82B2-3842845E3938	10005	0	5	0		2016-11-09 21
5	3C035B57-830C-9942-99B7-594B458AAAE4	10008	0	5	0		2016-11-09 21
6	C4B02201-91FD-A44F-B7AA-9CC851000F15	10011	0	5	0		2016-11-09 21
7	ED026EB1-6C83-914F-90DA-D17E45EB571E	10038	0	5	0		2016-11-10 17
8	9F25ACD7-673C-394A-8E19-9B53628E90CA	10041	0	5	0		2016-11-10 17
9	21D842F0-47D4-9140-A31E-B60BE6587A65	10044	0	5	0		2016-11-10 17
10	BA1C83C0-63E2-564E-8CB4-C4B265546224	10047	0	5	0		2016-11-10 17
11	BCFC486A-066F-9E4E-9847-8EAF4496DE5A	10050	0	5	0		2016-11-10 17
12	143A449F-41A9-9044-8772-FA0A2576ECDD	10053	0	5	0		2016-11-10 17
13	1F2619CE-9C11-084C-B15A-4495416C9E26	10056	0	5	0		2016-11-10 17

Figure 12-25: Track repository content

You should have additional records, one for each file that was processed by Exstream Communications Server.

8. *In Control Center start the **TS_DeleteJobs** application.*
9. *Wait at least 1 minute which is what was configured.*
10. *Back in In Microsoft SQL Server 2017 Management Studio, run the query again:*

```
SELECT * FROM [DB_DEVTenant].[dbo].[Tracker] order
by SequenceNumber
```

Notice that the track information of the expired jobs has been removed.

13. Installing and applying Hotfixes

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

- Install and apply hotfixes

About installing and applying hotfixes

Installing a hotfix means that files in the Exstream installation folder and the management gateway base directory are replaced. If the hotfix includes database changes, you must also apply the hotfix to your Exstream repositories. If the hotfix includes changes to the web applications, you must deploy the updated Web application ARchive (WAR) files to the Java application server.

Each hotfix incorporates all previously released bug fixes for the specific Exstream release. You must install all fixes 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 the release.

Each hotfix is identified by the release to which it applies and a build number. When you refer to a specific hotfix, you should use the release name and the build number.

Hotfixes are distributed by OpenText Customer Support. The support team keeps track of all hotfixes distributed, and will provide any additional information you may need to apply the hotfix. For information about the bug fixes related to a hotfix, see the hotfix documentation.

Installing a hotfix

Prerequisites Read the hotfix documentation for any additional preparations.

If you have made customizations directly in any Exstream configuration files, these files might be overwritten when the hotfix is installed. The following applies:

Windows - To keep customizations, you must make backups of the files and store them in a place where they cannot be overwritten. When you have installed the hotfix, you can compare the old files with the new files and manually restore your customizations.

UNIX - If you use the same management gateway root when you apply the hotfix, any customizations made to the configuration files are automatically identified. For each identified customization, a comment is issued and a copy of the old file is stored. You can then compare the old file with the new file and manually restore your customizations.

An encrypted system environment must be decrypted before the hotfix installation. After the hotfix installation, the Exstream installation can be re-encrypted again.

Installing a hotfix on Windows

1. Close all Exstream programs and stop all Communications Server applications.
2. Open the Hotfix Setup wizard by double-clicking the EXE file. For example, <Version>_Hotfix_YYYY-MM-DD_Build_nnn.exe, where nnn is the hotfix build number.
3. Follow the instructions in the wizard.

Applying a hotfix to the repositories

If a hotfix includes database changes, you must apply the hotfix to the Exstream repositories.

For the tenant repository, you apply the hotfix with ss_tenantadmin.

For the Tracking, Queue, Message, Document Broker, Statistics, Logging, Temporary data repositories, you apply the hotfix in one of the following ways:

- Apply the hotfix with Control Center.
- Apply the hotfix with the hotfix-repo script.
- Apply the hotfix manually.

Prerequisites

The login details for the database administration user that you originally used to create the repository must be available.

The Hotfix Setup has been run, resulting in updated database hotfixes installed in <Base directory>\<Version>\root\config\database\ Where <Base directory> is the path specified for Communications Builder Projects during the Communications Server installation. For example: C:\ManagementGateway

Preparations

Before applying a hotfix to a repository, OpenText recommends the following:

- Make sure that no other users are connected to the repository. If possible, set the repository to single user mode.
- Stop all applications accessing the repository.
- For the queue and tracking repositories: Make sure that all the jobs running against the repository are completed.
- Perform a backup of the repository.

Applying a hotfix to the tenant repository

You use the ss_tenantadmin tool to apply a hotfix to the tenant repository. First you find the available hotfixes with the list_tenant_repository_hotfix action. Then you apply the latest available hotfix with the apply_tenant_repository_hotfix action.

If several tenants share the same tenant repository, you only need to apply the hotfix to one of the tenants. This will apply the hotfix to the entire repository.

Syntax list_tenant_repository_hotfix

ss_tenantadmin.exe -list_tenant_repository_hotfix -tenantname <tenant_name> -tenantID <tenant_ID> -mtauser <user_name> -mtapassword <password> -dbadminusername <user_name> -dbadminpassword <password>

Syntax apply_tenant_repository_hotfix

ss_tenantadmin.exe -action apply_tenant_repository_hotfix -tenantname <tenant_name> -tenantID <tenant_ID> -mtauser <user_name> -mtapassword <password> -dbadminusername <user_name> -dbadminpassword <password>

Applying a hotfix to the repositories with Control Center

In Control Center, you can list the repositories, their current schema versions, and the latest available hotfixes. This gives you an overview of the current statuses and helps you decide whether the repositories need to be upgraded. If an upgrade is needed, you can apply the latest available database hotfixes directly in Control Center.

Applying a hotfix to the repositories with the hotfix-repo script

If Control Center is not available, you can list the available hotfixes for a repository and apply the latest available hotfix by running scripts from a command prompt. First you list the hotfixes by running the list-hotfix-repo script, and then you apply the latest available hotfix by running the hotfix-repo script.

The script files reside in the following directories:

Windows <Exstream_Installation_directory>\<version>\Server\bin

UNIX .../Exstream-<Version>.GA.<build>/<version>/Server/bin/

Applying a hotfix to the repositories manually

OpenText recommends that you apply a hotfix to the Exstream repositories using Control Center or the hotfix-repo script. However, if this is not possible for some reason (for example, due to internal company restrictions), you can apply the hotfix manually.

Applying a hotfix to the web applications

When you install a hotfix that includes changes to the web applications, the WAR files in the Exstream installation directory are updated. To apply the hotfix, you must replace the WAR files on the Java application server.

Prerequisites

The Hotfix setup has been run, resulting in updated template WAR files installed in the Exstream installation directory.

When you deploy the updated WAR files, any customizations made to the configuration files in the unpacked application folders in the portal root are lost. To keep your customizations, you must make backups of the unpacked application folders, including all sub-folders, and store the backups in a place where they cannot be overwritten (that is, not in the portal root). After redeploying the WAR files, you can restore your customizations from the backups.

Replace WAR files and folders

1. Delete the old WAR files from the portal root on the Java application server. Make sure the corresponding application folders are deleted.
2. Copy the new WAR files from the installation directory and paste them into the portal root. For information about the paths, see “Copying WAR files to the Java application server” on page 211.
3. Make sure that each WAR file is deployed as an exploded WAR, resulting in a separate folder for the web application in the portal root.

Update the configuration file for web application properties

Open your configuration file for web application properties and compare the file to the cc-webapp-config.xml template in the installation directory to make sure your file contains the properties required to run the updated web applications.

14. High Availability Guidelines

Objectives

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

- Review high availability guidelines
- Define cluster-related terms
- Cluster terminology
- Planning the cluster

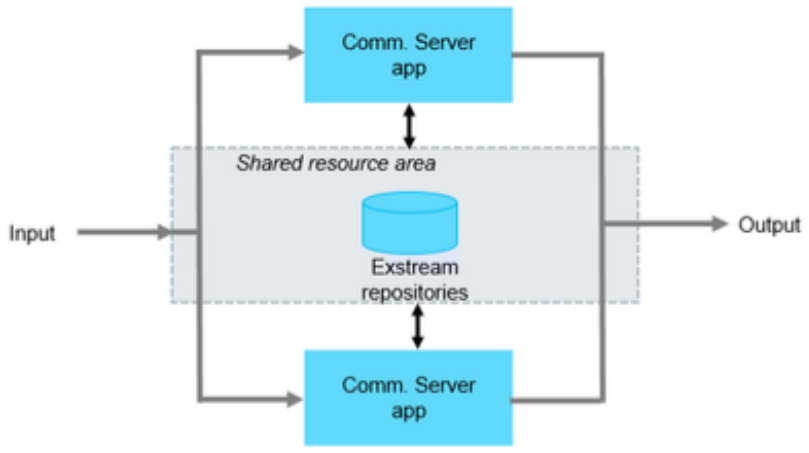
Introduction

This document provides an overview of how to set up OpenText Exstream products in a cluster environment.

The scope is to have a failover solution that provides load balancing of incoming jobs. You use several Exstream installations, and put the Exstream repositories on a shared resource area. The incoming jobs are processed by an active-active cluster solution (that is, traffic intended for a failed Communications Server application is passed on to another available Communications Server application).

Figure 14-1:

Example of an OpenText Exstream cluster environment



Cluster concepts

This section provides a brief overview of the different components of a cluster.

A cluster is monitored and handled by a third-party, platform dependent cluster software. Each cluster software provides an administration module to be used when configuring the cluster. In the administration module, you configure the cluster, including cluster nodes, cluster groups, and cluster resources.

Cluster nodes A node is a physical server in the company network. The most common size for a failover cluster is two nodes, which is the minimum to provide a failover. However, a cluster can consist of many more nodes, sometimes even dozens of nodes.

In general, we recommend that you install as much as possible on the cluster nodes, and only install common resources, such as file shares, database files, and database tables, on the shared resource area.

Cluster groups Apart from the nodes, you must also set up cluster groups where you configure the cluster resources (see Cluster resources below). Each cluster group requires a separate disk on the shared resource area.

The cluster groups reside on one of the cluster nodes. During a failure, the cluster groups are moved from one node to another in a matter of seconds.

Each cluster group is owned by one of the cluster nodes at any given time. You can assign possible owners and preferred owners (nodes) for each cluster group. If you only assign possible owners, the cluster groups are moved to another possible owner at a failover. When the original node is online and healthy again, the cluster groups remain on the second node. This saves time and resources, especially if large amounts of data needs to be moved.

However, in some situations you may want to assign different preferred owners. For example, if you run two different cluster groups in the same cluster, you can configure them to have different preferred owners. They then charge to different servers, but act as failover for each other.

Cluster resources Each cluster group includes cluster resources. A resource is an entity managed by the cluster software. For example, an IP address, a host name (network name), a physical disk, or one or more cluster services. A cluster service is, for example, a generic service or a file share.

Each resource type equates to a monitor agent type in the cluster software. Different types of monitor agents require different configurations. You can, for example, assign possible and preferred owners, and designate dependencies to other, already created, resources in the same cluster group. At a failover, cluster resources are taken offline by the cluster. When all resources are offline, the cluster group is moved to an alternate cluster node. Finally, the cluster takes the resources online again.

The order in which the cluster resources are taken offline or online is designated in the cluster dependencies. A cluster resource is not taken online, unless the dependent resources are available first. At a failure, resources are taken offline in the reversed order, meaning that the resource depending on other resources is taken offline first.

By using dependencies, you avoid unnecessary startup attempts and failures due to the fact that a resource fails because a dependent resources is yet not started.

Cluster terminology

The following terms are used in this document:

Scalability (horizontal scaling) The ability to add new physical servers to a cluster solution in order to increase job throughput. As the servers can be added to the cluster at runtime, zero down time is required.

Availability The ability to keep a service online, even if one or more physical servers are taken offline. For example, due to failure, maintenance requirements, etc.

Load balancing The ability to distribute a load across several servers.

Failover The ability to automatically switch over to a redundant or standby server in case of failure.

Fault tolerance The ability to not lose a job that is being processed at the time a failure occurs.

Planning the cluster

You must plan the cluster and decide what cluster nodes and cluster groups to be used. To eliminate every potential SPOF (Single Point Of Failure), all members of the cluster must allow failover.

How to set up the cluster depends on the company network environment, the expected loads, the performance requirements, etc. There might even be an already existing cluster on the network, which the OpenText Exstream components will share with other applications.

In general, we recommend that you install as much as possible on the cluster nodes, and only use the shared resource area for common parts.

Basic environment In a basic environment, it may be sufficient to set up a cluster with two nodes (physical servers), and install the Exstream software and the database software on both nodes.

Complex environment A more complex environment may require several nodes, where the Exstream software and the database software are installed on different nodes. You may even consider using several clusters.

High volume scenarios

When high volumes are expected and two physical Communications Servers are not expected to manage the load, you should not install the Exstream software on the same nodes as the database software.

Instead, you should install the Exstream software on separate nodes. These nodes should not be clustered, but the load should be balanced between the Communications Servers via scheduled spooling, transactional support, and using shared queues. Automatic spooling cannot guaranty job failover from another machine, however you can use automatic spooling together with scheduled spooling.

For extreme loads, you might even consider using more than two Communications Servers (assuming that the required hardware is available).

An installation with the Exstream software on separate nodes has the advantage of creating a cleaner installation, with nothing but clustered applications in the cluster. We also recommend this approach if the database shares the cluster with other clustered applications.

Exstream components

This section describes how to set up and use OpenText Exstream components, for example Communications Server applications and management gateways, in a cluster.

Exstream components are stateless. They are not cluster-aware and should not be set up as cluster resources.

However, the Exstream components do support horizontal scaling. This means you can install the Exstream software locally on the cluster nodes, or you can add new nodes with Exstream installed. Even though the Exstream components are not handled by the cluster software, you can still achieve load balancing and failover.

Failover and load balancing is achieved by sharing queues and setting up Transactional support and Scheduled spooling in the Manage Queues dialog box in Communications Builder.

When it comes to fault tolerance, the Exstream components can correct failures which occur within Exstream, the network, or the runtime repositories.

Load balancing and failover

An Exstream top job is created when an Communications Server application receives input. Each top job generates one input job processed by the Communications Server application. When the application processes the input job, it produces one or more output jobs.

Two or more Communications Server applications

If two or more Communications Server applications are running identical Communications Builder Projects and are sharing the queues, one of them can take over an input job or an output job if the currently processing Communications Server application fails. If all Communications Server applications are alive and functioning, the jobs are load balanced among the Communications Server applications.

Per child job in processing state

Since the jobs are not split, load balancing and failover are achieved per input or output job in processing state (that is, when the job is between the input queue and the output queue, or when the job is between the output queue and the output connector).

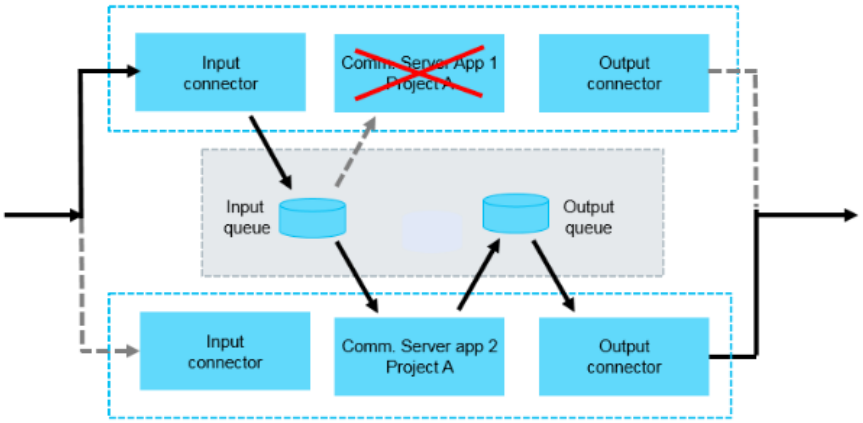


Figure 14-2:

Load balancing is achieved via shared queues, Transactional support and scheduled spooling

Load balancing via shared queues, transactional support, and scheduled spooling

Failover and load balancing is achieved by sharing queues and setting up Transactional support and Scheduled spooling in the Manage Queues dialog box in Communications Builder.

To enable load balancing, you must do the following:

- Run identical Communications Builder Projects on all Exstream nodes in the cluster.
- Set up shared input and output queues for the Communications Server applications. Note that when queues are shared, you cannot store queued documents on disk.
- Enable scheduled spooling in Communications Builder Center. The spooling interval (by default, 1 minute) specifies how often each Communications Server application polls a queue for new jobs when it is idle.

For each queue, you can edit the default spooling schedule and interval. For example, to poll the queue at certain times of the day. A shorter spooling interval decreases the execution time for a job (the time after input has been received by the input connector until output has been delivered to the output connector), but increases the workload on the database server.

Failover via heartbeats

Since Exstream itself is not monitored by the cluster, no cluster agent will control the included Communications Server applications.

Instead, heartbeats are used to monitor the status of the applications and to enable failover.

The failover is achieved between applications that must be deployed in the same Application Domain.

Failover with Communications Server applications

Each Communications Server application registers a heartbeat to the tenant database each 15 seconds by default (specified in the repositorymanager.xml in each Communication Server application working directory).

If an Communications Server application finds a job in processing state that has been inactive (that is, has not updated its heartbeat) during a specified heartbeat event interval, the Communications Server application considers the job as failed and re-queues the job.

Failover and load balancing for service gateway applications

Each service gateway applications registers a heartbeat to the tenant database each 15 seconds by default (specified in the repositorymanager.xml in each service gateway application working directory).

The management gateway polling thread supervises the heartbeats of the applications defined in the tenant.

The management gateway polls for the update of the heartbeat each 15 seconds. If the management gateway detects that the heartbeat has not been updated in the last 60 seconds, it sets the status of the service gateway application to unresponsive.

Upon a new request to the management gateway, such as a login request from an Exstream Web application, the management gateway randomly selects a running service gateway in the tenant and domain.

Note (up to 16.2) the current behavior of the web applications is that the URL of the service gateway used is defined as a cookie. When the service gateway is not responsive, the user must create a new session and clear the cache of the web application in order to re-authenticate and get a new service gateway assigned by the management gateway.

If the service gateway is used as a service end point from an application using the API for job submission, OpenText recommends using a load balancer to submit the requests containing the OTDS ticket for authentication to the provided service gateways.

Failover during web authentication request

The authentication request is sent in an authentication phase of the Exstream Web application to the management gateway URLs defined in the configuration file (cc-webapp-config.xml) which is used by Tomcat.

Database for Exstream repositories

The Exstream repositories include the multi-tenant repository, tenant repositories, and the runtime repositories (queues, message, tracking, etc.).

The runtime repositories can be installed on the same database server or different database servers depending of the hardware requirements.

Example: If you are using batches with high output rates rather than on demand or post processing operations, you can install the queue repository and tracker repository on a high-end database server that is configured for high write and read operations. These repositories can also share the same schema. You can then use a separate database (that is configured for lower read and write operations) for the other runtime repositories and the tenant repository. If you are using post processing you may benefit to separate the Document Broker repository from the other runtime repositories that may include the message store in case of on demand operations are involved.

The database instances are fault tolerant and cluster-aware, and can be managed by the cluster software.

Database cluster configurations

When clustering a database, you can either use a hot standby configuration or a mutual takeover configuration. In a hot standby configuration, at least one database server in the cluster is idle and dedicated as a backup in the event of a failure. In a mutual takeover configuration, all database servers are active and participate in the cluster.

Database software and common files

The database software must be installed on the local disk of each node. You can install the database software on the same nodes as the Exstream software, or you can use separate nodes. The database files and log files for the Exstream repositories must reside on a disk on the shared resource area.

Connections The database software accesses the shared resource area using ODBC or JDBC.

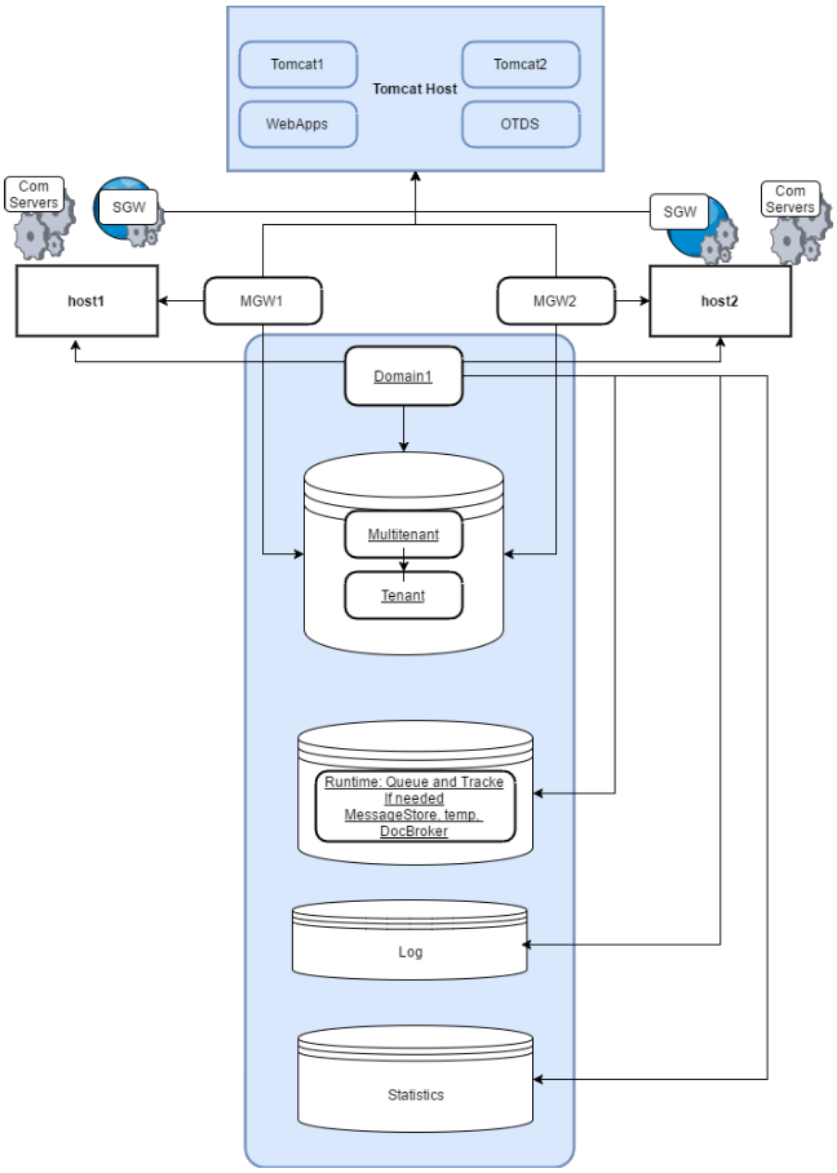


Figure 14-3:

Example of Exstream components in a cluster with two nodes

Setting up a database in a cluster

Clustering databases is a complex task, which is beyond the scope of this document. For detailed information on how to cluster a database, see the user documentation for your database vendor.

To cluster a database, you must have detailed knowledge about the database. We therefore recommend that the clustering is performed by a database administrator.

Printers and print servers

If you intend to print output to Exstream, the cluster must include one local print server for each node in the cluster, and one virtual print server for the cluster. All available printers and port monitors from all local print servers must be installed on the virtual print server.

You must set up at least three cluster resources for the virtual print server: an IP address, a host name, and a print spooler.

You can create a separate cluster group for the virtual print server, or you can include the virtual print server in an existing group.

In UNIX, most often you use an LPD (Line Printer Daemon) server as print server.

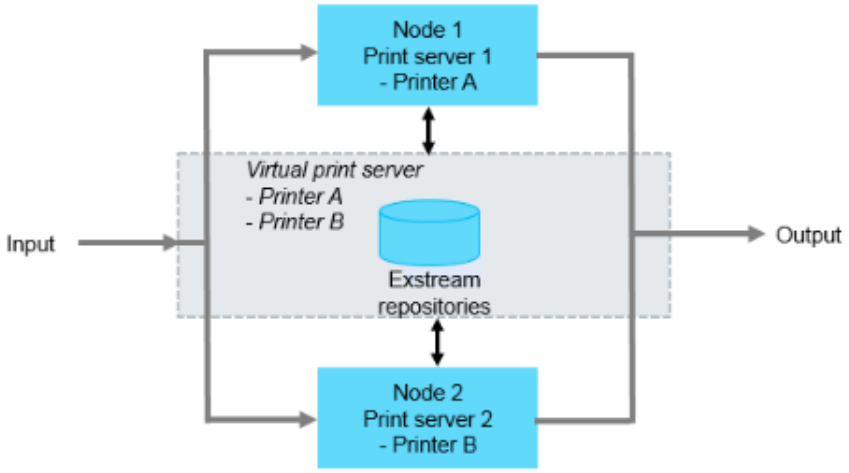


Figure 14-4:

Load balancing is achieved via shared queues, Transactional support and scheduled spooling

Exstream web applications

The Exstream web applications require the following services:

- A database instance (with the Exstream repositories) o An Exstream service gateway
- A Java application server
- A web server (optional)

To make the Exstream web applications highly available, you must make each one of the services above highly available.

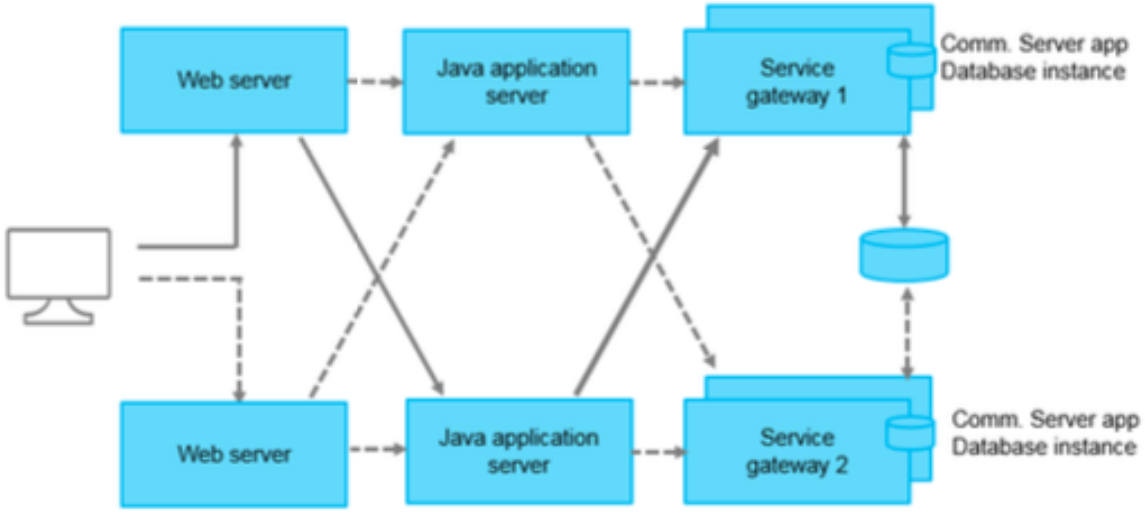


Figure 14-5: Exstream Web applications

Making the database instance highly available

The Exstream web applications use the following Exstream repositories:

- Multi-tenant repository
- Tenant repository
- Runtime repositories (Queue, Tracker, Message Store, DocumentBroker, Temp, Statistics, Log)
- Collector

The Exstream repositories reside on a database instance which is cluster-aware. You can achieve high availability for the database instance by configuring it in a failover cluster environment. For more information, see Database for Exstream repositories.

Making the Java application server highly available

The Exstream web applications run on a Java application server. For supported vendors and versions, see OpenText Exstream Release Notes.

The web applications are not cluster-aware. However, you can make the Java application server highly available, either via failover clustering using third-party tools, or via load balancing between two or more non-clustered nodes.

The Java application server is a third-party product. Descriptions of how to configure third-party products is beyond the scope of this document. For information, see the user documentation for the Java application server.

Tips when using Apache Tomcat and Apache HTTP Server 2.x

This section applies if you use Apache Tomcat as the Java application server and Apache HTTP Server 2.x as a frontend web server.

Using a JK connector running on the frontend web server, the web server can load balance incoming requests between multiple backend Java application servers. For more information, see the Apache Tomcat Connector documentation available on the Apache website.

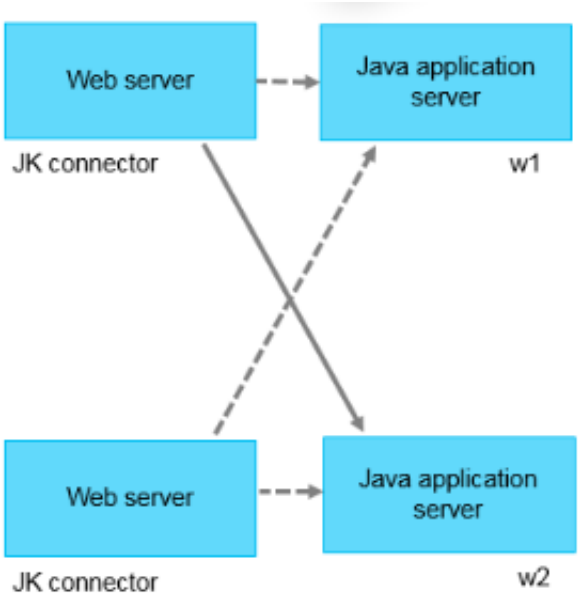


Figure 14-6:

Example of highly available Java application servers