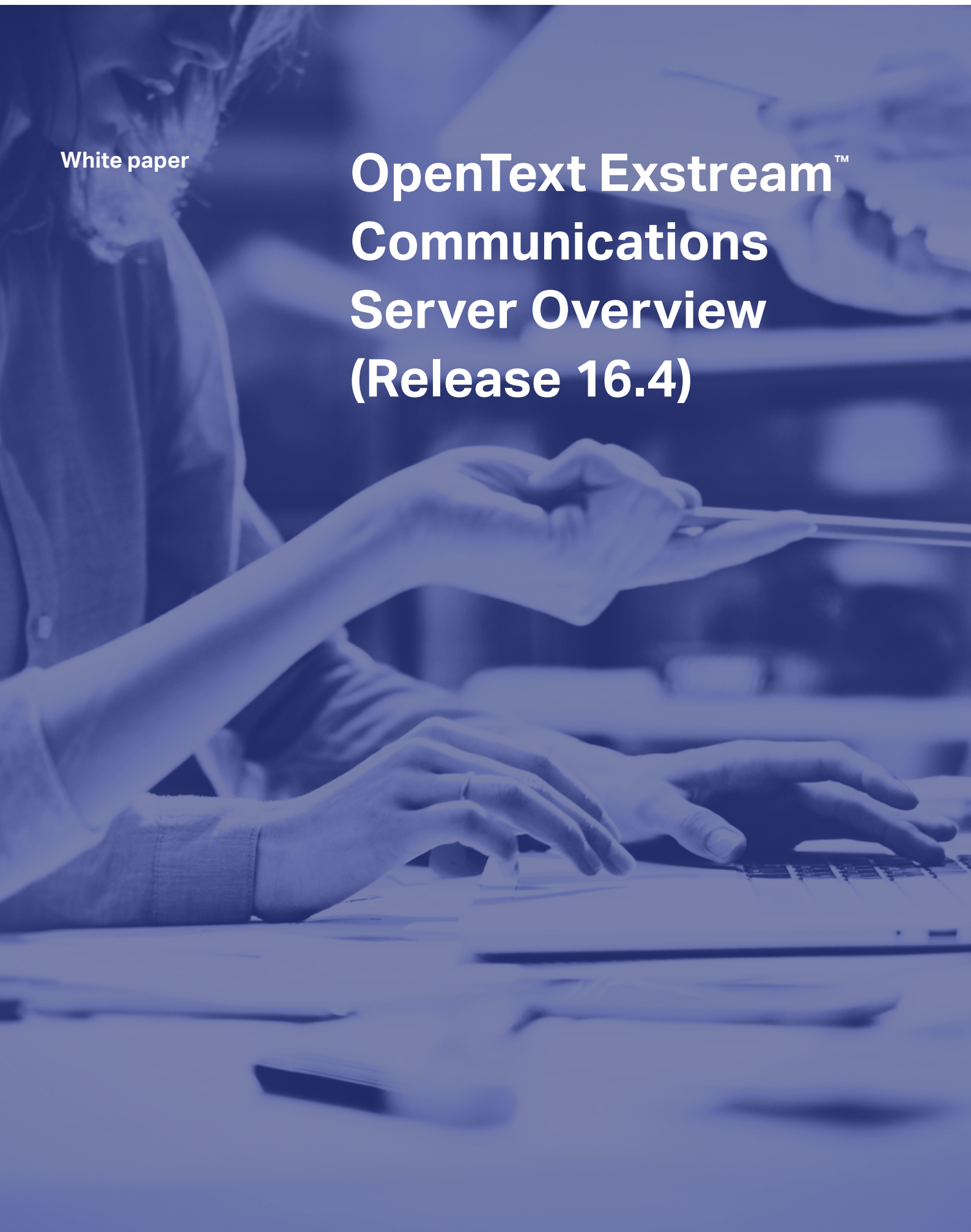


White paper

# OpenText Exstream™ Communications Server Overview (Release 16.4)



## Contents

Exstream Customer Communications platform orchestration	3
Communications Server	4
Communications Server product interfaces	4
Communications Builder	4
Control Center	5
Supervisor	5
Related components	5
Common Asset Service (CAS) and WorkShop	5
APIs	6
OpenText™ Directory Services (OTDS)	6
Repositories	6
Metadata model	6
Gateways	7
Communications Server Composition and Distribution Pipeline	8
Overview	8
Input connectors	9
Input filters	10
Processing engines	11
Document Broker and post-processing	13
Output filters	14
Output connectors	14

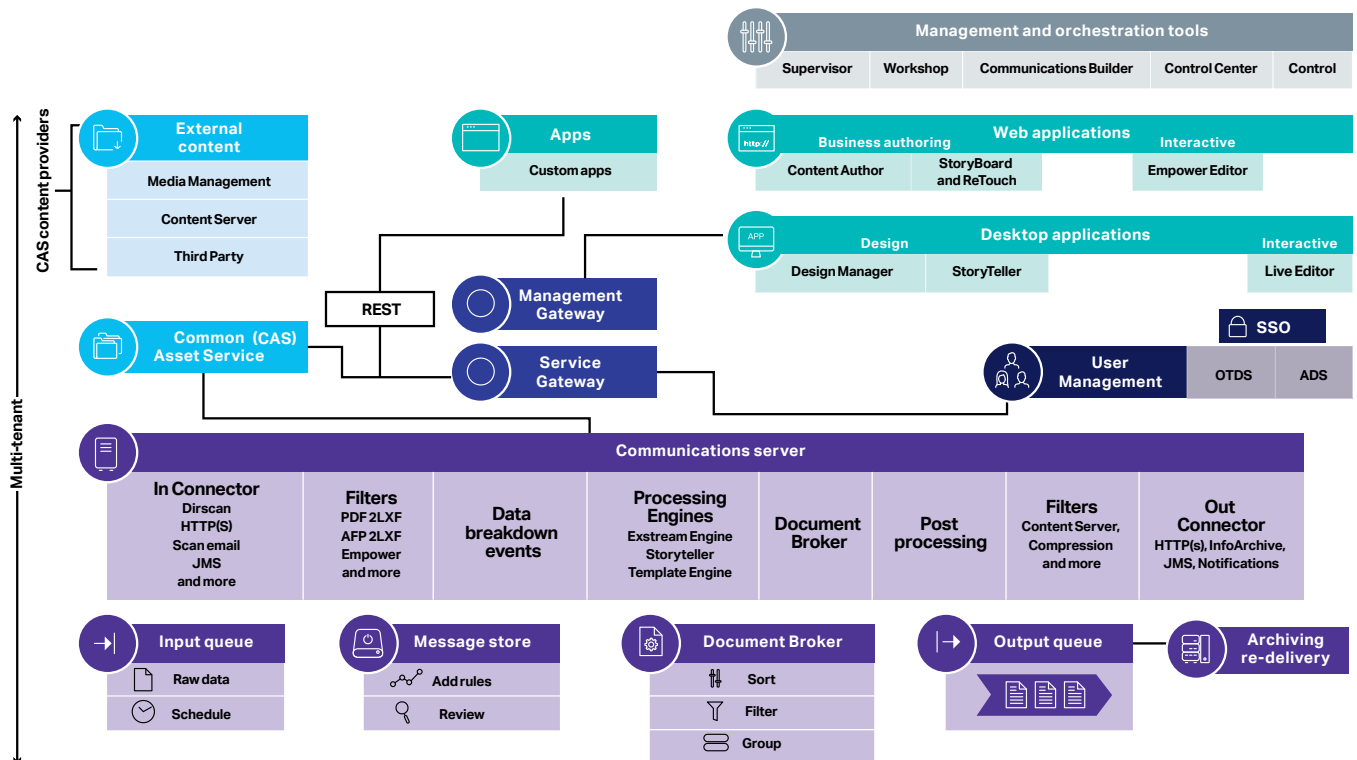
## Exstream Customer Communications platform orchestration

Exstream 16.2 and later combines multiple processing engine options and design tools into a single, powerful, integrated Customer Communications Management (CCM) platform, orchestrated by the Exstream Communications Server.

The primary role of orchestration in a CCM solution is to control execution of production engine runs and manage the inputs, outputs and data transformation processes related to the generation of highly personalized, omni-channel customer communications. With Exstream Communications Server, the execution of an engine run is realized through a flexible, yet highly efficient communications pipeline where discrete steps can be configured and modelled for the end-to-end process of producing communications; from connecting to source data through a rich array of standard connectors, to how that data can optionally be transformed through filter chains, through data mapping and composition formatting, to subsequent staging and post processing and finally to optional output transformation and physical or electronic delivery and tracking.

Each phase of the customer communications process orchestrated by Exstream Communications Server (input connectors, input filters and data breakdown events, engine processing, Document Broker and post processing, output filters and output connectors) is staged in queues, persisting in a centralized database. This architectural approach allows for dynamic, multi-threaded scaling of customer communications processes, as well as a controlled way for operators and business users to perform tasks such as previewing communications before delivery and re-sending failed communication runs.

The graphic below illustrates the phases and components of the Exstream Communications Server Composition and Distribution Pipeline.



The architecture of the Exstream Communications Server allows maximum flexibility and control, while providing a high-speed environment that supports all communications from event-driven, real-time messaging to high-volume batch processing.

## Exstream Communications Server

Exstream Communications Server provides a highly flexible and scalable environment for event-driven integration to source systems, delivering communications through a multitude of channels and protocols and orchestrating print and digital communications production.

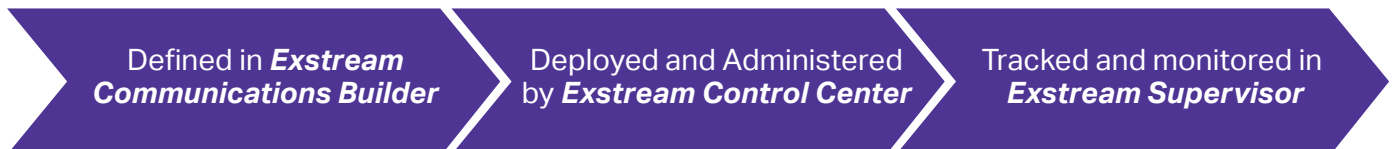
Communications Server includes the central components of the Exstream platform that connect to business systems, orchestrate the creation of business-critical documents and other types of communications and deliver communications in print or electronic format. Exstream Communications Server also contains the Common Asset Service (CAS), a multi-tenant content service layer that provides central access to, and storage for, resources used in Exstream solutions, such as project files, templates, text content, images and Exstream package files.

Offering a wide array of input connectors and filters, Exstream Communications Server processes incoming data and, if necessary, transforms it to allow for efficiency and optimized performance. During pre-processing, data can be sorted and re-arranged and additional data can be retrieved and integrated using ODBC/JDBC, web service protocols, custom API calls and more. Prepared data is then married with dynamic template designs and composed into highly personalized omni-channel customer communications and delivered through customers' preferred communication channels.

Communications Server applications run on the Exstream Framework, which contains a Management Gateway and the core platform services used by other types of Exstream applications, such as Service Gateways. Communications Server applications can be configured for various processing engines, including the classic Exstream production engine, the StoryTeller engine and a Template engine.

## Communications Server product interfaces

In the Exstream platform, customer communication processes are:



Communications Server includes the central components of the Exstream platform that connect to business systems, orchestrate the creation of business documents and other types of communications and deliver communications in print or electronic format.

## Communications Builder

The orchestration layer provided by the Communications Server run-time environment is configured using an orchestration modelling tool called the Exstream Communications Builder. Communications Builder is desktop software used to configure how input data is received (input connectors) and transformed (filters), which engine configuration is executed based on input analysis, how output from the engine is transformed (filters) and delivered (output connectors) and how conditional output delivery and sequenced or conditional engine runs can be configured.

## Control Center

Control Center is a desktop administration tool used to deploy, run and administer Exstream Communications Server applications. Control Center can also be used to manage additional services of the Exstream Framework, such as Service Gateway and Task Scheduler applications. Technically, the Control Center desktop application connects to a designated Management Gateway service over the HTTPS protocol, which allows users to easily perform remote administration tasks without having any other access to the remote environment.

In Exstream 16.4, a lightweight, web-based version of Control Center has been introduced. The "Control" web application allows for browser-based job start/stop and status tracking, access to the application logs and the ability to deploy projects and configure job event notification subscriptions.

## Supervisor

Supervisor is a web application that enables tracking and monitoring of Communications Server jobs and documents as they move through the repositories and queues during their lifecycle.

The web application provides the following functionalities:

- **Tracking:** For receiving status information about entire jobs via job trackers.
- **Review:** For reviewing paused documents before they are sent to customers.
- **Post-processing:** For examining, previewing and, if needed, deleting documents ready for post-processing.
- **Queue and delivery management:** For administering jobs, stored in queues in the queue repository. For example, resending, releasing and deleting jobs.
- **Statistics:** For monitoring processing statistics about the usage of Exstream Communications Server applications and data on key objects. The statistics are stored in the statistics repository.
- **Logging:** For examining application logs stored in the logging repository. To store logs in the repository, database logging must be enabled for the individual applications in Control Center.

## Related components

### Common Asset Service (CAS) and WorkShop

The Common Asset Service (CAS) provides central access to, and storage for, resources used in Exstream solutions, such as project files, templates, text content, images and Exstream package files. Whenever a resource is created, updated or accessed, it is done via the CAS. Many resources in the CAS are versioned and can be locked and unlocked to help prevent versioning conflicts.

CAS is connected to the Exstream Communications Server so design content can be loaded at runtime for composition. A common Service Gateway provides a central API entry point for interacting with the design and run-time environment through standard web applications or from custom built applications through an open REST API.

Exstream WorkShop is an OpenText-supplied web application integrated with CAS via the Service Gateway to provide designers and business users with an interface to manage commonly used design resources. Versioned Exstream package files, for example, can be controlled and managed with state and permission control through the Workshop interface.

## APIs

Exstream Communication Server deploys an integration-first methodology with a service oriented architecture (SOA) based on easily configured out-of-the-box connectors and event-driven composition. Standardized APIs are an essential component within Exstream Communications Server for connecting with enterprise systems or accessing resources managed by CAS. All official Exstream Communications Server APIs are Swagger documented and available by searching "Exstream" @ developer.opentext.com or directly in an installed instance by referencing the URL to the Service Gateway REST port with a suffix of /api (e.g. <http://mysgwhost:8080/api>).

## OpenText™ Directory Services (OTDS)

OTDS, the OpenText identity management system, provides the access control for Design Manager/Designer, Communications Builder, Control Center, Descriptor and the WorkShop, StoryBoard, ReTouch Control, Empower and Supervisor web applications. OTDS can synchronize with LDAP identity providers, such as Microsoft® Active Directory®, to retrieve user and group information, which is mapped to access roles providing secure access to the Exstream components. Each tenant in the Exstream environment has its own tenant OTDS configuration. Each tenant also requires access to a tenant repository.

## Repositories

The Exstream platform is built upon a series of database repositories. The types of repositories used in each environment will depend on what kind of solutions will be run. All environments, however, require repositories to store information about the environment, central resources used by the applications and processing statistics.

A tenant repository is used to store tenant specific information, such as:

- The metadata model for the tenant.
- Applications, domains and computers or hosts that are part of the environment.
- Resources available via the Common Asset Service, such as templates, images and Communications Builder Project and Exstream package files.
- Security information.

A multi-tenant repository stores information about the entire Exstream environment, which includes:

- Tenants names, descriptions, etc.
- Hosts in the environment.
- Mappings between each tenant in Exstream, the tenant OTDS and the tenant repository. All information in the multi-tenant repository is assigned a tenant ID ensuring that one tenant cannot access another tenant's applications or data.

## Metadata Model

Each tenant in the Exstream environment has its own metadata model, which contains all the properties used in the tenant's Exstream solutions. The metadata model is versioned and stored centrally. This allows the central model to be accessed by solution developers at design time and by Exstream applications at runtime in development, testing and production environments.



Metadata can be used to provide customer-defined properties that describe a communication instance or document. For example, metadata could be used in Supervisor to simplify tracking communications based on defined metadata properties attached to significant data elements, such as invoice number, account number or customer name. Another use of metadata is to enable variable data in texts or rules in Workshop. Metadata properties are populated at runtime by logic defined in the Communications Builder tool or through APIs.

## Gateways

### Management Gateway

A Management Gateway connects Exstream Platform desktop applications, such as Control Center and Communications Builder, to OTDS and to the tenant and multi-tenant repositories.

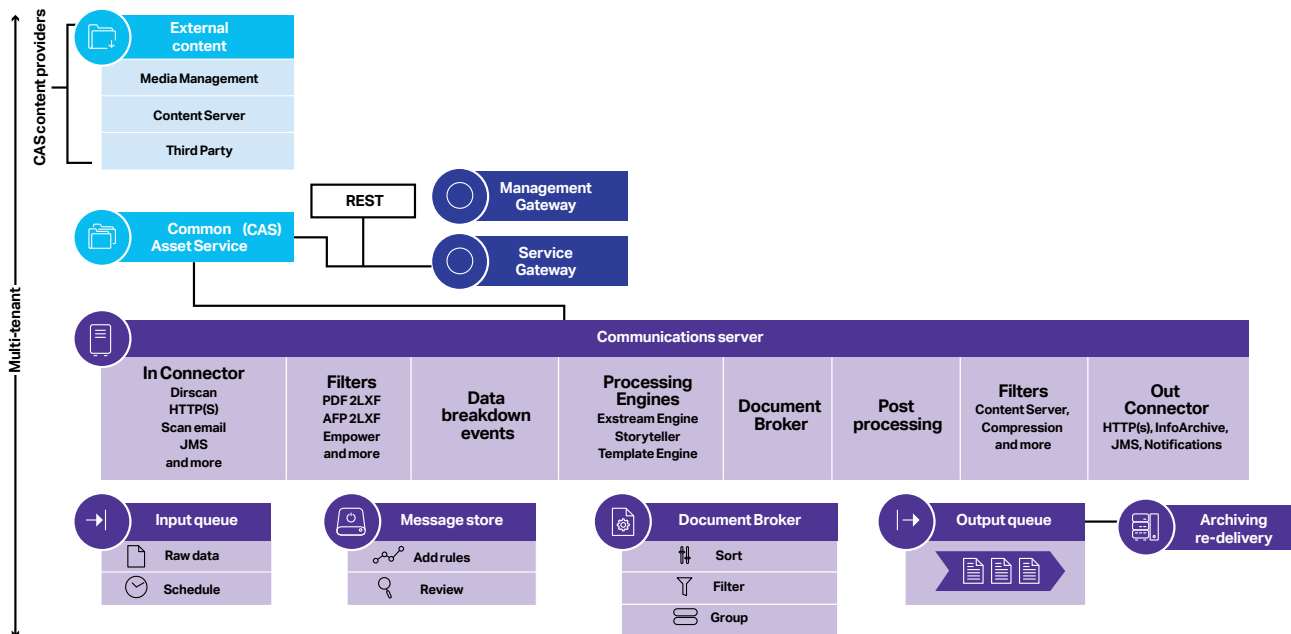
### Service Gateway

Service Gateways connect Communications Server components to Exstream web Applications, such as WorkShop and Supervisor. For failover and load balancing reasons, multiple Service Gateways can be configured.

When being accessed, a web client sends a request to the Management Gateway, asking for a Service Gateway end point. 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 then uses the returned Service Gateway to access the Exstream repositories, the Common Asset Service (CAS) and the metadata model.



## Communications Server composition and distribution pipeline



### Overview

The Exstream Communications Server Composition and Distribution Pipeline is comprised of multiple primary phases related to the generation of customer communications, including:

- **Input connectors** that allow seamless integration with enterprise and third-party systems for input data acquisition.
- **Input filters that are** able to apply standard and customized data manipulation and transformations to input data.
- Management of various composition **processing engines** for output composition, including the Exstream engine, StoryTeller engine and Template engine.
- **Document Broker** enables sorting and consolidation of documents generated by different Communications Server applications that have access to the same Document Broker repository.
- **Output filters that are** able to apply standard and customized data manipulation and transformations to generated output.
- **Output connectors** that allow integration with enterprise and third-party systems for delivery of composed output.



## Input connectors

Exstream Communications Server includes the following input connectors:

Input Connector	Description
<b>Device input</b>	This connector scans a UNIX® FIFO (first-in-first-out queue) for incoming jobs.
<b>Directory input</b>	This connector retrieves files from a named directory. The source application sends files to this directory and Communications Server retrieves the files.
<b>EmailIN</b>	This connector retrieves input sent via email.
<b>FTP</b>	This connector retrieves files from an FTP server. TCP/IP profile and Authentication profile resources are used to specify the connection settings and log-on credentials. In the TCP/IP profile resource, you can also select to use a secure channel between the connector and FTP server.
<b>HTTP(S)</b>	The HTTP and HTTPS connectors enable Communications Server to act as an HTTP server.
<b>HTTP(S) Poll</b>	The HTTP Poll and HTTPS Poll connectors enable Communications Server to function as an HTTP client that polls an HTTP server.
<b>Internal input</b>	This connector enables Communications Server to transfer data internally, for example, to pick up input received via an HTTP input connector.
<b>Java</b>	This allows a custom Java program to be used as an input connector by Communications Server.
<b>JDBC</b>	The JDBC input and output connectors provide a read-write interface for relational databases. The JDBC input connector is used to retrieve information from the database.
<b>JMS</b>	The Java Messaging Service (JMS) input and output connectors are used for transferring data via any JMS provider by using the standardized JMS interfaces.
<b>Pipe</b>	The Named Pipe input connector is used for retrieving input from a named pipe.
<b>Service Request</b>	<p>This connector exposes the Communications Server application as a web service to the client, and can be used in the following scenarios:</p> <ul style="list-style-type: none"> <li>• Generic connector that together with the configured service ID is used to receive input data submitted via the REST API communications end point</li> <li>• To retrieve output from other Communications Server applications via Service Call output connectors or Service Call filters</li> <li>• To retrieve input from Adobe® LiveCycle® ES processes</li> <li>• To retrieve IDoc data from SAP®</li> <li>• To retrieve data from SAP over the XOM interface</li> <li>• To receive the following types of notifications from Vista Plus Output Manager: <ul style="list-style-type: none"> <li>• Job status notifications that trigger custom actions, such as sending emails or updating SAP</li> </ul> </li> </ul>
<b>Delivery Manager</b>	<p>For information about using the connector in this type of scenario, see Configuration steps to set up job tracking.</p> <ul style="list-style-type: none"> <li>• Device status notifications. For information about using the connector in this type of scenario, see Configuration steps to set up device tracking.</li> </ul>
<b>Standard input</b>	This connector enables external applications to send data to Communications Server via StdIn.
<b>TCP/IP</b>	This connector enables external applications to send input to Communications Server over TCP/IP.
<b>WebSphere MQ</b>	The WebSphere MQ input and output connectors are used for transferring data via IBM® WebSphere® MQ messaging systems.

## Input filters

Exstream Communications Server includes the following input filters:

Input Filter	Description
<b>AFPIN</b>	<p>This converts AFP input to Layout eXchange Format (LXF) and enables Communications Server to identify and extract AFP formatted input documents. Typical usage scenarios are to:</p> <ul style="list-style-type: none"> <li>Recognize pages and read formatted text from the AFP document using PreformatIN.</li> <li>Convert AFP documents to other print formats, such as PCL and PostScript.</li> <li>Convert AFP documents to archival formats, such as PDF and TIFF, and reuse indexing information.</li> <li>Convert AFP overlays to LXF overlays for migration to Exstream environment.</li> <li>Convert AFP documents and hide or replace, for example, OMRs and barcodes.</li> <li>Store AFP document in the post-processor repository using AFP document metadata. The AFP documents can be sorted and included in the same envelope as other stored documents.</li> </ul>
<b>Assured delivery</b>	This filter captures and provides information to the Exstream Engine plugin on failed deliveries of customer communications for the purpose of reprocessing and delivery to an alternate output channel.
<b>Bypass</b>	This bypasses Communications Server and sends the input data directly to a designated output connector.
<b>Code page</b>	This filter is used to apply a code page for the input data.
<b>DAQ</b>	This takes a DAQ (Document Abstraction Query) as input and retrieves documents from a queue repository or Document Broker repository.
<b>Decompression</b>	This filter decompresses the input data.
<b>Empower</b>	This allows the use of Communications Server delivery capabilities for documents submitted from the Empower Editor application.
<b>External</b>	This sends and receives data to or from an external application. The filter can be used on both input and output data.
<b>Execute</b>	This filter converts characters in the input or output data using an external command similar to the External command filter, but with the difference that the Execute command can use direct files to receive input and produce output.
<b>File</b>	This converts characters in the input or output data using a conversion table.
<b>Internal</b>	This handles escape codes (HpPcl, EscO, and URL) in the input data.
<b>Java</b>	By adding the Java filter to a connector, a Java class can be called to process the data delivered to a Java connector. The Java filter can be used both as an input and output filter. This connector provides an easy API where a custom developed Java class can receive input data from an InputStream, apply some additional lookups or transformations of the input data and then write the changed data to an OutputStream. Java filters can process variables to communicate with the processing pipeline outside of the data stream itself.
<b>JavaScript</b>	The JavaScript filter can be used for both input (Input Analyzer or input connectors) and output (output connectors). The filter references a JavaScript file with JavaScript code that is applied to the stream that enters the filter. The JavaScript filter can also process and create variables used downstream in the processing chain.
<b>Job</b>	This enables Communications Server to divide large input jobs into several small jobs. Sequences in the input data determine the size of each job.
<b>Merge Sortindex</b>	This filter provides out-of-the-box integration between Communications Server and Exstream Sorting & Bundling and Application Consolidator modules. Merges all sort index and sort data files produced by the Exstream engine plugin in a first pass and provides them as input to the Engine plugin in a second consolidation run.
<b>PDFIN</b>	<p>The PDFIN filter converts PDF input to Layout eXchange Format (LXF) and enables Communications Server to identify and extract PDF formatted input documents. Typical usage scenarios are:</p> <ul style="list-style-type: none"> <li>Email attachments: Receiving PDF files as email attachments. Archiving or distributing the received files to multiple destinations</li> <li>Archiving: Converting PDF files to a format accepted by the archiving system</li> <li>Document design: Reusing the layout in the received PDF file</li> <li>Scanned documents: Receiving scanned documents that are converted to PDF format and archiving or distributing the received documents to multiple destinations</li> </ul>

*Continued on the next page*

Input Filter	Description
<b>Service Call</b>	This is used to call Communications Server applications exposed via Service Request input connectors. The Service Call filter can be used both as an input and output filter. Using a Service Call filter, users can generalize and reuse processing jobs inside the processing sequence of another job.

## Processing engines

Exstream customer communications are designed using desktop design tools, for example, Exstream Designer, Design Manager and StoryTeller. Communication designs (or "applications") are connected to source data and target delivery channels and have routing rules applied through processing flows defined in Communications Builder. The Communications Builder execution model is then deployed to the Communications Server, which orchestrates the execution of the appropriate Exstream platform processing engine to quickly and efficiently produce compelling omni-channel batch, on-demand and interactive customer communications and deliver them to targeted destinations, such as print, archive, email, web, mobile device or messaging service.

### Exstream engine

Customer communication designs created with Exstream Designer and Design Manager are compiled into portable, platform independent package files. A package file contains all of the design resources, business rules logic and data definitions required by the Exstream engine to facilitate optimized production and delivery of high-volume, on-demand and interactive customer communications. The Exstream engine can run as a standalone executable using custom scripts or mainframe JCL or be configured as a processing engine in a Communications Server orchestrated application. Exstream engine modules are licensed separately from Communications Server if a customer elects to license a specific number of transactions across the Exstream Platform. However, then the Exstream engine modules are included in this overall platform transaction license.

### StoryTeller engine

StoryTeller is a desktop application used to design the structure, content and layout of critical business communications. Once a design is complete, a StoryTeller Process definition is exported and deployed to a Communications Server application. Based on events such as the arrival of input data, web service requests from other enterprise systems or scheduled triggers, Communications Server orchestrates execution of the StoryTeller engine, which references the StoryTeller Process definition, in order to produce highly personalized and engaging customer communications to a wide variety of output device drivers and formats. StoryTeller user seats must be licensed to be able to use the StoryTeller engine with Communications Server if a customer elects to license a specific number of transactions across the Exstream Platform. However, then StoryTeller user seats are included in this overall platform transaction license..

### Template engine

The Template engine is not a traditional composition engine like the Exstream and StoryTeller engines, rather, it uses text-based content—such as HTML, XHTML and XML—as a template for generating output. The text-based template can contain static text that is output as-is, or may contain Template Engine Language (TEL) syntax that is parsed and processed by Template engine to apply further formatting. The Template engine is useful, for example, when an HTML template is already available and a communication design does not need to be recreated in Exstream Designer or StoryTeller. The existing HTML template can be processed as input by the Template engine to include additional JavaScript, apply CSS or insert interactive forms capabilities. The Template engine can be accessed as a process and via script functions

and filters. Any editor that produces text-based files can be used to create templates for the Template engine. The Template engine is included with Communications Server, with no separate licensing required.

## Input data formats

The following input data formats are supported by the Exstream and StoryTeller engines:

- Columnar/fixed-width
- Delimited
- Interactive (Exstream Empower, Live)\*
- JDBC data sources
- JSON (Exstream)
- ODBC data sources
- Page formatted (print file)
  - Plain text
  - Line printer
- AFP line data\*
- Line Conditioned Data Stream (LCDS) containing DJDEs\*
- PDF Forms
- PDF Formatted (AFP, PDF)
- Web Service
- XML

\*Exstream engine support only

## Output formats

The following output formats are supported by both the Exstream and StoryTeller engines:

- AFP
- DOCX
- HTML 4/5
- PCL 5
- PDF (PDF, PDF/A-1,2, PDF/UA, WCAG 2.0 PDF)
- PostScript
- RTF
- TIFF
- XML (multichannel)
- ZPL II

The following output formats are supported by the Exstream engine only:

- 3211 Line Data
- Exstream Empower
- IJPDS
- Live
- Metacode
- MIBF
- PCL 4
- PDF/VT
- PPML
- TOP
- VDX
- VIPP
- VPS
- XML (Composed)
- XML (Content)

The following output formats are supported by the StoryTeller engine only:

- Datamax DPL
- Image formats: GIF, PNG, BMP, JPEG
- LXF
- Matrixprinter Text
- PCL 6
- PDF/A-3
- PDF/X
- RFID (only legacy PageOUT)
- Windows Driver
- XPS



## Document Broker and post processing

Document Broker enables sorting, filtering and grouping of documents generated by various Exstream Communications Server applications to optimize production and reduce print/mail costs. Through Document Broker-provided functionality, documents produced by Communications Server applications are stored in a common repository (database), available to be collected and further processed by any Communications Server application that has access to the same Document Broker repository.

Document Broker provides the Communications Server platform with the following key capabilities:

- Document storage in a central database using the Document Broker Plus output connector and driver
- Ability to search for documents in the Document Broker repository through the use of the metadata model
- Retrieval of documents by any Communications Server application with access to the Document Broker repository, through query filters and input connectors
- Post-processing and finishing (sheet layout, sorting and enveloping) of documents collected from the Document Broker repository
- Preview of composed documents in 'Produce' view of the Exstream Supervisor web application.

## Output filters

Exstream Communications Server includes the following output filters:

Output Filter	Description
<b>Compression</b>	This filter compresses the output data.
<b>Content Server</b>	This filter makes Communications Server output available in OpenText™ Content Server.
<b>External</b>	This filter reads data from standard input, sends it to the specified filter and delivers the filtered data back on standard output. The filter can point to any executable.
<b>Execute</b>	This filter sends and receives data to or from an external application. The filter can be used on both input and output data. The filter can point to any executable.
<b>File</b>	This filter converts characters in the input or output data using a conversion table.
<b>Generic Archive</b>	This filter archives output data in an external archive. Any metadata is archived in an index file together with the data. Archiving via the Generic Archive filter is an alternative to archiving via the Generic Archive output connector. When using the Generic Archive filter, users can use any type of output connector for the final delivery of the output data.
<b>Java</b>	By adding the Java filter to a connector, users can call the appropriate Java class to use for processing the data delivered to a Java connector. The Java filter can be used both as an input and output filter.
<b>JavaScript</b>	The JavaScript filter can be used for both input (Input Analyzer or input connectors) and output (output connectors). The filter references a JavaScript file with JavaScript code that is applied to the stream that enters the filter.
<b>Resource</b>	This output filter is used to store output from the connected Process as resources in the repository.
<b>Service Call</b>	This is used to call Communications Server applications exposed via Service Request input connectors. The Service Call filter can be used both as an input and output filter.
<b>XML Stylesheet</b>	This filter is used when XML transformation is required, and applies an XSLT stylesheet to the XML.

## Output connectors

Exstream Communications Server includes the following output connectors:

Output Connector	Description
<b>Command output</b>	This enables use of commands to specify the output destination via a one-line command, batch file or script.
<b>Custom Java</b>	This connector allows a custom Java program to be used as an output connector by Communications Server.
<b>EasyLink</b>	<p>This lets users send output via OpenText™ Notifications (formerly EasyLink) cloud-based fax and notification services. By using these connectors, output can be sent via the following channels:</p> <ul style="list-style-type: none"> <li>• Email</li> <li>• Text message (SMS)</li> <li>• Fax</li> </ul>
<b>Email output connectors</b>	<p><b>SMTP (MIME) connector:</b> The SMTP (MIME) connector is used to deliver output via SMTP email. The output is retrieved from the Process connected to the SMTP (MIME) connector and/or added as attachments to the email.</p> <p><b>MAPI connector:</b> The MAPI connector is used to attach output from a Process to a MAPI email. The output is retrieved from the Process connected to the MAPI connector and attached to the email delivered via the same connector.</p> <p><b>MailOUT Process and MailOUT connector:</b> A MailOUT Process and an SMTP (MIME) for MailOUT or MAPI for MailOUT output connector can be used to send emails.</p> <p><b>EasyLink Email connector:</b> The EasyLink Email output connector can be used to send HTML or text-based output using Notifications cloud delivery services. For more information, see EasyLink connector above.</p> <p><b>SparkPost® connector:</b> The SparkPost output connector enables sending output via the SparkPost cloud service.</p>

*Continued on the next page*

Output Connector	Description
<b>Fax</b>	Output via fax can be sent using the following connectors: <ul style="list-style-type: none"> <li>• Fax Connect output connector</li> <li>• TOPCALL output connector</li> <li>• EasyLink Fax connector via the Notifications cloud delivery service</li> </ul>
<b>File Output</b>	This connector writes output to files on the file system.
<b>FTP Output</b>	This connector transfers files to an FTP server. TCP/IP profile and Authentication profile resources are used to specify the connection settings and logon credentials. In the TCP/IP profile resource, users can configure a secure channel between the connector and FTP server.
<b>SFTP Output</b>	The SFTP connector enables sending output to a sftp server.
<b>Generic Archive</b>	This connector archives output (output data and corresponding index files) in an external archiving system. The output is temporarily stored in directories before it is sent to the archiving system. Each index file contains the path to the output data in relation to the directory where the index file is stored.
<b>HTTP(S) Submit</b>	The HTTP Submit and HTTPS Submit connectors enable Communications Server to function as an HTTP client submitting output to an HTTP server.
<b>HTTP(S) Response</b>	This connector enables Communications Server to respond to HTTP requests.
<b>InfoArchive</b>	This enables sending output to OpenText™ InfoArchive, a unified enterprise archiving platform.
<b>Internal</b>	This connector enables Communications Server to loop back output to an internal input connector for further processing.
<b>JDBC</b>	The JDBC input and output connectors provide a read-write interface for relational databases. The JDBC output connector is used to insert or update information in the database.
<b>JMS</b>	The Java Messaging Service (JMS) input and output connectors are used for transferring data via any JMS provider by using the standardized JMS interfaces.
<b>Null output connector</b>	This connector enables Communications Server to send output to a dummy connector that does not direct output to any file or printer. It is used for example when a user wants to enable "Include Result in service response" to dispatch output back to the caller of a service request or to delegate all final processing to an earlier step in the processing pipeline such as an output filter.
<b>OpenText Archive</b>	This connector writes output to an OpenText Archive Document Pipelines transfer directory. Previously named LiveLink ECM.
<b>OpenText Output Server</b>	This enables integration with OpenText™ Output Server for assured delivery of business-critical documents.
<b>Pipe</b>	The Pipe output connector (UNIX only) is used for sending output to a named pipe.
<b>Push Notification</b>	The Push Notification connector lets you send push notifications to Android™ devices using the Google™ Cloud Messaging (GCM) service.
<b>Resource</b>	The Resource output connector stores output as resources in the repository. For example, a StoryTeller document can be stored as a resource via a Resource output connector, and this resource can then be wrapped into the output from a Template Engine Process.
<b>Service Call</b>	The Service Call output connector can be used to send IDoc data to a SAP system or to send output to other Communications Server applications.
<b>SNMP trap</b>	This connector sends SNMP traps to Network Management Systems (NMS) that use SNMP v1.
<b>SparkPost</b>	The SparkPost connector lets you use the SparkPost REST API to send output via the SparkPost cloud service.
<b>Spool</b>	This connector sends output to a printer.
<b>Standard output</b>	This connector enables external applications to receive output from Communications Server via StdOut.
<b>StreamServe External Viewer</b>	This connector sends output to a Previewer. The Previewer reads the file extension and opens the output file in the corresponding application (*.pdf in Acrobat Reader, etc.). The Previewer recognizes the following formats: pdf, ps, tif, dcx, html, xml, pcl. It tries to open other formats as *.txt.
<b>TCP/IP</b>	The TCP/IP output connector is used for sending output to a TCP/IP address.

*Continued on the next page*



Output Connector	Description
<b>TOPCALL</b>	Allows for page formatted output (PDF or PCL) to be attached to a MailOUT Process and sent via the Topcall connector to the attachment directory specified in the Topcall connector settings. TOPCALL collects the attachment from the attachment directory and sends it via fax or email.
<b>WebSphere MQ output connector</b>	The WebSphere MQ input and output connectors are used for transferring data via IBM WebSphere MQ messaging systems.
<b>Vista Plus Output Manager</b>	This enables sending output from Exstream jobs to Vista Plus Output Manager, which is part of the OpenText Report & Output Management Suite.

## About OpenText

OpenText, The Information Company, enables organizations to gain insight through market leading information management solutions, on-premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTEX) visit: [opentext.com](https://www.opentext.com).

## Connect with us:

- [OpenText CEO Mark Barrenechea's blog](#)
- [Twitter](#) | [LinkedIn](#)