opentext™

OpenText Exstream™ Communications Server and Command Center comparison

June 2018

Note: Please reference the official OpenText Exstream Platform, Command Center/Delivery Manager and Communications Server release notes and user guides for more information regarding specific features and functionality highlighted in this document.

opentext™

Contents

opentext™

Introduction

With the introduction of Exstream 16 in April 2017, OpenText introduced a next-generation Customer Communications Management platform, combining the best features from both Communications Server Enterprise (CCE, formerly known as StreamServe) and Exstream (formerly HP Exstream). One of the powerful CCE capabilities made available for use with the Exstream Engine is Communications Server.

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.

In the new Exstream platform, Communications Server effectively replaces and extends functionality previously provided by Command Center and Delivery Manager, introduced in 2010 as add-on capabilities to Exstream for job management and output delivery.

This document provides a comparison between the capabilities of Communications Server and Command Center/Delivery Manager.

Product interfaces

Communications Server	
Communications Builder	Desktop orchestration modelling software used to configure input/output connectors and data transforms (filters).
Control Center	Desktop admin tool used to deploy, run and administer Communications Server applications and services.
Supervisor	Web application that enables job monitoring, communications tracking and reprocessing of failed jobs, as well as viewing of output, logs, statistics and transaction tracking.
Control	Web application introduced in Exstream 16.4. This lighter-weight, web-based version of Control Center allows for job start/stop and status tracking, access to the application log, ability to deploy projects and job event notifications subscriptions.
Command Center	
Command Center Control Panel	Web-based user interface from which users can define, manage and schedule jobs and job notifications.

opentext™

User management

Communications Server	
OpenText™ Directory Services (OTDS)	OTDS provides the access control for Communications Builder, Control Center and Supervisor. OTDS can synchronize with identity providers like Microsoft® Active Directory® to retrieve user and group information that is mapped to access roles, which provides secure access to the Exstream components.
Supervisor	Provides web-based role management to control access to specific capabilities within the product.
Command Center	
Native/Proprietary	Default security settings for Command Center, user roles and permissions are managed on the 'Manage Users' tab of the Command Center Control Panel.
Third-party integration	Command Center can be configured via Spring Security to use external authentication methods (i.e. LDAP).

Platform support

Communications Server
Windows®
Linux®
Command Center
Windows
Linux
AIX®
HP-UX
Solaris

APIs

Communications Server	
API	Standardized APIs are an essential component within Communications Server for connecting with enterprise systems or accessing resources managed by the Common Asset Service (CAS). All official Exstream Communication Server APIs are Swagger documented and available by searching "Exstream" at 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).
Command Center	
API	Command Center uses action providers, which consists of an 'Action Message' that Command Center receives from an external system and an 'Action Response' that Command Center sends back to the external system. Action provider types include: File, HTTP, RMI and SOAP.

opentext™

Input connectors/Data channels

Communications Server	
Device input connector	This connector scans a UNIX® FIFO (First-In-First-Out queue) for incoming jobs
Directory input connector	This connector retrieves files from a named directory. The source application sends files to this directory and Communications Server retrieves the files.
EmailIN connector	This connector retrieves input sent via email.
FTP connector	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, users can also select to use a secure channel between the connector and FTP server.
HTTP(S) connector	The HTTP and HTTPS connectors enable Communications Server to act as an HTTP server.
HTTP(S) Poll connector	The HTTP Poll and HTTPS Poll connectors enable Communications Server to function as an HTTP client that polls an HTTP server
Internal input connector	This connector enables Communications Server to transfer data internally, for example, pick up input received via an HTTP input connector.
Java connector	Allows use of a custom Java program to be used as an input connector by Communications Server.
JDBC connector	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.
JMS connector	The Java Messaging Service (JMS) input and output connectors are used for transferring data via any JMS provider by using the standardized JMS interfaces.
Pipe connector	The Named Pipe input connector is used for retrieving input from a named pipe.
Service Request connector	This connector exposes the Communications Server application as a web service to the client.
Standard input connector	This connector enables external applications to send data to Communications Server via StdIn
TCP/IP connector	This connector enables external applications to send input to Communications Server over TCP/IP.
WebSphere MQ connector	The WebSphere MQ input and output connectors are used for transferring data via IBM® WebSphere® MQ messaging systems.
Command Center	
HTTP Data Channel	This channel uses HTTP to communicate between clients and servers. HTTP Data Channel connections are stateless and synchronous.
JMS Data Channel	This is an enterprise messaging API that supports business applications that asynchronously send and receive data and events.
Watched Folder service	This service monitors a directory in a file system for inbound requests. When a request is received, the Watch service launches a workflow.
SOAP Web Service	Command Center supports the use of SOAP and REST web service data channels. These web services are stateless and synchronous.
REST Web Service	Command Center supports the use of SOAP and REST web service data channels. These web services are stateless and synchronous.

opentext™

Input filters/Pre-processing

Communications Server	
AFPIN filter	This filter converts AFP input to Layout eXchange Format (LXF) and enables Communications Server to identify and extract AFP-formatted input documents.
Assured delivery	Communications Server captures and provides information to the Exstream Engine plug-in on failed deliveries of customer communications for the purpose of reprocessing and delivery to an alternate output channel.
Bypass filter	This bypasses Communications Server and sends the input data directly to a designated output connector.
Code page filter	This filter applies a code page for the input data.
DAQ filter	This takes a DAQ (Document Abstraction Query) as input and retrieves documents from a queue repository or Document Broker repository.
Decompression filter	This filter decompresses the input data.
Empower filter	This extracts Empower interactive document files from the Empower repository and submits as input to the Exstream Engine plugin for final fulfillment and delivery.
External filter	This filter sends and receives data to or from an external application. It can be used on both input and output data.
Execute filter	This filter converts characters in the input or output data using an external command similar to the External command filter, but the Execute command can use direct files to receive input and produce output.
File filter	File filter converts characters in the input or output data using a conversion table.
Internal filter	This filter handles escape codes (HpPcl, Esc0 and URL) in the input data.
Java filter	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.
JavaScript filter	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.
Job filter	This filter enables Communications Server to divide large input jobs into several small jobs. Sequences in the input data determine the size of each job.
Merge Sortindex filter	This filter enables classic Exstream Sorting and Bundling in Communications Server. It merges many types of 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.
PDFIN filter	The PDFIN filter converts PDF input to Layout eXchange Format (LXF) and enables Communications Server to identify and extract PDF formatted input documents.
Service Call filter	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.
Command Center	
External Management (job phase)	This allows running of external programs (such as shell scripts, Java programs, etc.) for processing data. Typical external program phases are Pre-Processing, Post-Processing and Sorting.

opentext™

Jobs administration/Management

Communications Server	
Supervisor (web application)	Supervisor allows users to perform the following tasks: **Track**–For receiving status information about entire jobs via job trackers. The job trackers, including tracker IDs and status information, are stored in the tracking repository. **Review**–For reviewing paused documents before they are sent to customers. The documents are stored in the Message repository. **Produce**–For examining, previewing and, if needed, deleting documents ready for post-processing. The documents are stored in the Document Broker repository. **Queue**–For administering jobs, stored in queues in the queue repository. For example, resending, releasing and deleting jobs. To store jobs in queues, queues must be enabled in the back-end configuration. **Collect**–For viewing and examining documents that are stored in the Collector Archive. **Statistics**–For monitoring processing statistics about the usage of Communications Server applications and data on key objects. The statistics are stored in the statistics repository. This repository is also used to monitor transaction counts and report usage for transaction-based licenses. **Log**–For examining application logs, stored in the logging repository. To store logs in the repository, database logging must be enabled for the applications. **Roles**–For managing the roles for your tenant, available via the common asset service (CAS).
Control Center	This desktop admin tool is used to deploy, run and administer Communications Server applications and services.
Control (web application)	Introduced in Exstream 16.4, this lighter-weight web-based version of Control Center allows for job start/stop and status tracking, access to the application log and the ability to deploy projects and job event notifications subscriptions.
Command Center	
Command Center Control Panel (web application)	This is used to manage user access, permissions and roles. It allows users to perform the following tasks: Create jobs (custom workflows that can retrieve data, compose documents, generate output and deliver composed output to multiple channels). Sort jobs View and manage jobs Schedule jobs Clean up/delete jobs Create and manage real-time applications Configure delivery queues Create and manage delivery reports Configure delivery trackers Manage workflows of the Delivery Manager output delivery solution

opentext™

Job initiation and scheduling

Communications Server	
Control Center Task Scheduler	This allows users to define schedules for different types of actions, for example, how frequently to collect input or pool queues. Users can set a single time interval, or create more complex schedules.
Input connectors	These provide scheduled (through scheduled spooling) and event driven processing through a wide array of input connectors (see input connectors section of this document). Jobs containing multiple connectors can have a priority assigned to each connector (highest, high, normal, low and lowest) to further control order of job processing.
Command Center	
API	Command Center lets users create schedules that specify when certain actions take place (such as running a job). Schedules can be created that run just once. It uses CRON expressions to set scheduling criteria.
JMS Data Channel	This uses action providers, which consist of an 'Action Message' that Command Center receives from an external system and an 'Action Response' that Command Center sends back to the external system. Action provider types include: File, HTTP, RMI and SOAP
Data channels	Using data channels, external applications can submit data channel requests to Command Center. These support job-based requests and real-time stream requests. Channels include HTTP, JMS, Watched Folder and Web Services (SOAP and REST).

Job notifications

Communications Server	
Notification subscription	Using the "Control" web application interface introduced in v16.4, users can subscribe to notifications on critical events such as job completion.
Java notifications SDK	Allows third-party or custom code solutions to react to server events such as job failure (e.g. send an email or push a custom notification to another system).
Supervisor	Alternatively, robust job status tracking and review/approval workflows are provided through the Supervisor web application interface.
Command Center	
Notification channels	Command Center can send job status information through preconfigured channels to a user or client program. Notification Channels include ACTION, COMMANDLINE, DELIVERY_QUEUE, EMAIL, JMS, SMS, SNMP, WEBSERVICE.

opentext™

External program execution

Communications Server	
External filter	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.
Execute filter	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.
Java filter	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.
JavaScript filter	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, which is applied to the stream that enters the filter. The JavaScript filter can also process and create variables used downstream in the processing chain.
Custom Java input and output connector	This allows Communications Server to use a custom Java program as an input or output connector.
Command Center	
External Management (job phase)	This allows the running of external programs (such as shell scripts, Java programs, etc.) for processing data. Typical external program phases are Pre-Processing, Post-Processing and Sorting.

Processing (Composition) engine support

Communications Server	
Exstream Engine	Exstream Engine modules are licensed separately from Communications Server.
StoryTeller Engine	StoryTeller user seats must be licensed to use StoryTeller Engine with Communications Server.
Template Engine	The Template Engine is included with Communications Server. Separate licensing is NOT required, unless business user access is needed to control content of the templates in StoryBoard (then an Authoring user license is needed). Template Engine is ideal for fast, text-based processing with a text-based template, such as HTML from an external design application, or just producing custom XML or JSON for systems integration purposes.
PowerDocs Engine	PowerDocs is licensed separately from Communications Server.
Command Center	
Exstream Engine	Exstream Engine modules are licensed separately from Command Center.

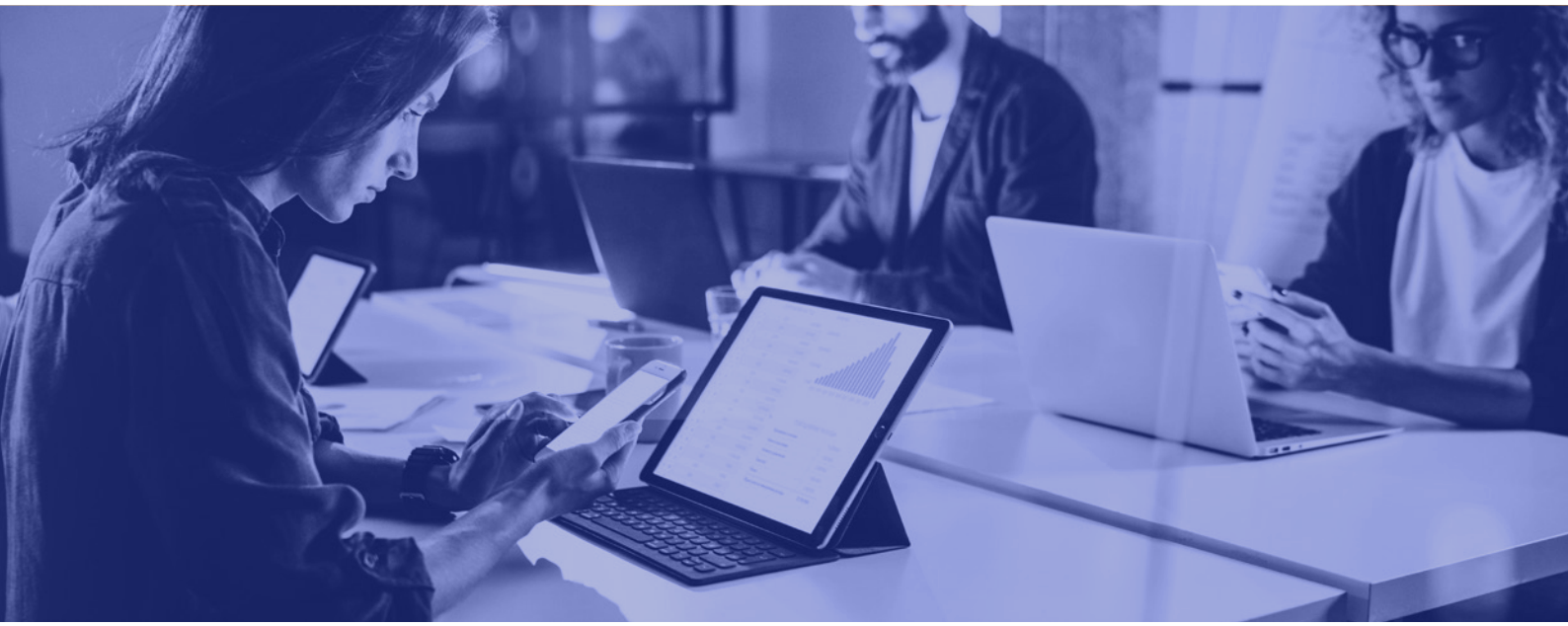
opentext™

Real-time application support

Communications Server	
Real-time (on-demand) Exstream Engine	Supports running the Exstream Engine in real-time mode as a managed process. Input is received via a wide selection of input connectors and delivered via output connectors.
Event-driven architecture	More than 15 easily configured input connectors can be used to trigger Exstream Engine processing based on events such as the arrival of input data, web service requests from other enterprise systems and more.
Command Center	
Real-time (on-demand) Exstream Engine	Exstream Engine runs continuously as a service and waits for requests. When the engine receives a request through one of Command Center's data channels, the engine composes and returns the output through another channel.

Output sorting and bundling

Communications Server	
Exstream Engine output sorting and bundling	As of v16.4, Communications Server natively integrates with Exstream Engine output sorting and bundling capabilities. The built-in Merge Sortindex filter also consolidates sort index and sort data files from first pass for inclusion into the second Engine run.
Document Broker**	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. **As of v16.4, Document Broker support for output created by the Exstream Engine is limited.
Command Center	
Exstream Engine output sorting and bundling	Command Center natively integrates with Exstream Engine output sorting and bundling capabilities.



opentext™

Output filters/ Post-processing

Communications Server	
Compression filter	This filter compresses the output data.
Content Server filter	This filter makes Communications Server output available in Content Server.
External filter	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.
Execute filter	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.
File filter	This filter converts characters in the input or output data using a conversion table.
Generic Archive filter	This filter archives output data in an external archive. Any metadata is archived in an index file together with the data. Archiving via this filter is an alternative to archiving via the Generic Archive output connector. When using this filter, users can use any type of output connector for the final delivery of the output data.
Java filter	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.
JavaScript filter	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.
Resource filter	This output filter is used to store output from the connected process as resources in the repository.
Service Call filter	This filter 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. Service Call is a powerful concept that allows running a composition job or transformation job as a step in another job.
XML Stylesheet filter	This filter is used when XML transformation is required and applies an XSLT stylesheet to the XML.
Script contexts	This allows custom logic to be executed at various phases in job processing. Scripts can use function files for re-use and have a lot of flexibility. Scripts placed after an output connector can, for example, capture the output delivery status and apply custom logic to automatically reprocess on alternative channels if primary delivery fails.
Command Center	
External management (job phase)	This allows running of external programs (such as shell scripts, Java programs, etc.) for processing data. Typical external program phases are Pre-Processing, Post-Processing and Sorting.

opentext™

Output connectors/Delivery destinations

Communications Server	
Command output connector	The Command output connector enables use of commands to specify the output destination via a one-line command, batch file or script.
Custom Java connector	This allows Communications Server to use a custom Java program as an output connector.
EasyLink connector	EasyLink connectors let users send output via OpenText™ Notifications (formerly EasyLink) cloud-based fax and notification services. By using these connectors, output can be sent via: • Fax • Text message (sms) • Email
Email output connectors	**SMTP (MIME) connector** 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. **MAPI connector** 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. **MailOUT Process and MailOUT connector** A MailOUT Process and an SMTP (MIME) for MailOUT or MAPI for MailOUT output connector can be used to send emails. **EasyLink Email connector** The EasyLink Email output connector can be used to send HTML or text-based output (email, SMS, etc.) using OpenText Notifications cloud delivery services. For more information, see EasyLink connector above. **SparkPost® connector** The SparkPost output connector enables sending emails via the SparkPost cloud service.
Fax connector	Output via fax can be sent using the following connectors: • Fax Connect output connector • TOPCALL output connector • EasyLink Fax connector via the OpenText Notifications cloud delivery service
File Output connector	This connector writes output to files on the file system.
FTP Output connector	This connector transfers files to 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, users can configure a secure channel between the connector and FTP server.
SFTP Output connector	The SFTP connector enables sending output to a sftp server.

Continued on the next page

Communications Server	
Generic Archive connector	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.
HTTP(S) Submit connector	The HTTP Submit and HTTPS Submit connectors enable Communications Server to function as an HTTP client submitting output to an HTTP server.
HTTP(S) Response connector	This connector enables Communications Server to respond to HTTP requests.
InfoArchive connector	This enables sending output to InfoArchive, OpenText's unified enterprise archiving platform.
Internal connector	This connector enables Communications Server to loop back output to an internal input connector for further processing.
JDBC connector	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.
JMS connector	The Java Messaging Service (JMS) input and output connectors are used for transferring data via any JMS provider by using the standardized JMS interfaces.
Null output connector	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.
OpenText Archive connector	This connector writes output to an OpenText Archive Document Pipelines transfer directory. Previously named LiveLink ECM.
OpenText™ Output Server connector	This enables integration with OpenText Output Server for assured delivery of business-critical documents.
Pipe connector	The Pipe output connector (UNIX only) is used for sending output to a named pipe.
Push Notification connector	The Push Notification connector lets users send push notifications to Android™ devices using the Google™ Cloud Messaging (GCM) service.
Resource connector	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.
Service Call connector	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.
SNMP trap connector	This connector sends SNMP traps to Network Management Systems (NMS) that use SNMP v1.
SparkPost connector	The SparkPost connector lets users use the SparkPost REST API to send output via the SparkPost cloud service.
Spool connector	This connector sends output to a Windows print server.
Standard output connector	This connector enables external applications to receive output from Communications Server via StdOut.
StreamServe External Viewer connector	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.

Continued on the next page

opentext™

Communications Server	
TCP/IP connector	The TCP/IP output connector is used for sending output to a TCP/IP address.
TOPCALL connector	This 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.
WebSphere MQ output connector	The WebSphere MQ input and output connectors are used for transferring data via IBM WebSphere MQ messaging systems.
Vista Plus Output Manager connector	This enables sending output from Exstream jobs to Vista Plus Output Manager, which is part of the OpenText Report and Output Management Suite.
Command Center	
Local Folder Delivery Queue	This sends documents/output produced during an Exstream Engine run to a local folder location.
Line Printer Remote (LPR) Delivery Queue	This sends documents/output produced during an Exstream Engine run to a LPR queue on a LPR server.
FTP Delivery Queue	This sends documents/output produced during an Exstream Engine run to an FTP site.
JMS Delivery Queue	This sends documents/output produced during an Exstream Engine run to a JMS provider.
CMIS destination (requires Delivery Manager)	This connects to deliver output to CMIS destination.
Custom protocols (requires Delivery Manager)	This delivers output to an unsupported protocol, such as to an external system via web services.
ExactTarget destination (requires Delivery Manager)	This is a destination that uses an ExactTarget account to send email messages and leverage the reporting functions of Delivery Engine.
File destination (requires Delivery Manager)	This connects to deliver output to a file destination.
FTP destination (requires Delivery Manager)	This connects to deliver output to an FTP destination.
FTPS destination (requires Delivery Manager)	This connects to deliver output to an FTPS destination.
HTTP destination (requires Delivery Manager)	This connects to deliver output to an HTTP destination.
IMAP destination (requires Delivery Manager)	This connects to an internal destination that the SMTP and SMTPS destinations use to handle emails that are bounced back from a destination server (returned by the destination server without being delivered).
JMS destination (requires Delivery Manager)	This connects to deliver output to a JMS destination.

Continued on the next page

Command Center	
Push Notification (requires Delivery Manager)	This delivers output to the operating system of a mobile device.
SFTP destination (requires Delivery Manager)	This connects to deliver output to an SFTP destination.
SMPP destination (requires Delivery Manager)	This connects to send SMS (text) messages to an SMPP destination.
SMTP destination (requires Delivery Manager)	This connects to send email output to an SMTP destination.
SMTPS destination (requires Delivery Manager)	This connects to send email output to a SMTPS destination.
SparkPost destination (requires Delivery Manager)	This connects to send email output to a SparkPost destination.
WS REST destination (requires Delivery Manager)	This connects to deliver output to a REST web service destination.
WS SOAP destination (requires Delivery Manager)	This connects to deliver output to a SOAP web service destination.

opentext™

Delivery tracking

Communications Server	
EasyLink (OpenText Notifications) output connector	This allows users to send output (email, SMS, fax) via OpenText Notifications cloud-based delivery services. External Job Completion tracking allows for Communications Server applications to be configured to wait for a job completion notification from OpenText Notifications before marking the job as completed.
SparkPost output connector	This enables users to use the SparkPost REST API to send output via the SparkPost cloud delivery service. It can retrieve and use status reports from SparkPost to update job status
OpenText Output Server output connector	This enables integration with OpenText Output Server for assured delivery of business-critical documents.
Command Center	
ExactTarget Delivery Destination (requires Delivery Manager)	This sends email messages and uses the reporting functions of Delivery Engine to import email delivery information from the ExactTarget email application
JMS Delivery Tracker (requires Delivery Manager)	This allows Delivery Engine to query the JMS server for information about messages sent through the JMS delivery destination.
SparkPost Delivery Tracker (requires Delivery Manager)	This tracks messages sent through delivery destinations that use the SparkPost protocol, such as the built-in SparkPost_SMTP destination.
SMS-SMPP Delivery Tracker (requires Delivery Manager)	This tracks status of SMS messages sent by the Delivery Engine destinations that use the SMPP protocol.
Custom Delivery Tracker (requires Delivery Manager)	This provides the ability to create custom delivery trackers for delivery channels without a built-in tracker.

Assured delivery

Communications Server	
Assured Delivery	As of v16.4, Communications Server supports out-of-the-box Assured Delivery, which captures delivery failures and retries to a different output channel , with the following features: Assured Delivery setting on processing engine, Assured Delivery input filter and a AddProcessingEngineSwitch function.
Connector Scripts	Script contexts on connectors can be set as 'before' or 'after' scripts in Communications Builder. This allows the use of an 'after connector' script to access output job status easily and drive follow-on actions, such as performing alternate output channel delivery based on an initial delivery status.
OpenText Output Server output connector	This enables integration with OpenText Output Server for assured delivery of business-critical documents.
Command Center	
Assured Delivery job type (requires Delivery Manager)	Assured Delivery jobs send communications to a customer's preferred destination, for example, email and receives verification that the communication has been delivered. If delivery fails, the communication can be delivered to an alternate channel, for example, print.

opentext.com/contact