opentext™

OpenText™ Content Connect

Deployment and Administration Guide

Deploy and configure OpenText Content Connect.

EDCCO250400-IGD-EN-01

OpenText™ Content Connect
Deployment and Administration Guide
EDCCO250400-IGD-EN-01
Rev.: 2025-Oct-23

This documentation has been created for OpenText™ Content Connect CE 25.4.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

Introducing Content Connect

Content Connect is a connector that provides a technology bridge between Microsoft Office desktop or web applications and OpenText™ Documentum™ Content Management repository. As an administrator, you deploy and configure Content Connect, and provide it to users as an Microsoft Office add-in for their applications. It is component of OpenText™ Documentum™ Content Management for Microsoft® 365™.

With Content Connect, users can use Microsoft Office (Word, Excel, and PowerPoint) desktop or web applications to add folders to repositories, create documents and import these into repositories, and lock and unlock repository files for editing and submit updated versions. In addition, users can perform searches, view and edit file metadata properties, and add files to Favorites.

Content Connect also enables users to use Microsoft Outlook desktop or web applications to open emails and import these and their attachments into repositories. Or, users can compose emails and add files from repositories as attachments. For more information, see *OpenText Content Connect - User Guide (EDCCO250400-UGD)*.

1.1 Terms and definitions

The following terms are used in this guide when discussing Content Connect.

Table 1-1: Terms and definitions

Microsoft Office web applications	Microsoft Office 365 thin clients (Microsoft Word, Microsoft Excel, Microsoft PowerPoint, and Microsoft Outlook).
Microsoft Office desktop applications	Microsoft Office thick clients (Microsoft Word, Microsoft Excel, Microsoft PowerPoint, and Microsoft Outlook).
Microsoft Office add-in	Application that can interact with the content of Microsoft Office documents and email using Microsoft Office APIs. The Content Connect add-in is a combination of an XML file (manifest file) and a web application (HTML or JavaScript).
Manifest file	XML file that must be loaded to enable an Microsoft Office add-in. Content Connect utilizes two manifest files: one for the Microsoft Office add-in (Microsoft Word, Microsoft Excel, and Microsoft PowerPoint), the other for the Microsoft Outlook add-in.

Note: The product *Release Notes* on My Support (support.opentext.com) provides the details of the supported version of Microsoft Office.

For information about deploying Content Connect on the cloud platforms, see *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.

1.2 Supported repositories and services

Content Connect supports:

- Repositories: OpenText™ Documentum™ Content Management Server
- Services: OpenText™ Documentum™ Content Management Foundation REST API, OpenText™ Documentum™ Content Management Client REST API

Note: The product *Release Notes* on My Support (support.opentext.com) provides the details of the supported version of repositories and services.

Chapter 2

Deploying Content Connect

This section provides the deploy and upgrade instructions for Content Connect.

You can deploy and configure Content Connect with or without OpenText™ Documentum™ Content Management client.

Note: This guide is applicable to both Classic View and Smart View interfaces unless addressed specifically as applicable for the Smart View interface.

You must deploy and configure various products to use various features. The following table lists the products that may be required for your Content Connect deployment.

Products	Content Connect with and without OpenText Documentum Content Management (CM) client
NodeJS	Mandatory
PostgreSQL/Microsoft SQL/Oracle	Mandatory
OpenText Documentum Content Management (CM) Foundation REST API/ OpenText Documentum Content Management (CM) Client REST API	Mandatory
Documentum xPlore	Mandatory
App Registration with Microsoft Entra ID	Mandatory
OTDS	Mandatory
Brava	Optional
OpenText™ Documentum™ Content Management Branch Office Caching Services / ACS	Optional
Documentum Security Integration Services (DSIS)	Optional

2.1 Prerequisites

Deploy or upgrade the following instructions, ensure that your system meets the supported environments and compatibility requirements stated in the product *Release Notes* on My Support (support.opentext.com).

- NodeJS: Download and install NodeJS.
- Database: PostgreSQL or Microsoft SQL or Oracle
- If you are configuring Content Connect with OpenText Documentum CM client, then refer to "Configuring OpenText Documentum CM client" on page 45 after you deploy Content Connect.
- Ensure that Foundation REST API, OpenText Documentum CM client, and Client REST API are configured to run in HTTPS mode.
- If you are using OpenText Documentum Content Management (CM) Branch Office Caching Services, ensure that the Branch Office Caching Services server is configured to run in HTTPS mode and CORS enabled.

2.2 Installing Documentum Client Manager

Documentum Client Manager acts as a content transfer plug-in. Documentum Client Manager is a browser extension-free implementation that uses web sockets to communicate with the application server.

There are two ways in which the Documentum Client Manager v2 application can be installed on the target machines:

- End user installation option
- IT Administrator push option

For more information about Documentum Client Manager, see *OpenText Documentum Content Management - Client Installation Guide (EDCCL250400-IGD)*.

2.2.1 End user installation

For the first time, when the end user is importing or saving a file into the Content Connect repository, they will be prompted to download and install the latest `DCMAppInstaller.msi` file (Documentum Client Manager) for the following applications:

- Content Connect with OpenText Documentum CM client (Outlook, Word, Excel, and PowerPoint)

Install the `DCMAppInstaller.msi` file and refresh the screen to use Content Connect.

Note: The Documentum Client Manager installation is not required if the current or latest version is installed as part of the OpenText Documentum CM client installation.

2.2.2 IT Administrator push

For Client REST API servers, administrators can push the installer on end user machines using System Center Configuration Manager (SCCM).

Run the `DCMAppInstaller.msi` (with Administrator privileges) located at `\Content_Connect\wsctf\` from a command prompt:

```
msiexec /i [Path to the installer.msi] MSIINSTALLPERUSER="{}" /qn
```

For more information on Administrator push on Client REST API server, see *OpenText Documentum Content Management - Client Installation Guide (EDCCL250400-IGD)*.

Note: The *`MSIINSTALLPERUSER="{}"`* parameter specifies that the installation will be per machine. The */qn* parameter specifies that the installation will occur quietly, bypassing the interface windows for installation.

2.3 Deploying Content Connect

After the prerequisites are completed, deploy Content Connect.

To deploy Content Connect:

1. Download and extract OpenText Documentum CM - Microsoft Integrations 25.4 from OpenText My Support. Extract the `Content_Connect_25.4` ZIP file.
2. Obtain x509 certificate to configure the Content Connect server to run in HTTPS mode.

   Note: x509 certificate is required for Microsoft O365 to allow the third-party add-in deployment. Certificates for Node JS must be in the `.pem` format.
3. In the command prompt, navigate to the Content Connect folder that has the extracted binaries.
4. Run the following commands to create the `.pem` and `.key` formats for Node JS:
   - `openssl pkcs12 -in <filepath\nameofthe file.pfx> -nocerts -out <filepath\key.pem> -nodes`
   - `openssl pkcs12 -in <filepath\nameofthe file.pfx> -nokeys -out <filepath\cert.pem> -nodes`
5. Use the following command to copy the certificate location and certificate name to the Content Connect database configuration file:

   ```
   node initialize.js certificate <keypath/key.pem> <certpath/cert.pem>
   ```

Example 2-1:

```
node initialize.js certificate C:/ssl/sslkey.pem C:/ssl/sslkey-
cert.pem
```

Note: Ensure that you specify the absolute path for the certificate and key files. You can use any filename for the certificate and key files.

6. Optional To set up Vault utility, see *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

   a. Use the following command to configure the Vault with Content Connect:

   ```
   node initialize.js configurevault <vaultenabled> <dsisEndPoint> <secretname>
   <keyname> <token>
   ```

   where:
   `<vaultenabled>` is used to enable or disable Vault. You must set the value as `true` to enable the Vault or set the value as `false` to disable the Vault.
   `<dsisEndpoint>` is the host URL of the DSIS service to connect to the Vault server.
   `<secretname>` is the secret name set in the Vault server. You must set the value as `CC_DBPASSWORD`.
   `<keyname>` is the unique key set in the Vault server. You must set the value as `dbpassword`.
   `<token>` is the token configured in DSIS service.

   b. Use the following command to check the status of the DSIS service:

   ```
   node initialize.js checkstatus
   ```

   Note: If the DSIS service is not running properly or if you see the error message, *Something went wrong with vault service*, see the DSIS service logs for more details.

7. Run the following commands from the command prompt:

   - **Initialize the database configuration**

   ```
   node initialize.js dbconfig <db-host> <db-port> <db-name> <dbusername> <db-
   password> <db-server-type>
   ```

   where:
   `<db-host>` is the IP address of the machine where the database is installed.
   `<db-port>` is the port for database. The default port for Microsoft SQL is 1433, PostgreSQL is 5432 and Oracle is 1521. However, the port can vary in different environments.
   `<db-name>` is the database name to be created.
   `<dbusername>` is the database user having privileges to create new database.
   `<db-password>` is the password for the database user.
   `<db-server-type>` can be set as `"postgres"` for PostgreSQL or `"mssql"` for Microsoft SQL or `"oracle"` for Oracle.

Example 2-2:

```
node initialize.js dbconfig localhost 1433 content_connect_db sa
manager mssql
```

- **SSL or TLS database configuration**

  Perform the following configurations when database is SSL enabled:

For PostgreSQL	Use the following command to configure the database:
	`node initialize.js dbconfigssl <ssl-enable> <ssl-ca> <ssl_mode>`
	where:
	<ssl-enable>, set it to true to enable the SSL database.
	<ssl-ca>, specify the absolute path of the root certificate.
	<ssl_mode>, specify the value for example verify-full.
	For example:
	`node initialize.js dbconfigssl true C:/certs/certificate.cer verify-full`

For Oracle	Before configuring the Oracle SSL database, you must perform the following prerequisite steps:
	– Get <wallet_name>.sso file from Oracle server. For example, cwallet.sso and place it in the Content Connect app server.
	– Copy tnsnames.ora file from Oracle server to the location: <content_connect>\node_modules\oracledb\build\Release\network\admin and update it in tnsnames.ora > MY_WALLET_DIRECTORY value. This is the absolute path of the wallet. For example, MY_WALLET_DIRECTORY value as C:\oracle\client_wallet.
	Use the following command to configure the database:
	`node initialize.js dbconfigssl <ssl-enable> <net_service_name>`
	where:
	<ssl-enable>, set it to true to enable the SSL database.
	<net_service_name>, specify the value as it is in tnsnames.ora file as a second parameter.
	For example:
	`node initialize.js dbconfigssl true net_service_name`

For Microsoft SQL	Use the following command to configure the database:
	`node initialize.js dbconfigssl <ssl-enable> <ssl-ca>`
	where:
	<ssl-enable>, set it to true to enable the SSL database.
	<ssl-ca>, specify the absolute path of the root certificate.
	For example:
	`node initialize.js dbconfigssl true C:/certs/rootcertificate.cer`

- **Create a database**

```
node initialize.js database
```

Notes

  - To configure Content Connect Database on Microsoft SQL Server, make sure to enable the SQL authentication.
  - When there is any issue with database connection or authentication, Admin can see a error along with the statements such as tables created and user created.

- **Create the System Admin and Business Admin users to access the Admin Console**

```
node initialize.js adminuser
```

This command creates the default users, System Admin, and Business Admin. These users will be created with user name and password.

Note: Ensure to note down the password for the users created after you run this command.

Administrators can change their login credentials in the Admin Console. For more information, see "Logging into the Content Connect Admin Console" on page 27.

- **Set the default Authentication Type for the Administration Console**

Available Authentication Types: OTDS, ct-OTDS, Basic, and Basic-ct.

  - OTDS/ct-OTDS: Make sure to add **clientid** and **OTDS server URL**.

```
node initialize.js adminservice <AuthType> <clientId> <otdsurl>
```

  - Basic/Basic-ct: The Content Connect admin services will use the Basic or Basic-ct authentication.

```
node initialize.js adminservice <AuthType>
```

When you set the default Authentication Type, the Admin Console configuration is synchronized with this setting and displays the set Authentication Type. You can change the Authentication Type from the Admin Console, if required.

Note: When you change the authentication configurations in the Admin Console, you must restart the Content Connect server.

- **Content Connect Application Registration with Microsoft Entra ID**

Go to App registrations

a. Click **New Registration**.

b. Specify the name of the application.

c. Click **Register**.

Go to Manage > Authentication

a. Select the **Accounts in this organizational directory only (Single tenant)**.

b. Click **Add a Platform**.

c. In Configure platforms, click **Web**.

d. Specify the **Redirect URIs**.

For example:

```
https://<cc server host>:<port>/cc/ui/templates/msgraphdialog.html
```

e. In **Implicit grant and hybrid flows**, select the following check boxes:

- **Access token (used for implicit flows)**
- **ID token (used for implicit and hybrid flows)**

f. Click **Configure**.

g. Go to **Overview** and copy the `Application (client) ID` and `Directory (tenant) ID`.

Go to Manage > Certificates & secrets

a. Click **New client secret**.

b. Specify the description in the **Description**, and click **Add**.

c. Copy the added value of the secret and save.

Go to Manage > API permissions

a. Click **Add a permission**.

b. Select the **Microsoft Graph**.

c. Select the **Delegated permissions**, and select the following delegated permissions:

```
Mail.read
offline_access
Mail.Read.Shared
Mail.ReadWrite
Mail.ReadWrite.Shared
```

d. Click **Add a permission**.

e. Administrator must select the **Grant admin consent** to grant a permission.

Go to Manage > Expose an API

a. Click **Add** next to the **Application ID URI**.

b. In the **Edit application ID URI** screen, under **Application ID URI**, specify the scope name as illustrated in the following format:

```
api://<cc server host>:<port>/<auto generated id>
```

For example:

```
api://banddqa902.otxlab:8443/988beae6-8c4b-46ab-9013-22f8e9033355
```

c. Click **Save**.

d. In the **Add a scope** > **Add a client application** section, for **Client ID**, add the following globally unique identifier (GUID):

- Microsoft Office:

  ```
  d3590ed6-52b3-4102-aeff-aad2292ab01c
  ```

- Office on the web:

  ```
  93d53678-613d-4013-afc1-62e9e444a0a5
  ```

- Outlook on the web

  ```
  bc59ab01-8403-45c6-8796-ac3ef710b3e3
  ```

8. Update the `MSGraphConfig.json` file using the following command:

```
node initialize.js graphconfig <client-id> <client-secret> <tenant-id>
```

Where,

<*tenant-id*>: A unique identifier of the Microsoft Entra ID instance. Copy the value from **Directory (tenant) ID** in Microsoft Entra admin center.

<*client-id*>: This identifier will be assigned when Seq is set up as an application in the directory instance of the Microsoft Entra ID. Copy the value from **Application (client) ID** in Microsoft Entra admin center.

<*client-secret*>: The secret key Seq will use when communicating with the Microsoft Entra ID instance. Copy the value from **Certificates & secrets** in Microsoft Entra admin center.

Example 2-3:

```
node initialize.js graphconfig 0d9e7f33-ed20-47cd-be4b-9df221edba31
VgM8Q~WrvSmkeYlcLSbRhiCuQTfSkUwI-BHEMarH 59f4deea-2d1c-4797-8ad1-54fcf2253b91
```

9. Applicable when only IPv4 is installed on the server machine. Run the following command:

```
node initialize.js updateprotocol ipv4
```

By default, Content Connect runs on the IPv6 protocol.

10. Run the following command to update the extension for Content Connect Server.

```
node initialize.js ccextension <extension name>
```

By default, the <*extension name*> is set as *cc*.

Note: If you change the `ccextension` value, make sure that you update the values during app registration in Microsoft Entra admin center and access the Content Connect Admin console using the updated value.

11. On OpenText Documentum Content Management (CM) Server, install the `Email_Attachments_Relation.dar` file located at `Content_Connect\customscripts\cc_dar_installer\dar`.

    Note: The DAR file installation is required to establish a relation between:

    - The email and its attachments.
    - The attachments when the parent email is not selected. In such instances, the first attachment is considered as the parent entity.

    The "Enabling OpenText Documentum CM relation between an email and its attachments" on page 36 section provides more details.

12. Ensure that the following ports are available in the Content Connect server:

    - 8443 for Content Connect.
    - 1607 for Content Connect to connect to the Admin Configuration service.

    Note: If the preceding ports are unavailable, you can also update the available ports. Perform the following step to change the ports:

    - Update port in the following command prompt in the *<ContentConnect_installdir>*`\gulpfile.js` file:

```
gulp.task('cc_webserver', function () {
  gulp.task('AdminConsole')();
  gulp.src(config.release)
    .pipe(webserver({
      https: {key: certData.server['ssl_key_path'], cert:
certData.server['ssl_cert_path']},
      port: '8443',
      host: '0.0.0.0',
      directoryListing: false,
      fallback: 'index.html',
      middleware: function (req, res, next) {
        return [cors(req, res, next)];
      }
    }));
});
```

    - Use the following steps to update the admin service port:

      – Update the admin service port in *<ContentConnect_installdir>*`\common\scripts\services\ConfigService.js`:

```
var configServicePort = 1607;
```

      – Update the admin service port in *<ContentConnect_installdir>*`\server\AdminConsole\configurations\general.json` file:

```
"server": {
        "port": 1607,
        "ssl_key_path": "<Absolute folder path>/key.pem",
```

```
        "ssl_cert_path": "<Absolute folder path>/cert.pem",
        "Allowed-Origins": "*",
        "purge_old_logs": "true"
        "secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "extension": "cc"
        "protocol": "IPV6"
                },
```

2.4 Running Content Connect as a background service

1. Open command prompt with Administrator privileges, go to the Content Connect build location (`%Content_Connect%`).
2. Run the following commands:

To start the service:

```
npm run start-service
```

To stop the service:

```
npm run stop-service
```

Note: If you want to restart the service, use the following command:

```
npm run restart-service
```

To delete the service:

```
npm run delete-service
```

To check the logs:

```
npm run logs
```

To check the status:

```
npm run status
```

To check the service-info:

```
npm run service-info
```

2.5 Post-deployment task

2.5.1 Licensing OpenText Documentum CM

OpenText Documentum CM uses OpenText Directory Services (OTDS) to apply licenses for all OpenText Documentum CM components. For more information about procuring the license file and configuring OTDS and license, see *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

Chapter 3

Upgrading Content Connect

1. Stop and remove the current Content Connect service: `service.bat remove`.
2. Take a backup of `general.json`, `service.json`, and `MSGraphConfig.json` files located at `%Content_Connect%/server/AdminConsole/configurations` on the Content Connect server.

   You can use this file for references to the existing data.
3. Download and extract the contents of the Content Connect binary.
4. Download and install the supported Node.js version. For more information on all the supported versions, see the product *Release Notes* on My Support (support.opentext.com).
5. Perform the following actions:

   a. From the back up location, open the `general.json` file.

   b. From the downloaded and extracted folder, navigate to `%Content_Connect%/server/AdminConsole/configurations` location and open the `general.json` file.

   c. From the back up `general.json` file, copy the following certificate reference values listed in the `server` attributes section and replace the respective values in the `general.json` file in the extracted folder.

   Example:

   ```
   "server": {
   "ssl_key_path": "<key location>",
   "ssl_cert_path": "<certificate location>",
   ```

   d. From the back up `general.json` file, copy the following database reference values listed in the `db` attributes section and replace the respective values in the `general.json` file in the extracted folder.

   Example:

   ```
   "db": {
   "host": "<host ip>",
   "port": "<port>",
   "username": "<username>",
   "password": "<password>",
   "dbname": "database name",
   "dbserver": "database server name",
   "ssl": {
               "enabled": "<false>",
               "ca": "",
               "sslconfig": ""
   },
   ```

Notes

- If the Vault is enabled, ensure the `password` value is blank in the `general.json` file for database.
- Verify that the port is set the same as in the `gulp.js` file. Refer to step 12 for changing the ports.

e. Update the `service.json` file based on the previously configured values. Refer to the `service.json` file in the backup location.

f. Update the `MSGraphConfig.json` file based on the previously configured values. Refer to the `MSGraphConfig.json` file in the backup location.

6. Start the Content Connect service: `npm run start-service`.

7. Download the latest manifest file from the Admin Console and update the manifest file on the corresponding Microsoft application (Microsoft Office 365 Admin center).

8. Install the `Email_Attachments_Relation.dar` file on Documentum CM Server located at `Content_Connect\customscripts\cc_dar_installer\dar`.

Note: This step is optional. The DAR file installation is required to establish a relation between:

- The email and its attachments.
- The attachments when the parent email is not selected. In such instances, the first attachment is considered as the parent entity.

The "Enabling OpenText Documentum CM relation between an email and its attachments" on page 36 section provides more details.

Note: To know if the upgrade is successful, log in to the Admin Console and check the version in the header.

3.1 Post-upgrade task

3.1.1 Licensing OpenText Documentum CM

OpenText Documentum CM uses OpenText Directory Services (OTDS) to apply licenses for all OpenText Documentum CM components. For more information about procuring the license file and configuring OTDS and license, see *OpenText Documentum Content Management - On-Premises Upgrade and Migration Guide (EDCCS250400-UMD)*.

Chapter 4

Configuring Foundation REST API or Client REST API

4.1 Configuring Tomcat

Disable TLS v1.0 and v1.1 for Foundation REST API or Client REST API port 443 at the Tomcat level.

Add the SSLProtocol setting to the connector configuration in `$CATALINA_BASE/conf/server.xml` and restart Tomcat.

```
<!< Define an SSL Coyote HTTP/1.1 Connector on port 443 <>
<Connector
protocol="org.apache.coyote.http11.Http11AprProtocol"
port="443" maxThreads="200"
scheme="https" secure="true" SSLEnabled="true"
SSLCertificateFile="/usr/local/ssl/server.crt"
SSLCertificateKeyFile="/usr/local/ssl/server.pem"
SSLVerifyClient="optional" SSLProtocol="TLSv1.3+TLSv1.2"/>
```

Note: To mitigate security vulnerabilities, install or configure a default application to hide the Tomcat welcome page. Additionally, configure a default error page that does not include the Tomcat server version number. You can configure the error page by repackaging the `CATALINA_HOME/server/lib/catalina.jar` file with the updated `ServerInfo.properties` file.

4.2 Configuring authentication

You can configure Content Connect authentication with Foundation REST API or Client REST API.

To authenticate Content Connect with Foundation REST API or Client REST API:

1. Add the following tags in the `rest-api-runtime.properties` file located at *<application-server>*`\webapps\`*<dctm-rest>* or *<D2-rest>*`\WEB-INF\classes`. For details, see *OpenText Documentum Content Management - Foundation REST API Development Guide (EDCPKRST250400-PGD)*.

```
rest.cors.enabled=true
rest.cors.allowed.origins=https://<contentconnect url>:<port>
rest.cors.allowed.methods=GET, POST, PUT, DELETE, OPTIONS, HEAD
rest.cors.allowed.headers=DOCUMENTUM-CUSTOM-UNAUTH-SCHEME, Authorization, Content-
Type, Accept, X-CLIENT-LOCATION, X-CLIENT-APPLICATION-NAME, OT-DCTM-PRODUCT-CODE, x-
no-redirection
rest.cors.exposed.headers=Accept-Ranges, Content-Encoding, Content-Length, Content-
Range, Authorization, Content-Disposition
rest.dql.access.mode=all
rest.security.client.token.cookie.samesite=None; Secure; Partitioned
rest.security.csrf.enabled=true
rest.security.csrf.http_methods=POST,PUT,DELETE
```

```
rest.security.csrf.generation.method=server
rest.security.csrf.header_name=DOCUMENTUM-CSRF-TOKEN
rest.security.csrf.parameter_name=csrf-token
rest.security.csrf.token.length=256
rest.security.auth.mode=<auth-mode>
rest.should.parse.emails=true
```

where, `rest.cors.allowed.origins` specifies the Content Connect Admin Console URL and Microsoft Exchange Server URL.

The following table provides descriptions for certain tags:

Tag	Description

`rest.security.csrf.enabled=true`
`rest.security.csrf.http_methods=POST,PUT, DELETE`
`rest.security.csrf.generation.method=server`
`rest.security.csrf.header_name=DOCUMENTUM-CSRF-TOKEN`
`rest.security.csrf.parameter_name=csrf-token`
`rest.security.csrf.token.length=256`
	This enables the Cross-Site Request Forgery (CSRF) token.

`rest.security.auth.mode=<auth-mode>`	Used to set the authentication mode. Supported authentication modes: • basic • basic-ct • otds_token • ct-otds_token **Note:** The authentication mode value must be same as specified in Admin Console.
`rest.should.parse.emails=true`	This setting helps the email aspect attributes to be mapped to the Documentum object when an Outlook user imports an email from Content Connect.
`rest.dql.access.mode=all`	This mode provides the backward compatibility with the DQL resource, where any user would be able to access the DQL resource. The property `dql.disallowed.types` holds good only when the access mode is configured to `all`.

2. To ensure that the non-admin users perform successful operations, remove the following values from the `dql.disallowed.types` parameter in the `rest-api-runtime.properties` file:

   - dmr_content
   - dm_user

- dm_relation
- dm_format
- dm_type
- dmi_type_info

3. For Smart View, configure the following tags in the `rest-api-runtime.properties` file located at *<application-server>*`\webapps\`*<D2-Smartview>*`\WEB-INF\classes`:

```
rest.security.client.token.cookie.samesite=none
rest.security.client.token.session.cookie=true
rest.security.headers.csp.allowed_frame_ancestors=self https://
*.sharepoint.com https://outlook.office.com https://*.officeapps.live.com https://
outlook.office365.com <Contentconnect Admin Console url:port>
```

4. Restart Foundation REST API or Client REST API application server and Smart View application server if modified. For more information, see *OpenText Documentum Content Management - Foundation REST API Development Guide (EDCPKRST250400-PGD)*.

Notes

- To support aspect attributes in OpenText Documentum CM client, configure the O2 in OpenText™ Documentum™ Content Management client configuration.
- Branch Office Caching Services is not supported when working with .eml or .msg type files. Hence, the email aspect attributes work only when it is not enabled for Branch Office Caching Services.
- Basic authentication is not recommended for production servers. It is recommended for users to choose OTDS authentication for greater security.

4.3 Configuring OTDS authentication

If you are using OTDS authentication, then complete the following tasks:

- Configure OTDS with Documentum CM Server. *OpenText Documentum Server Administration and Configuration Guide* provides more details.
- Configure runtime properties in Foundation REST API.
- Configure the OAuth clients and system attributes in OTDS.

4.3.1 Configuring runtime properties

The **ct-otds_token** authentication mode is supported in Foundation REST API when integrating OTDS with Documentum CM Server using token-based authentication.

To configure Foundation REST API properties for OTDS, see the *OpenText Documentum Content Management (CM) Foundation REST API* documentation. When using client with Content Connect, see *OpenText Documentum Content Management - Client Installation Guide (EDCCL250400-IGD)* to configure Client REST API properties for OTDS.

4.3.2 Configuring OAuth clients on OTDS for Content Connect

1. Login to OTDS server (`https://<otds_ip>:<otds_port>/otds-admin`) with the administrator credentials.
2. In OTDS Admin, click **OAuth Clients** > **Add**.
3. In the **General** page of the **New OAuth client** wizard, specify the following details:
   - Specify a unique name for **Client ID**.

     For example: CC_Client.
   - Select the **Confidential** check box.
4. Click **Next** and navigate to **Redirect URLs** section.
5. Click **Add** and specify the Content Connect URL. For example: `https://<Content Connect server>:<port>`
6. Click **Save**.
7. Add the following sites to the **Trusted Sites** section:

```
1 https://<ContentConnectServer:port>
2 https://*.sharepoint.com
3 https://outlook.office.com
4 https://*.officeapps.live.com/
5 https://outlook.office365.com
```

8. Save the configuration.

Chapter 5

Configuring Content Connect

Log into the Content Connect Admin Console to configure an endpoint URL for the REST server and perform other configurations. In the Admin Console, you can modify user settings, view log entries, adjust parameters to track Content Connect activity, and configure object types.

The Admin Console has pre-configured default values for all configurations other than Endpoint URLs and Outlook REST URL. If you want to update these values, then refer to the following sections for more information.

Notes

- After configuring Content Connect, when the user logs in to the Content Connect application for the first time, a dialog box is prompted to know their geographical location. The user details such as location, IP address, application name is shared with Documentum CM Server for tracking and auditing purposes.
- The audit logs for the desktop applications will not include the geographical location.

5.1 Logging into the Content Connect Admin Console

In the Admin Console, you can set up, change configuration, log settings, and view log records.

To log into the Admin Console:

1. Go to <*https://<NodeJS server host address>:<port>/<ccextension>/admin*> to open the Content Connect Admin Console.

   where,

   - <*NodeJS server host address*>: Host address of the Content Server
   - <*port*>: Admin Console by default uses 8443 port.
   - <*ccextension*>: Content Connect by default uses `cc` and is initialized in step 10. If you change the default value, make sure that you use the relevant ccextension name.

   Example:

   ```
   https://CCserver.com:8443/<ccextension>/admin
   https://CCserver.com:8443/cc/admin
   ```

2. Type administrator user name and password, and select **Login**.

Note: For the first-time log in, enter the credentials for System Admin user or for Business Admin user. Login with the same password while creating the users. On first time login, you are prompted to change the password. You can then change your password. To change the password, hover over the **User** name and click **Change Password**.

Important

If you enter an incorrect password three consecutive times, the system locks your account. In such cases, you must update the password as per the new encryption policy. From the command prompt, Content Connect build location, run the following command to update the Administrator's password:

```
node initialize.js updatepassword <username> <password>
```

Example: `node initialize.js updatepassword SystemAdmin SystemAdmin@123`

After the command is run successfully, log in to the Admin Console using the changed password. Administrator is prompted to change the password from the Admin Console interface.

- **Configuration** tab: Configure settings for endpoint type and URL, OpenText Documentum CM client interfaces, the authentication type, the log level, email attachments, set thresholds, download manifest files according to the selected REST endpoint, and perform any other configurations.
- **Folder view settings** tab: Restrict the users to import and navigate documents into the predefined folder path or search for a document within a repository.
- **Logs** tab: View, filter, and download log entries.
- **Types** tab: Configure OpenText Documentum CM object types against repositories.

5.2 Updating product name for end user auditing

When the user logs in to the Content Connect application for the first time, the user is prompted to share their system's geographical location with the Documentum CM Server. The user details such as, location, IP address, product display name is shared with Documentum CM Server for auditing purposes.

To update the product name as per the display name in the manifest file:

1. In the Admin Console > **Configuration** tab, specify the product name in the **Product Name** field. By default, the product name is `ContentConnect`.

Note: The Content Connect display name that is provided in the manifest file must match with the product display name to fetch successful auditing results.

2. Select **Save**.

5.3 Configuring the authentication in Admin Console

In the Admin Console > **Configuration** tab, select one of the following types of authentication required for users to log into Content Connect:

- **Basic** – Requires users to enter their user name and password when they log in.

  Note: Basic authentication requires Content Connect to transmit and store user credentials in reversible format on the user's machine. Basic authentication is not recommended for production servers.

- **OTDS** – This is OpenText Directory Services (OTDS) SSO authentication. By default, the **OTDS** authentication type is selected.
- **Basic-ct** – This is client token authentication type.
- **ct-OTDS** – This is client token OpenText Directory Services (OTDS) SSO authentication.

Note: To use Smart View, you must configure the Basic-ct and ct-OTDS authentication.

To configure Basic or Basic-ct authentication:

1. In the Admin Console > **Configuration** tab, select **Basic** or **Basic-ct** in the **Authentication Type** box.
2. In the **Endpoint Type** field, update the endpoint URL, if necessary, to support the type of authentication.
3. Select **Save**.
4. Ensure that you have configured Foundation REST API or Client REST API as per "Configuring authentication" on page 23.
5. Select **Test** for the Endpoint URL. If a popup message confirms success, then the endpoint has been configured correctly. Ensure to have the same authentication mode as set in the selected endpoint.

To configure OTDS or ct-OTDS authentication:

Before proceeding with this configuration, ensure that you have completed the configuration in "Configuring OTDS authentication" on page 25.

1. In the Admin Console > **Configuration** tab, select **OTDS** or **ct-OTDS** in the **Authentication Type** box.
2. Configure the following fields:

   - **OTDS Server**: URL of OTDS service, example: `https://<your server>:<port>/otdsws`

- **OTDS Client ID**: Client ID Name provided in "Configuring OAuth clients on OTDS for Content Connect" on page 26.

3. Click **Save**.

   Admin Services needs to be restarted once the authentication mode is updated.

5.4 Configuring an endpoint URL

After Content Connect has been deployed on your system, you must provide a URL for the endpoint server and then test the configuration.

To configure an endpoint URL:

1. In the Admin Console > **Configuration** tab > **Rest endpoints** section, select **Documentum-REST-Server** or **Documentum-D2-REST-Server**.

   - If you select **Documentum-REST-Server**, enter the endpoint URL in the **REST URL** field.
     For example: `https://<Rest-Server:Port/dctm-rest>`.
   - If you select **Documentum-D2-REST-Server**, enter the endpoint URL in the **REST URL** field and the **D2 Classic / D2 Smart View URL** based on the client interface chosen. For details, see "Configuring OpenText Documentum CM client interfaces" on page 34 section.

     For example: `https://<D2-Rest-Server:Port/D2rest>`
     `https://<D2-client-server:Port/D2>`

2. Select **Save**.

5.5 Configuring Microsoft Outlook email configuration

You can either copy or move emails from Microsoft Outlook during an import operation. You can also validate the checksum value of the email with the move operation.

- In the Admin Console > **Configuration** tab > **Outlook Settings** section > **Outlook email configuration** list:
  - Select **Copy** to create a copy of the email in the OpenText Documentum CM repository while performing the import operation. It retains the email in Microsoft Outlook. By default, the **Copy** option is selected.
  - Select **Move** to create a copy of the email in the OpenText Documentum CM repository while performing the import operation. It moves the email to the **Archive** folder in Microsoft Outlook if checksum is not enabled.
- In the Admin Console > **Configuration** tab > **Outlook Settings** section, enable **Checksum**.

Note: Make sure that you have enabled checksum on Documentum CM Server before it is enabled in the Content Connect.

If the checksum is not enabled in the Documentum CM Server, you must create a new Filestore in the Documentum CM Server to use the checksum feature.

Make sure to set the new Filestore as the default Filestore, else the new content will be saved in the old Filestore.

For more information, *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)*.

Content Connect then checks if the checksum value of the email matches with the checksum value of the email after importing into the repository. By default, the **Checksum** check box is not selected in Content Connect.

- If the value matches, then the import is successful and the emails are moved to the **Deleted Items** folder in Microsoft Outlook.
- If the checksum value does not match, then the move operation is cancelled. This feature is not supported with the copy operation.

Note: Email aspect attributes are supported only for the .MSG and .EML email formats.

5.6 Enabling Content Connect for Web version of Microsoft applications

Enable users to use Content Connect to access the web version of Microsoft applications such as, Word, Excel, and PowerPoint.

Note: When users open a document in the web version of Microsoft Office applications, Microsoft automatically saves the document to OneDrive.

To enable Content Connect for Microsoft applications:

1. In the Admin Console > **Configuration** tab > **Office Settings** section, select the **Enable Content Connect Add-in for Office Online** check box.

   By default, this check box is not selected.

2. Select **Save**.

Note: Configuration changes made in the Admin Console take effect only when users log out and then log in, or when the browser cache is cleared.

5.7 Configuring Branch Office Caching Services

Use Content Connect with the Branch Office Caching Services Server with Documentum CM Server. You must do the following configurations to use Content Connect with the Branch Office Caching Services server.

- Ensure that the Branch Office Caching Services server is configured for SSL communications. For more information, *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.
- In the Admin Console > **Configuration** tab > **BOCS Settings** section, select the **Enable network location selection** check box to route the content transfer through a Branch Office Caching Services server. By default, this option is not selected.

  Notes

  – When the **Enable network location selection** check box is not selected and if the Branch Office Caching Services locations are configured with the predefined IP Address Range, then this import will also be routed through the Branch Office Caching Services server. However, the content upload automatically picks the network location that falls within the defined IP Address Range.

  – If the network location is not within the defined IP Address Range, then Asynchronous content transfer mode is chosen.

  – If multiple network locations are configured on the Branch Office Caching Services servers, then the user will get an option to select the preferred network location before the user logs into the repository.

- Add the following `response-header` tag within the `filters` tag in the `standalone.xml` file located at `%Tomcat_HOME%\server\DctmServer_BOCS\configuration` on the Branch Office Caching Services server:

```
<response-header name="Access-Control-Allow-Headers" headername="Access-Control-Allow-Headers" header-value="accept,authorization, content-type, x-requested-with, DOCUMENTUM-CUSTOMUNAUTH-SCHEME "/>
```

  Note: Ensure that you restart the Branch Office Caching Services server after you update the `standalone.xml` file.

- If Branch Office Caching Services is configured with Asynchronous content transfer mode, ensure that the **Content Transfer mode (Asynchronous)** check box is enabled on the **Configuration** tab of the Admin Console. By default, this check box is selected.
- If the Branch Office Caching Services server is configured and the services are not running on the Branch Office Caching Services server then the Import operation fails from Content Connect. To identify such issues, view the logs in Admin Console to troubleshoot the issue.

5.8 Configuring client side validation

To configure client side validation:

1. In the Admin Console > **Configuration** tab > **Client Side Validation** section, enable the **Regular Expression for Validation** check box.

   By enabling this configuration, the value of text fields in the Content Connect application during import are validated against Regular Expression.

   For example: ^[A-Za-z0-9!$&'()*+,;=._~:@/?%\s-]*$

2. Click **Save**.

   Note: This setting is not applicable for OpenText Documentum CM client users.

5.9 Configuring the Foundation Java API date format

You can import emails or documents with custom date attributes.

Note: This setting is not applicable for OpenText Documentum CM client users.

To configure the Foundation Java API date format when the value of rest.should.parse.emails is set to true:

Note: By default, the `rest.should.parse.emails` attribute value is set to *`false`*.

1. In the Content Connect Admin Console > **Configuration** tab > **DFC date format** field, the default date format that Content Connect supports is populated. You can modify the Foundation Java API date format based on the supported formats for Foundation Java API. For the list of supported Foundation Java API date formats, see *OpenText Documentum Content Management - Foundation Java API Development Guide (EDCPKCL250400-DGD)*.

   Note: When using Foundation Java API supported date formats, use M/MM for month, D/DD for day, YY/YYYY for year, h/hh for hour, m/mm for minute, s/ss for seconds, and a for am/pm (case sensitive tokens). Default format is 'M/D/YYYY h:mm:ss a'.

2. Click **Save**.
3. Restart the Foundation REST API App server.
4. Clear the cache and reload Content Connect Admin Console.
5. Logout and login to Content Connect to import emails or documents with custom date attribute.

5.10 Enabling proxy server to connect to the outbound portals

Enable this configuration to route the outbound request through the proxy server. This configuration is applicable only for SSO with Microsoft Identity Platform. For more information on enabling SSO with Microsoft Identity Platform, see "Deploying Content Connect" on page 9.

To enable proxy server to connect to the outbound portals:

1. In the Admin Console > **Configuration** tab, enable the **Enable proxy server** check box.
2. Provide the host and the port details.

   Example: `http://<host>:<port>`
3. Click **Save**.

5.11 Configuring OpenText Documentum CM client interfaces

Depending on the OpenText Documentum CM client setup in your organization, choose from the following two interfaces:

- **Classic View**: The traditional, widget-based OpenText Documentum CM client interface. By default, the Classic View interface is enabled (The **D2 Smart View** check box is disabled).

  To enable Classic View in Content Connect:

  1. In the Admin Console > **Configuration** tab, select the required **Authentication Type**.
  2. Provide the Foundation REST API Server URL in the **REST URL** field.
  3. Provide the Classic View endpoint URL for the Foundation REST API Server in the **D2 Classic / D2 Smart View URL** field.

     For example: `https://<D2-Rest-Server:Port/D2rest>`
     `https://<D2-client-server:Port/D2>`
  4. Clear the **D2 Smart View** check box, if selected.
  5. Click **Save**.
  6. Add the parameters listed for Classic View in the "Configuring authentication" on page 23 section in the `rest-api-runtime.properties` file located at `%TOMCAT-HOME%\webapps\<D2-rest>\D2-Smartview\WEB-INF\classes`.
- **Smart View**: Uses the OpenText tile-based Smart View user interface.

Note: Content Connect is not certified with Smart View and OpenText Documentum Content Management (CM) client Branch Office Caching Services combination.

To enable Smart View in Content Connect:

1. In the Admin Console > **Configuration** tab, select the required **Authentication Type**.
2. Provide the Client REST API URL in the **REST URL** field.
3. Provide the Smart View endpoint URL in the **D2 Classic / D2 Smart View URL** field.

   For example: `https://<hostname>:<port>/D2-Smartview`
4. Select the **D2 Smart View** check box.
5. Click **Save**.
6. Add the parameters listed for Smart View in the "Configuring authentication" on page 23 section in the `rest-api-runtime.properties` file located at `%TOMCAT-HOME%\webapps\<D2-rest>\D2-Smartview\WEB-INF\classes`.

5.12 Configuring Brava for preview

In the Admin Console > **Configuration** tab, provide the **Brava Server Host URL** to view documents or emails.

For example: *https://<Brava server:Port/BravaServer>*

5.13 Configuring the import of email attachments

Content Connect lets users import emails into a repository. Attachments are always imported with the email, but you can enable users to select or clear email attachments to import copies of selected attachments as separate files.

Notes

- Attachment separation is enabled by default.
- For OpenText Documentum CM client, attachment separation is enabled using linked document configuration in client configuration.

To enable separate import of email attachments:

1. In the Admin Console > **Configuration** tab > **Outlook Settings** section, select the **Extract attachments as separate objects** check box.
2. Select **Save**.

5.14 Enabling OpenText Documentum CM relation between an email and its attachments

When you import an email with attachments and if the attachment separation is enabled in the Admin Console, you can establish a relation between the email and its attachments (parent and child relation respectively) or between the attachments when the parent email is not selected.

This feature supports only:

- Foundation REST API flows.
- Single email and attachment import.
- Multiple email and attachment import.

Important

- On Documentum CM Server, you must install the `Email_Attachments_Relation.dar` file located at `Content_Connect\customscripts\cc_dar_installer\dar` to enable this feature.
- This DAR file must be installed on each repository that is used for Content Connect, including the global repository.

To install the `Email_Attachments_Relation.dar` file on Documentum CM Server, follow the instructions provided in the *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)*.

Notes

- OpenText recommends to run the `dardeployer.exe` file as an administrator and install the `Email_Attachments_Relation.dar` file. The `dardeployer.exe` file can be found at the Documentum CM Server installation location. For example: `C:\Documentum\product\<version>\install\composer\ComposerHeadless\dardeployer.exe`
- After the successful import, run the following DQL query from a OpenText Documentum CM client to retrieve the relationship:

```
select * from dm_relation where relation_name = 'content_connect_email_attachment'
```

5.15 Configuring threshold parameters for multiple files bulk import

Configure a threshold value to import multiple files using bulk import feature for OpenText Documentum CM client in the **Outlook Settings** section > **D2 multi file bulk import threshold** field. By default, the threshold value is set to `0`.

The threshold value distinguishes the import method (OpenText Documentum CM client multiple file import or OpenText Documentum CM client folder structure import) based on the number of files being imported.

- **OpenText Documentum CM client multiple files import**: Supports when the number of files being imported is within the threshold value. In such cases, the user can provide the file properties individually to each file. This approach is recommended for small number of files being imported.
- **OpenText Documentum CM client folder structure import**: Supports when the number of files being imported exceeds the threshold value or when the threshold value is set to 0. In such cases, the properties are inherited from the OpenText Documentum CM client folder structure configuration in OpenText Documentum Content Management (CM) client configuration. This approach is recommended for large number of files being imported. Folder structure import supports client configuration for O2 and attachment separation.

  The *OpenText Documentum Content Management - Classic View User Guide (EDCCL250400-UGO)* provides more information on the OpenText Documentum CM client folder structure import.

Notes

- To import files or emails into a folder, users must have the WRITE permission:
  - on the selected folder.
  - on both the selected and the destination folders when **Autolinking** is enabled.
- The **D2 multi file bulk import threshold** field value must not exceed the **Bulk Import Threshold** value.

  The **D2 multi file bulk import threshold** can contain only integer values.

5.16 Configuring bulk import parameters

Import multiple emails simultaneously. In the Admin Console > **Configuration** tab > **Outlook Settings** section, specify the maximum number of emails that you can import simultaneously in the **Bulk Import Threshold** field. By default, the threshold value is set to `100`.

Note: The Bulk Import threshold can contain only integer values.

You can rename the bulk import folder in Microsoft Outlook using the **Bulk Import Folder Name** in the Admin Console > **Configuration** tab > **Outlook Settings** section. The bulk import folder is automatically created in the Inbox after the user clicks the Content Connect icon in any mail in Microsoft Outlook. By default, the **Bulk Import Folder Name** value is `ContentConnect`.

5.17 Configuring repositories with predefined folder paths

Use the **Settings for folder navigation** setting to restrict the users from importing documents and emails into the predefined location in a repository. You can add multiple repositories and the corresponding locations. Users can select among the accessible folder paths to save, import, or search the documents. This setting is not applicable for OpenText Documentum CM client users.

To configure repositories with predefined folder paths:

1. In the Admin Console > **Folder view settings** tab, enable the **Settings for folder navigation** check box.

   The repository information table is displayed.

2. In the repository information table, specify the following information:

   - **Repository name**: Specify the configured repository name. This field is mandatory.
   - **Import folder path**: Folder path is the location, where the user can import or save the documents or emails. This field is mandatory. You can set only one import folder path with one repository.

     Example: `/<cabinet name>/<folder>/<folder>`
   - **Search path**: Limits the search feature for a user within the defined repository location. This field is optional. If this field is left blank, then the user can search for a document across the repository.

     Example: `/<cabinet name>/<folder>/<folder>`

     Note: OpenText recommends to configure the same path for **Import Folder path** and **Search path** per repository.

3. Click the + icon to add more repositories.

Note: Click the x icon against each repository row to delete the repository.

4. Click **Save**.

5.18 Configuring and viewing logs

Note: While troubleshooting the issues, ensure to check the log details in both Content Connect Admin Console log entries and the Content Connect services log file entries.

5.18.1 Content Connect logs

Modify Content Connect log settings and view log entries to track issues. To view only specific log entries, apply a filter. You can also download logs for printing or viewing.

To configure log levels:

1. In the Admin Console > **Configuration** tab:

   - Select one or more **Log Levels** (Error, Warning, Information, and Debug) to set the type of events that will be logged. By default, the **Error**, **Warning**, and **Information** check boxes are selected.
   - Specify the number of days after which the logs must be purged in the **Automatically purge logs older than __ day/s** field. By default, the value is 1. Minimum value = 1. Maximum value depends on the database capacity.

     Notes

     – To avoid impact on Content Connect client performance, it is recommended to purge the log data in the database frequently.

     – Only positive integer values are applicable to purge logs.

2. Select **Save**.

To view, filter, and download log entries:

1. Select the **Logs** tab to display the current page of log entries. Scroll through pages and view previous and next pages.
2. Select one or more items in the list of log parameters, or enter the text in the text box, and select **Apply Filter** to display only specific log entries.
3. Select **Download Logs** if you want to download the current page of log entries you are viewing.

5.18.2 Content Connect services logs

Modify Content Connect services log settings and view log entries to track issues.

To configure log data for Content Connect services:

1. Open the `general.json` file located at `%Content_Connect%/server/AdminConsole/configurations` on the Content Connect server.
2. In the **logger** > **file** section, configure the following:
   - *`level`*
     - *`info`*: Contains both information and error logs. Default level.
     - *`error`*: Contains only error logs.
   - *`filename`*: *`error.log`*: Contains the log data based on the selected level.

     Default file name. You can modify the file name.

To view log entries:

- Open the *`error.log`* file located at `%Content_Connect%/server/AdminConsole` on the Content Connect server.

5.19 Configuring object types

Use the Admin Console to configure object types for importing emails and documents. Create a single group or multiple groups and add repositories and object types as required. Groups allow you to manage multiple object types across repositories.

Custom object types can be created using Documentum Composer. For custom attributes to display in Content Connect, in Documentum Composer, from the **Display** tab, select the **Display Configuration** > **Scope** as `webtop` and pull all the required attributes into the **Attributes in Display Configuration** panel.

Content Connect supports **Query based Value Assistance** and **Fixed List** object types. For more information on configuring object types, see *OpenText Documentum Content Management Composer* documentation.

To configure object types, click **Import types masterlist** to import the JSON file containing the master list with repository names and their object types. A sample JSON file is provided with the build at: `<Content_Connect_location>/admin/types-masterlist-sample.json`

Sample `types-masterlist-sample.json` file:

```
{
    "repositories":
    [
        {
            "repo_name":"Repo_1",
```

```
        "types":
        [
            {"type":"dm_document", "name": "DM Document"},
            {"type":"cx_type1", "name": "custom type 1"},
            {"type":"cx_type2", "name": "custom type 2"}
        ]
    },
    {
```

Note: In this file, update the relevant details for your deployment.

Create multiple groups with single repositories, or a single group with multiple repositories. When you create a group with multiple repositories, ensure that the same object types are configured for all selected repositories. If the attachment separation feature is enabled, make sure that you configure object types, including dm_document.

To configure object types:

1. In the Admin Console > **Types** tab, select **Create Group** and enter a name and description for the group.
2. Select **Add Repositories** to display a list of available repositories.
3. Select one or more repositories from the list, and select **Add**. Their associated types appear below the repository name.
4. Click **+** to add multiple types. Types are added in the same sequence as in the master list of the JSON file.

   Notes

   - When you configure a single object type, while importing an email with attachments, the user can apply the specified metadata for the email to all the selected attachments at once using the **Apply to all attachments** button.
   - When you configure two or more object types, while importing an email with attachments (emails, documents, or images), the user can apply the specified metadata for the parent email to all the selected emails at once using the **Apply for all emails** button and to the selected document attachments using the **Apply for all attachments** button.
5. Beside each type are two check mark icons labelled **Document Creation Type** and **Email Type**. If you do not want one or more types to be available for users to create documents, or emails, or both, then click on the appropriate icons to clear them from the list.

   Notes

   - If object types are not configured, then only dm_document is available by default for import.

- If either **Document Creation Type** or **Email Type** is configured, then the other object type is set to **dm_document** by default for import.

6. Select **Save**. Select **Delete Group** to remove the group at any time.

Note: If new object types are added to a repository, you need to import the master list file containing the new types to replace the current master list.

Chapter 6

Configuring the preferred language

Content Connect supports only the following languages:

- English: EN-US
- German: DE-DE
- Spanish: ES-ES
- French: FR-FR
- Chinese (Simplified): ZH-CN
- Hebrew: HE-HE
- Hebrew(Israel):HE-IL
- Brazilian Portuguese
- Italian: IT-IT
- Arabic: AR

To configure the preferred language for Admin Console:

- Configure the preferred language in the Browser settings before you log in to the Admin Console.

To configure the preferred language for clients:

- If you are using Microsoft Office web client (Microsoft Office 365), then configure the preferred language and the country in the web version of Microsoft Office. For more information, refer to the *Microsoft documentation*.

  Note: For OpenText Documentum CM client, configure the preferred language in the browser.

- If you are using Microsoft Office desktop client, then configure the preferred language in **File** > **Options** in Microsoft Office.

  Note: For OpenText Documentum CM client, configure the preferred language in Windows.

Chapter 7

Configuring additional OpenText products with Content Connect

This chapter provides information to configure other OpenText products to work with Content Connect server.

7.1 Configuring OpenText Documentum CM client

If you are using OpenText Documentum CM client with Content Connect, then ensure that you configure the following settings in OpenText Documentum CM client before you start Content Connect:

- Configure OpenText Documentum CM client to run in HTTPS mode with a trusted certificate issued by CA.
- Configure OpenText Documentum CM client to run in the WSCTF mode. Open the `\\webapps\D2\WEB-INF\classes\settings.properties` file and update the following entry:

```
browser.plugin.mode =wsctf
```

- Add the following in the OpenText Documentum CM client `settings.properties` file:

```
csp.header.value = frame-ancestors <Rest Server URL> <Office/SharePoint URLs>
https://*.officeapps.live.com/ https://outlook.office365.com https://
<contentconnect url>:<port>
```

  Example:

```
csp.header.value = frame-ancestors https://restserver:8444 https://
xxx.sharepoint.com https://outlook.office.com https://*.officeapps.live.com/
https://outlook.office365.com https://contentconnect:8443
```

  Note: The URL ports or the values are separated by space.

- In client configuration, select **Tools** > **Options** > **Allowed actions in URL**. Select the following options to enable intelligent URL actions:
  - D2_ACTION_CONTENT_IMPORT
  - D2_ACTION_DISPLAY_DIALOG
  - D2_ACTION_CONTENT_CHECKIN
  - D2_ACTION_FOLDER_CREATE
  - D2_ACTION_CONTENT_IMPORT_STRUCTURE
  - D2_ACTION_CONTENT_VIEW
- Modify the following files:

1. Edit `ESAPI.properties` (`D2\WEB-INF\classes`) and update `HttpUtilities.ForceSecureCookies` to `true`;
2. For Tomcat, create `context.xml` with the following content and place the file in the OpenText Documentum CM client folder: `D2\META-INF`.

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
 <CookieProcessor sameSiteCookies="None" />
</Context>
```

- Enable breadcrumb in Doclist widget to use intelligent URL using client configuration.
- Configure OpenText Documentum CM client to import the folder structure in client configuration.
- To allow users to edit previous versions of documents with OpenText Documentum CM client, versioning must be enabled in client configuration.
- When multiple browser widgets are configured for Classic View, select the default browser widget. The Startpath configured in this default browser widget is considered in Content Connect.
- When multiple Doclist widgets are configured for Smart View, select the default Doclist widget. Add this Doclist widget in the Smart View landing page > `sample.xml` file and import. The Startpath configured in this default Doclist widget is considered in Content Connect.

  For example:

```
<doclist-widget>svdoclist</doclist-widget>
```

- When Autolink is enabled, the location must be specified within the Startpath configured in client configuration.
- When the Startpath is configured in client configuration, run the following command in the browser to refresh the cache and for changes to reflect immediately in Content Connect:

  `<D2 server url:port>/D2/servlet/LoadOnStartup?_docbase=testenv&_username=<username>&_password=<password>&propagate=true`

- When Startpath is configured for Smart View in client configuration and Smart View is enabled for Content Connect Admin Console, the Content Connect landing page reflects the configured Smart View Startpath.

Autolink and O2 transfer are supported with Content Connect. Attachment separation is supported for single and multi object import and folder structure import using Content Connect.

Refer to the *OpenText Documentum Content Management - Client Configuration Guide (EDCCL250400-AGD)* for more information.

7.1.1 OpenText Documentum CM client application evaluation mode

There are requirements to use Content Connect with OpenText Documentum CM client in evaluation mode:

- *LoadOnStartup* must be defined in the Client REST API and OpenText Documentum CM client `D2FS.properties` file to enable or disable evaluation mode.
- in client configuration, ensure the following is true for the application parameter:
  - The value is not empty.
  - The value is unique.
  - There are no spaces.

For more information about setting up application evaluation mode, see *OpenText Documentum Content Management - Client Configuration Guide (EDCCL250400-AGD)*.

7.2 Configuring facets from Documentum xPlore for repository searches

Content Connect provides predefined facets that are by default configured in Documentum xPlore. Facet configuration in Documentum xPlore is a prerequisite for obtaining search results in Content Connect. You can also configure custom facets from Documentum xPlore in Content Connect.

To configure Content Connect to use custom facets from Documentum xPlore:

1. Configure the `facetconfiguration.xml` configuration file located at `%Content-Connect-Server%\common\configurations` on the Content Connect server. Add `facet-definiton` in this file for each custom facet that is configured in Documentum xPlore in `indexserverconfig.xml`. The following is a sample code snippet of the `facetconfiguration.xml` file:

```
<facet-definitions>
  <facet-definition id="r_object_type" group-by="alpharange" max-values="8">
    <attributes>
      <attribute>r_object_type</attribute>
    </attributes>
    <properties>
      <property name="range">a:r, s:z</property>
    </properties>
    <sort>FREQUENCY</sort>
  </facet-definition>
</facet-definitions>
```

   Where:

   - `facet-definiton` is the name for the facetdefiniton-ID configured in Documentum xPlore.

- `group-by` defines the strategy, depending on data type. *OpenText Documentum xPlore - Administration and Development Guide (EDCSRC220100-AGD)* provides more details.
- `max-values` is the maximum number of results that must be displayed for this facet.
- `attribute` is the preferred display name that is shown to the end user in Content Connect.

Note: The Docker / Cloud administrator can modify the `facetconfiguration.xml` file on the Cloud environment to support the custom facets.

2. Add the following parameter in the `rest-api-runtime.properties` file located at `%TOMCAT-HOME%\webapps\<dctm-rest or D2-rest>\WEB-INF\classes`:

```
rest.search.facet.property.validation=false
```

3. Restart Content Connect server after you update the `facetConfiguration.xml` file.

7.3 Configuring Brava! server

You must setup the Brava! server in HTTPS mode. For more information about configuring Brava! Server for HTTPS, see *OpenText Brava! Enterprise Administration Guide*.

Ensure the following is installed on the Brava server:

- Microsoft Office
- JDK 1.8 or later

Before starting the Content Connect server, open a command prompt in the `Source` folder and run the following command:

```
node postInstallScript.js
```

You must add the IP address of the machine which initiates requests. Open the `Brava! Enterprise\Brava! Server\server.properties` file in the install directory and add the following entry:

```
legal.ip.get.sessionid=10.194.*.*,10.193.71.11,*,*
```

7.4 Configuring Documentum Secret Integration Services

For information about configuring Documentum Secret Integration Service (DSIS), see *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

Chapter 8

Deploying the second instance of Content Connect

If you have deployed Content Connect with one Endpoint and want to deploy a second instance of Content Connect on the same machine, perform the following steps:

1. Make a copy of the Content Connect source and follow the remaining steps in the second copy of the Content Connect source.
2. The administrator must create a new database for the second instance of Content Connect. You must also update the certificates path. "Deploying Content Connect" on page 9 provides more details.
3. Update the `MSGraphConfig.json` file with the new registration details. For more information, see step 8 in the *Deploying Content Connect* section.
4. Update the Content Connect port in the *<ContentConnect_installdir>*`\gulpfile.js` file at the following location:

```
gulp.task('cc_webserver', function () {
  gulp.task('AdminConsole')();
  gulp.src(config.release)
    .pipe(webserver({
      https: {key: certData.server['ssl_key_path'], cert:
certData.server['ssl_cert_path']},
      port: '8443',
      host: '0.0.0.0',
      directoryListing: false,
      fallback: 'index.html',
      middleware: function (req, res, next) {
        return [cors(req, res, next)];
      }
    }));
});
```

5. Update the admin service port in *<ContentConnect_installdir>*`\common\scripts\services\ConfigService.js`:

```
var configServicePort = 1607;
```

6. Update the admin service port in *<ContentConnect_installdir>*`\server\AdminConsole\configurations\general.json` file:

```
"server": {
        "port": 1607,
        "ssl_key_path": "<Absolute folder path>/key.pem",
        "ssl_cert_path": "<Absolute folder path>/cert.pem",
        "Allowed-Origins": "*",
        "purge_old_logs": "true"
        "secret": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
        "extension": "cc"
        "protocol": "IPV6"
                },
```

Note: Ensure that you specify the same port number in the `ConfigService.js` and `general.json` files in the preceding steps.

7. Update the Content Connect service name in *<ContentConnect_installdir>*/`package.json` file:

```
"config": {
 "service_name": "contentconnect"
    },
```

   For example, `service_name` as `contentconnect_new`.

8. Run the Content Connect server. Content Connect will now run on the port mentioned in step 3.

9. Ensure that you update the manifest file with relevant changes and upload it for the second server instance. "Configuring manifest files" on page 53 provide more details.

Chapter 9

Deploying Content Connect on client machine

To use Content Connect with Microsoft Office web and desktop applications, users require the Content Connect Microsoft Office add-in. The administrator deploys the add-in using the manifest file, which is obtained from the Content Connect Admin Console.

Note: For web applications, users can also upload the add-in using the manifest file that is provided by the Administrator.

You can deploy the add-in for all users from the Microsoft Office 365 Admin center and for individual users from the Microsoft Office application.

- `manifest-office-connector.xml` for the Microsoft Office add-in
- `manifest-outlook-connector.xml` for the Microsoft Outlook add-in

9.1 Downloading manifest files

Download the manifest files from the Admin Console after adding the Foundation REST API or Client REST API and OpenText Documentum CM client server endpoint details.

To download a manifest file:

1. In the Admin Console, navigate to the **Configuration** tab.
2. Click **Download manifest** and select the required manifest files **Outlook** or **Office** to download.

   The selected manifest file is downloaded to your computer.

9.2 Configuring manifest files

Administrators can manually modify manifest files before uploading to Microsoft by directly editing the files using the following supported modifications.

Update the following parameters in the manifest file:

- **DisplayName**: Change it to a relevant display name.
- **Description**: Add a description.
- If you have configured a load balancing server for Content Connect server the load balancer admin must open service port 1607 and 8443 for Content Connect nodes to communicate with the database server.

Note: You cannot upload the add-in using manifest files for the Microsoft Outlook desktop application. You must deploy the add-in in the Exchange Admin Center.

If you are using Brava, make sure you update the manifest file with the Brava domain and Content Connect domain in the `AppDomains` section. For example:

```
<AppDomain><cc-host>:<cc-port></AppDomain>
<AppDomain><brava-host>:<brava-port></AppDomain>
```

where,

- *<cc-host>* is Content Connect server name
- *<cc-port>* is Content Connect server port number
- *<brava-host>* is Brava server name
- *<brava-port>* is Brava server port number

If the administrator modifies and re-uploads manifest files that have previously been deployed, changes may not be seen by users until they next log-in to the OpenText Documentum CM repository.

9.3 Deploying and publishing the add-in using the Microsoft Office 365 admin center

For Microsoft Office and Microsoft Outlook web and desktop applications, the administrator can provide the Content Connect add-in through centralized deployment using the Microsoft Office 365 Admin center.

To upload the add-in for Microsoft Office and Microsoft Outlook application:

1. Log in to the Microsoft Office 365 Admin center.
2. Navigate to the **Settings**.
3. In the **Settings**, click **Integrated apps**.
4. In the **Integrated apps** section, select **Upload custom apps**.
5. Select the **App type** as **Office Add-in**.
6. Select the manifest file and then click **Next**.
7. Configure the app settings by assigning the app to specific users or groups.
8. Review the app details, click **Next** to upload and deploy.
9. Verify the deployment and check if the app is available for the targeted users.

Chapter 10

Integrating New Relic with Content Connect

New Relic is an application performance monitoring tool. This tool provides information that helps to analyze the application behavior, create real-time dashboards, and customization charts that are useful for application metrics. Go to https://newrelic.com for more information about New Relic.

New Relic has been integrated with both Content Connect on-premises and cloud deployments.

10.1 Integrating New Relic with Content Connect On-Premise deployment

Open the `contentconnect/newrelic.js` file and set the *`agent_enabled`* flag to `true`. Provide the appropriate value for the variable depending on your environment to pass them to your templates as described in the following table:

Note: By default, the *`agent_enabled`* flag is set to `false`.

Category	Name	Description
newRelic	appName	Provide a unique application name. Content Connect constructs the application name as *<NODE_NAME>*, *<appNameSuffix>* in the `newrelic.js` file. After logging into the NewRelic portal, search for the application name you have provided while configuring the New Relic integration.
newRelic	licenseKey	Set the license key for newRelic.

Note: Once the New Relic parameters are passed successfully, the `newrelic_agent.log` file is generated. This log file helps in monitoring the application connectivity with the New Relic portal.

For hosted servers, ensure that you have the required proxy settings configured to connect to the New Relic portal.

Chapter 11

Troubleshooting

This chapter describes some common errors and appropriate resolutions.

11.1 Database

- If you have issues when installing PostgreSQL, then install PostgreSQL without stack builder.
- If the database is not created properly or if the database already exist with missing tables, then delete the database and create database again as per "Deploying Content Connect" on page 9. Ensure that you stop the Content Connect node server before you run the `initialize.js` command.

11.2 Cannot upload manifest file

If you are not able to upload the manifest file or if Content Connect add-in is not visible, ensure that you have updated the manifest file as per "Configuring manifest files" on page 53.

11.3 Repository not listed in Login screen

If repository is not listed in the Content Connect log in screen:

- Ensure that you have configured authentication as per "Configuring the authentication in Admin Console" on page 29.
- Ensure that the uploaded manifest file is in accordance with the Endpoint Configuration.
- Run `https://<Rest-Server>:<Port>/dctm-rest/repositories` in the browser for Foundation REST API or Client REST API and ensure the application server is running.
- Ensure that Documentum CM Server is running.

11.4 Unable to access Admin Console

1. Stop the Content Connect service.
2. Navigate to the Content Connect folder in command prompt and run `AppServer-Start.bat`.
3. View the deployment steps. If there are any errors, resolve the error displayed in console.
4. Run `AppServer-Start.bat` again.
5. After successful deployment, the message at the command prompt should state the following:

```
Server is up on port: <port number>
```

6. Stop the Content Connect server from console and start the Content Connect service.

11.5 Endpoint test fails

Ensure that you have configured Foundation REST API or Client REST API as per "Configuring Foundation REST API or Client REST API" on page 23.

11.6 Changes in the updated manifest file are not reflected

If the updated manifest file is not reflecting the changes, then ensure that you clear the browser cache. If you are using desktop version of Microsoft Office then clear the cache in Microsoft Edge.

For Desktop version of Office, clear the contents in the location: `%LOCALAPPDATA%\Microsoft\Office\16.0\Wef\`

Ensure to delete the contents of the following folder, if it exists: `%userprofile%\AppData\Local\Packages\Microsoft.Win32WebViewHost_cw5n1h2txyewy\AC\#!123\INetCache\`

11.7 Search does not work

If search does not work, then check if the Documentum xPlore services are accessible and if the content is indexed.

11.8 Issues with OpenText Documentum CM client

- If OpenText Documentum CM client dialog boxes does not open, then ensure that you enable pop-ups in the browser.
- If OpenText Documentum CM client intelligent URL is not working, then ensure that you have downloaded and installed Documentum Client Manager and DCMApp is running.
- If the Import operation is failing with intelligent URL, verify if import works with OpenText Documentum CM client and resolve the issue before proceeding with intelligent URL from Content Connect.
- If you are using a proxy server, then disable the proxy on the system where the OpenText Documentum CM client server is deployed and add the proxy settings in the Content Connect OpenText Documentum CM client system.
- If OpenText Documentum CM client intelligent URL does not come up, ensure that you have configured OpenText Documentum CM client as per "Configuring OpenText Documentum CM client" on page 45.
- If you are using a proxy server, add the following attributes in the `catalina.properties` file located in App server where the Smart View is installed. This improves performance of the bulk import feature:

```
http.nonProxyHosts=<CC_IP>|<Documentum_Server_IP>
https.nonProxyHosts=<CC_IP>|<Documentum_Server_IP>

http.proxyHost=<Your proxy host IP address>
http.proxyPort=<Your proxy port address>

https.proxyHost=<Your proxy host IP address>
https.proxyPort=<Your proxy port address>
```

11.9 Using add-ins with Outlook on the web

For more information about using add-ins with Outlook on the web, see *Microsoft documentation*.

11.10 Preview fails

If the **Something went wrong** error message appears when trying to preview documents or emails, then ensure that the correct Brava server has been configured in the Admin Console. Additionally, validate the enterprise license key in the Brava server.

11.11 CORS error occurs in browser

SSO (OTDS) Authentication: If CORS error occurs during log in, add the Admin Console URL, D2–rest/dctm-rest URL in **OTDS Admin server** > **Trusted Sites** section.

11.12 Import fails on desktop clients

If the import fails with OpenText Documentum CM client on OS version Windows 10 1909 or later, then run the following command on the desktop client:

1. Open command prompt in the Administrator mode.
2. Run the following command:

```
checknetisolation LoopbackExempt -a -n=Microsoft.Win32WebViewHost_cw5n1h2txyewy
```

3. Close the desktop applications and relaunch.

11.13 Clear cache in client

When Content Connect is upgraded and the client is not reflecting the updated changes, refer to the following articles in Microsoft documentation to clear the cache:

- Troubleshoot user errors with Office Add-ins
- Troubleshoot Outlook add-in activation
- For Desktop Office version,
  - Clear the contents in the location: `%LOCALAPPDATA%\Microsoft\Office\16.0\Wef\`
  - Ensure to delete the contents of the following folder, if it exists: `%userprofile% \AppData\Local\Packages\Microsoft.Win32WebViewHost_cw5n1h2txyewy\AC \#!123\INetCache\`

11.14 Frequent prompts to login

When the Adminstrator switches URLs between Classic View and Smart View, the Administrator is frequently prompted to relogin using the repository credentials. To avoid these frequent prompts, perform the following steps:

1. Navigate to `<application-server>\webapps\d2-rest\WEB-INF\classes` and open the `rest-api-runtime.properties` file.
2. Add the following command:
   ```
   rest.use.relative.url=false
   ```
3. Restart the Client REST API application server.

11.15 Cabinets are not visible for users with special characters in Username

If the user name contains any special characters, the cabinets are not visible.

Perform the following step to view the cabinets in Content Connect:

- In Tomcat, open the `server.xml` file and add the following attribute to whitelist the special charaters:

  `relaxedQueryChars="[]|{}!<>"`

  For example:

  ```
  <Connector port="8080" protocol="HTTP/1.1" relaxedQueryChars="[]|{}"
  connectionTimeout="20000" redirectPort="8443" />
  ```

11.16 Microsoft applications thick client freezes upon OTDS authentication

If Microsoft Word and Microsoft Outlook thick client freezes upon OTDS authentication, perform the following step to relaunch Microsoft applications:

- Kill the Microsoft Word and the Microsoft Outlook instances and relaunch the applications.

11.17 Smart View—Lock operation fails while submitting changes

When the user completes the import document process in Content Connect and tries to lock and check-in (Submit) the document within a short time span, the lock operation fails due to the cached data retrieval.

Perform the following steps to blacklist the Objects API from the Smart View client-side caching:

1. Navigate to the application server where Smart View is deployed.
2. Locate the `sw-config.json` file in the `D2-Smartview\ui\sw-config.json` directory.
3. Remove or comment the following configuration in line number 30. (Add **"!!"** before the URL expression).

   Example: "!![a-f0-9] {16,}"
4. After updating `sw-config.json`, restart the application server and clear the browser cache.

11.18 Content Connect admin console—Warning appears while starting the server

Perform the following steps in the Node.js configuration files:

1. Navigate to the `\nodejs` folder in Program Files and open the `npm.cmd` and `npm` files in the text editor with Admin access.
2. Replace *`prefix -g`* with *`prefix --location=global`* in both the `npm.cmd` (line number 12) and `npm` files (line number 23).
3. Save both the files.
4. Run `AppServer-Start.bat`.

11.19 Internal server error occurs in OpenText Documentum CM client iURL during bulk import

While performing a bulk import in OpenText Documentum CM client, if the error `500 Internal Server error` occurs for a user, then check in client configuration if the **Linked document** > **Relation type** > **Relation Name** value is set to *`is_attachment_of`*.

If *`is_attachment_of`* is unavailable, then run the following command using Documentum CM Server **IAPI** to add:

```
create,c,dm_relation_type
set,c,l,relation_name
is_attachment_of
set,c,l,parent_type
dm_sysobject
set,c,l,child_type
dm_sysobject
set,c,l,security_type
NONE
save,c,l
```

After adding the value, set **Relation Name** to *`is_attachment_of`*.

11.20 Error 1053: Unable to start Content Connect on Windows Services

The following error occurs when Content Connect on Windows Service does not start within the expected time frame:

`Error 1053: The service did not respond to the start or control request in a timely fashion.`

In order to start Content Connect on Windows Service, make sure that the **ServicesPipeTimeout** entry exists in **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control**. If the entry is unavailable, create a new entry.

To verify the ServicesPipeTimeout entry:

1. Go to **Start** > Click **Run** > Type *`regedit`* > Click **OK**.
2. Locate and click the following registry subkey **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control**.
3. In the right side of the pane, check for the **ServicesPipeTimeout** entry.

   If the **ServicesPipeTimeout** entry does not exist, create an entry using the following steps.

To create the ServicesPipeTimeout entry:

1. From the **Edit** menu, click **New** > **DWORD Value**.
2. Specify the entry name as `ServicesPipeTimeout`.

3. Right-click **ServicesPipeTimeout** and click **Modify**.
4. In the **Edit DWORD Value** dialog box, specify the following:
   - **Value data**: 60000

     This value represents the time in milliseconds before a service times out.
   - **Base**: Decimal
5. Click **OK**.
6. Restart the computer.

11.21 Content Connect does not load in Microsoft Outlook web client

If you get the error, “Failed to read the 'sessionStorage' property from 'Window': Access is denied for this document”, when opening Content Connect in Microsoft Output web client, then you must block the third-party cookies in the browser. For Google Chrome, complete the following steps:

1. In Google Chrome, open **Settings**.
2. In **Privacy and security**, click **Cookies and other site data**.
3. Select the **Block third party cookies in Incognito** option.

11.22 NodeJS Servers unable to verify CA certificates

NodeJS Servers cannot verify CA certificates if there is multiple certificates in a chain. When there is a certificate chain, append all the chain certificates file to a single file.

To append the CA certificates to a single file:

1. Create a `rootCA.pem` file.
2. Add a CA certificate string in `rootCA.pem` file and save it. If there are multiple certificates, append all the certificates strings to the `rootCA.pem` file to create a single file.
3. On your computer, right-click **This PC** and select **Properties**.
4. Click **Advanced system** settings.
5. In the **System Properties** dialog box, click **Environment Variables**.
6. In the **Environment Variables** dialog box, under the **System variables** area, click **New**.
7. In the **New System Variable** dialog box, in the **Variable name** field, enter `NODE_EXTRA_CA_CERTS` and **Variable value** as `certificate path(rootCA.pem)`.

8. Restart your computer.

11.23 Bulk import failure

When using Foundation REST API with IBM WebSphere Liberty, the bulk import fails.

Resolution: To resolve this issue, in the `IBM_Liberty_profile\wlp-webProfile8-19.0.0.8\wlp\usr\servers\defaultServer\server.xml` file, add the `protocolVersion` as follows:

```
<httpEndpoint id="defaultHttpEndpoint" host="*"
httpPort="9080"
httpsPort="9443"
protocolVersion="http/1.1"/>
```

11.24 Invalid application resource URL

When a user imports emails, the import operation fails with the error "Invalid application resource Url."

Resolution: You must verify that the **Application ID URI** value under **Expose an API** (see "Content Connect Application Registration with Microsoft Entra ID" on page 14) is the same as the value in the manifest file. Make sure that the client ID is added at the end.

Example:

```
<Resource>api://<domain>/<clientid></Resource>
<Resource>api://opentext.corp.uk/2287faca-dbd8-449d-b44c-xxxxxxx</Resource>
```

Chapter 12

Content Connect Extension Framework

12.1 Introduction

The extension framework allows customers to control or extend certain aspects of the attribute / meta-data screen in OpenText™ Content Connect. Customers can write their own code to retrieve and provide values of attributes of custom types to the main Content Connect application. The extension receives events when the values of these attributes are changed on the user interface allowing the extension to reload or modify the values as appropriate to the business context.

Note: Content Connect is a client side JavaScript application programmed in AngularJS.

12.2 Interfaces / Objects

The following sections describe the interfaces and objects available in the extension through which it is possible to control the meta-data values on the user interface of Content Connect.

12.2.1 CCAttributeExtension

This is the main AngularJS service that will contain the business logic of the custom extension. This service must contain the following functions:

- getExtAttributes
- onChange
- populateLists

12.2.2 function getExtAttributes

This is required to return the list of attributes, their values, and any other associated flags as required.

Inputs

This function has no input parameters.

Return

The function should return a promise that resolves when the values are available.

Remarks

An object is to be returned upon promise resolution containing names and values of all external attributes. The name field is mandatory, and all other fields are optional.

- **name**—Must match the attribute name in OpenText Documentum CM as returned by dctm-rest.
- **propValue**—Field value.
- **uiType**—Defined only for pick lists and the value must be “list”.
- **values**—List of values (array) if this field is a pick list.
- **readonly**—Can be true or false. Used to make the field readonly.
- **hidden**—Can be true or false. Used to hide the field.

Important

- Mandatory attributes cannot be made read-only or hidden.
- Extension framework will support only the values for the attributes returned by dctm-rest.

```
Example (pseudo code):
    var extAttributes = {
        "attributes" : [
        {
            "name": "attribute 1 name",
            "propValue": attribute1Value,
            "readonly": <true/false>,
            "hidden": <true/false>
        },
        {
            "name": "attribute 2 name",
            "propValue": attribute2Value,
            “readonly": <true/false>,
            "hidden": <true/false>
        },
        …]
    };
```

12.2.3 function onChange (attribName, newValue)

This function is required to listen to change events from the user interface.

Inputs

This function is provided the name of the changed attribute and its new value.

Return

The function should return true or false.

Return `true` to have Content Connect retrieve the values of attributes again (that is, instruct Content Connect to call `getExtAttributes` again). Return `false` if no further action is needed.

12.2.4 function populateLists (attributes)

This function is required specifically to populate values in the drop-down lists in Content Connect.

Inputs

List of attributes and their current values.

Return

Should return a promise that resolves when the list values are available.

Remarks

An object must be returned upon promise resolution containing options to be displayed in the pick lists. The structure of the object is similar to that returned in getExtAttributes, except that only the "values" field is honored while all other fields are ignored.

12.2.5 CCExtensionHelperService

A helper service available to the extension. It provides configuration details and runtime values to the extension. The extension can make use of the following functions as required by its business logic:

- **getEndpointBaseUrl**—Returns the URL of the currently configured endpoint.
- **getAuthToken**—Returns the authentication token for the currently logged in user.
- **getRepositoryName**—Returns the name of the logged in repository.

12.2.6 Configuring an extension

Content Connect comes with a blank extension containing the required functions. This can be modified as needed for the specific business use case. A sample extension is also provided to help understand how the different aspects of the extension can be coded.

The only file that requires changes is "`CCAttributeExtension.js`" located in the "`extensions`" folder in the root directory. If you want to use the sample extension, it should be renamed to "`CCAttributeExtension.js`". After making any changes to this file, restart the Content Connect web server by running the "AppServer-Start.bat" (or the appropriate OS specific script) which will also minify the files and place them in the "dist" folder. You may also clear the Browser cache.

> **Important**
> Ensure that the "CCAttributeExtension.js" file is not renamed incorrectly or deleted during the configuration process.

12.2.7 Debugging the extension

Since Content Connect is a JavaScript application, it can be debugged almost entirely using the built-in developer tools in modern browsers. Developer tools can be accessed in Chrome from the “More Tools” sub-menu.

During the development phase, to allow debugging of the extension’s un-minified code, use the following command to start the web server instead of the provided batch file:

```
npm run-script gulp ContentConnect_Source
```

This will not minify the source and run the webserver from the root directory instead of the “dist” directory.

Note: Clear the browser’s cache after changes are made to the source to allow the browser to receive the latest code. During debugging, this can also be done by holding down the reload button in Chrome and selecting “Empty Cache and Hard Reload”.

12.2.8 Extended validation for user interface

In order to extend Content Connect attribute validation, in the `CCAttributeExtension.js` file, uncomment the lines after the following comment:

```
/**In order to extend Validation Framework from Content Connect (UI validations on the
input fields), uncomment the following lines and add your own validations. */

/*
```

These lines when uncommented, overrides the attribute validation directive and takes priority over Content Connect attribute validation.

Note: If you are upgrading Content Connect to the latest version, ensure to merge the changes from the `customtype.js` file.