IBM Content Manager OnDemand Native Encryption

January 6, 2023

Greg Felderman

Chief Architect - IBM Content Manager OnDemand

Contents

"IBM Content Manager OnDemand Native Encryption" Rev: January 6, 2023

Introduction

Overview

Content Manager OnDemand native encryption encrypts your physical data, requires no hardware, software, or application changes, and provides transparent and secure key management.

Encryption is the process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process. It is an effective way of protecting sensitive information that is stored on media or transmitted through untrusted communication channels. Encryption is mandatory for compliance with many government regulations and industry standards.

In an encryption scheme, the data requiring protection is transformed into an unreadable form by applying a cryptographic algorithm and an encryption key. A cryptographic algorithm is a mathematical function that is used in encryption and decryption processes. An encryption key is a sequence that controls the operation of a cryptographic algorithm and enables the reliable encryption and decryption of data.

Some data encryption solutions for protecting data at rest are suitable in cases of physical theft of disk devices, and some can protect against privileged user abuse. With Content Manager OnDemand native encryption, the system itself encrypts the data before it calls the underlying storage manager to write that data to media. Content Manager OnDemand native encryption is suitable for protecting data in cases of either physical theft of disk devices or privileged user abuse.

A local or external key manager is typically used to manage the keys. A Content Manager OnDemand data encryption key (DEK) is the encryption key with which actual user data is encrypted. A master key (MK) is a "key encrypting key"; it is used to protect the DEK. Although the DEK is stored and managed inside the Content Manager OnDemand instance database, the MK is stored and managed outside of the Content Manager OnDemand instance database.

Encrypted master keys are stored in either a PKCS#12 compliant keystore, which is a storage object for encryption keys that exists at the operating system level or to a Key Management Interoperability Protocol (KMIP) version 1.1 or greater compliant server. Only the Content Manager OnDemand library server interacts with the keystore or the KMIP server; it is not required for object server(s).

Enabling encryption support for Content Manager OnDemand for Multiplatforms

IBM Global Security Kit (GSKit)

Content Manager OnDemand Version 10.5 uses the IBM Global Security Kit to support FIPS 140-2 certified cryptographic capabilities. You must ensure that GSKit is installed on your system. If installing on AIX or Linux, you must manually install GSKit. When installing on the Windows platform, the Content Manager OnDemand installer will install GSKit for you.

Content Manager OnDemand ships with GSKit 8.0.55.12 and uses the default GSKit installer (on UNIX, the GSKit shared libraries will have links in /usr/lib). Other products from IBM also support using GSKit, however the version of GSKit that ships with other products might be different than what ships with Content Manager OnDemand. You must ensure that Content Manager OnDemand utilizes GSKit 8.0.55.12 or later. Other IBM products might not use the default GSKit installation and therefore have their own version of GSKit in their product installation directory. On UNIX platforms, you might need to set the appropriate dynamic library path order environment variable (LIBPATH or LD_LIBRARY_PATH) in which to search for GSKit when running a Content Manager OnDemand command. Use the gsk8ver or gsk8ver_64 command to determine which version of GSKit is in the default path.

See the Content Manager OnDemand README file for further information on installing GSKit.

Operating System	GSKit Install Location
AIX	/usr/opt/ibm/gsk8 /usr/opt/ibm/gsk8_64
Linux	/usr/local/ibm/gsk8 /usr/local/ibm/gsk8_64
Windows	Standard install path: C:\Program Files\IBM\GSK8 Windows 32 bit installations on x86_64 systems: C:\Program Files (x86)\IBM\GSK8

To configure Content Manager OnDemand native encryption, you will need to create and manage your master keys in either a PKCS#12 keystore or to a Key Management Interoperability Protocol (KMIP) version 1.1 or greater compliant server.

Using a PKCS#12 keystore database

Configuring a PKCS#12 keystore database

Invoke GSKCapiCmd by using the gsk8capicmd or gsk8capicmd_64 command.

Note: Windows only: When invoking the GSKCapiCmd tool, you must ensure that the GSKit bin and lib directories are in your PATH.

64bit:

PATH=C:\Program Files\IBM\GSK8\bin;C:\Program Files\IBM\GSK8\lib64;%PATH%

32bit on a Windows 64bit system:

PATH=C:\Program Files (x86)\IBM\GSK8\bin;C:\Program Files (x86)\IBM\GSK8\lib;%PATH%

Use the GSKCapiCmd tool to create your PKCS#12 keystore database. The GSKCapiCmd is a non-Java-based command-line tool. Java™ does not need to be installed on your system to use this tool.

Although not required, it is recommended that you store the key database in the Content Manager OnDemand server installation config sub-directory.

AIX:	/opt/IBM/ondemand/V10.5/config
Linux:	/opt/ibm/ondemand/V10.5/config
Windows:	C:\Program Files\IBM\OnDemand\V10.5\config

For example, the following command creates a PKCS#12 keystore called odkeys.p12 and a stash file called odkeys.sth:

```
gsk8capicmd_64 -keydb -create -db "odkeys.p12" -pw "myKeyDBpasswd" -type pkcs12 -stash
```

The -stash option creates a stash file at the same path as the key database, with a file extension of .sth. At Content Manager OnDemand start-up, GSKit uses the stash file to obtain the password to the key database. When you create a key database with the -populate option, it is automatically populated with several predefined trusted certificate authority (CA) certificates. A trusted CA is one whose root certificate is noted as trusted in the key database. You would specify this option if you plan to also use the same keystore to store SSL certificates to be used by Content Manager OnDemand. Although it is allowed, we recommend using different keystores for the SSL certificates and the master key.

Note: You should use strong file system protection on the keystore database and stash file.

Defining the PKCS12 keystore to the Content Manager OnDemand instance

Use the following command to configure the keystore to the Content Manager OnDemand instance:

```
arssockd -I <INSTANCE> -d “keystore_type=PKCS12,keystore_location=/opt/IBM/ondemand/V10.5/config/odkeys.p12,keystore_mkl=*”
```

This command will configure the instance to use the specified pkcs12 keystore. Using “keystore_mkl=*”, tells the command to do several actions: 1) Create a data encryption key (DEK); and 2) create a master key (MK) which will be used to encrypt the DEK. The DEK will be stored in the Content Manager OnDemand database. The MK will be stored in the pkcs12 keystore. Both sets of keys are randomly generated. Depending on individual corporate security requirements, you can change the MK as often as you wish. Changing the MK causes the DEK to be restored based on being encrypted by the new MK. This can be done by issuing the following command:

```
arssockd -I <INSTANCE> -d “keystore_mkl=*”
```

If you look at the keystore, you will see that Content Manager OnDemand creates a label for every master key (MK), which contains both the instance name and owner as well as a timestamp. Only one master key (MK) is used at a time; the others are maintained for historical records. To see the list of labels, issue this command:

```
gsk8capicmd_64 -cert -list all -db "odkeys.p12" -stashed

#   ONDEMAND_ARCHIVE_ROOT_2017-01-24-08.10.01.918006
#   ONDEMAND_ARCHIVE_ROOT_2017-02-24-07.16.25.678148
```

These steps should only be done on the Content Manager OnDemand library server. Content Manager OnDemand object servers do not need the keystore database, since they will communicate with the library server directly.

The steps to enable encryption for an instance are now complete. No encryption will be performed until it is enabled in a Content Manager OnDemand application group.

Using Key Management Interoperability Protocol (KMIP)

Configuring Key Management Interoperability Protocol (KMIP)

To set up a centralized keystore with a key manager that is configured for the Key Management Interoperability Protocol (KMIP) for use with Content Manager OnDemand native encryption, you need to create a KMIP keystore configuration file. Once you have created the configuration file, you can enter parameter values to configure SSL communication between the Content Manager OnDemand instance and the key manager.

On the Content Manager OnDemand library server, create the KMIP keystore configuration file in a text editor (such as /opt/IBM/ondemand/V10.5/config/ars.kmip).

Example

```
VERSION=1
PRODUCT_NAME=ISKLM
ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=FALSE
SSL_KEYDB= /opt/IBM/ondemand/V10.5/config/ondemand.p12
SSL_KEYDB_STASH=/opt/IBM/ondemand/V10.5/config/ondemand.sth
SSL_KMIP_CLIENT_CERTIFICATE_LABEL=cmod_client_label
MASTER_SERVER_HOST=serverName.domainName
MASTER_SERVER_KMIP_PORT=kmipPortNumber
CLONE_SERVER_HOST=clone1.domainName
CLONE_SERVER_KMIP_PORT=kmipPortNumber
CLONE_SERVER_HOST=clone_n.domainName
CLONE_SERVER_KMIP_PORT=kmipPortNumber
```

Parameters

VERSION

Required. Version of the configuration file. Currently, 1 is the only supported value.

PRODUCT_NAME

Required. Key manager product. Supported values:

ISKLM

IBM® Security Key Lifecycle Manager

KEYSECURE

SafeNet KeySecure

OTHER

For any other key manager that supports the Key Management Interoperability Protocol (KMIP) version 1.1 or higher

ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP

Optional: Allow Content Manager OnDemand to insert new keys into the KMIP key manager. New keys are inserted when the arssockd -I <INSTANCE> -d “keystore_mkl=*” command is run. When this parameter is set to TRUE, new keys are allowed to be inserted. If it is set to FALSE, an error is returned. You should only set this parameter to TRUE when you are creating the initial master key or rotating the master key. Otherwise, you should set this parameter to FALSE. Default value: FALSE.

SSL_KEYDB

Required. Absolute path and name of the local keystore file that holds the SSL certificates for communication between the Content Manager OnDemand library server and the KMIP key manager.

SSL_KEYDB_STASH

Required. Absolute path and name of the stash file for the local keystore that holds the SSL certificates for communication between the Content Manager OnDemand library server and the KMIP key manager.

SSL_KMIP_CLIENT_CERTIFICATE_LABEL

Required. The label of the SSL certificate for authenticating the client during communication with the KMIP key manager.

DEVICE_GROUP

Name of the KMIP key manager device group containing the keys used by the Content Manager OnDemand library server. This parameter is only required for IBM Security Key Lifecycle Manager (ISKLM).

MASTER_SERVER_HOST

Required. Host name or IP address of the KMIP key manager. (For ISKLM, this information is available on the "Welcome" tab of the web console.)

MASTER_SERVER_KMIP_PORT

Required. The "KMIP SSL port" of the KMIP key manager. (For ISKLM, this information is available on the "Welcome" tab of the web console.)

CLONE_SERVER_HOST

Optional. Host name or IP address of the secondary KMIP keystore. Default value: None. You can specify up to five clone servers by repeating the CLONE_SERVER_HOST and CLONE_SERVER_KMIP_PORT parameter pairs in the configuration file, specifying each host with a different value. Clone servers are considered read-only and are only used for retrieving existing master keys from the KMIP keystore. Clone servers are not used when inserting a new key.

CLONE_SERVER_KMIP_PORT

Optional. The "KMIP SSL port" of the secondary KMIP keystore. Default value: None. You can specify up to five clone servers by repeating the CLONE_SERVER_HOST and CLONE_SERVER_KMIP_PORT parameter pairs in the configuration file, specifying each host with a different value.

COMMUNICATION_ERROR_RETRY_TIME

Optional. The number of times the Content Manager OnDemand library server cycles through the list of configured master and clone KMIP key managers if the connection fails or an error is returned from all of the KMIP key managers. A wait of a length specified in the ALL_SERVER_UNAVAILABLE_SLEEP parameter is inserted before each cycle. Default value: 50.

UNAVAILABLE_SERVER_BLACKOUT_PERIOD

Optional. The amount of time, in seconds, to skip sending key requests to a particular master or clone KMIP key manager after a failed connection attempt or after it has returned errors. Default value: 300 seconds.

ALL_SERVER_UNAVAILABLE_SLEEP

Optional. When all master and clone KMIP key managers are unavailable and in a blackout period, this parameter is the amount of time to wait, in seconds, before removing the blackout period and reattempting connections to all KMIP key managers. Default value: 0 seconds.

Defining the Key Management Interoperability Protocol (KMIP) to the Content Manager OnDemand instance

Use the following command to configure KMIP to the Content Manager OnDemand instance:

```
arssockd -I <INSTANCE> -d “keystore_type=KMIP,keystore_location=/opt/IBM/ondemand/V10.5/config/ars.kmip,keystore_mkl=*”
```

This command will configure the instance to use the specified KMIP keystore. The use of “keystore_mkl=*” tells the command to do several actions: 1) Create a data encryption key (DEK); and 2) create a master key (MK) which will be used to encrypt the DEK. The DEK will be stored in the Content Manager OnDemand database. The MK will be stored in the KMIP compliant server. Both sets of keys are randomly generated. Depending on individual corporate security requirements, you can change the MK as often as you wish. Changing the MK causes the DEK to be restored based on being encrypted by the new MK. This can be done by issuing the following command:

```
arssockd -I <INSTANCE> -d “keystore_mkl=*”
```

Note that whenever you are creating a new MK, you must set the following in the KMIP keystore configuration file: ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=TRUE

These steps should only be done on the Content Manager OnDemand library server. Content Manager OnDemand object servers do not need the keystore database, because they will communicate with the library server directly.

The steps to enable encryption for an instance are now complete. No encryption will be performed until it is enabled in a Content Manager OnDemand application group.

Configuring TLS between a Content Manager OnDemand instance and an IBM Security Key Lifecycle Manager (ISKLM) centralized KMIP key manager

To store master keys in a centralized keystore with Content Manager OnDemand native encryption, you need to set up TLS (Transport Layer Security) communication between the Content Manager OnDemand instance and the centralized KMIP key manager. Consult the IBM Secure Key License Manager documentation for general information about ISKLM.

After you create a local keystore to store your TLS certificates, the gsk8capicmd_64 command is used on the Content Manager OnDemand library server to create, extract, and add TLS certificates to the local keystore.

Some examples below show self-signed certificates. Self-signed certificates are suitable for test environments, but for production environments, certificates that are signed by third party certificate authorities are more appropriate.

Some information about using the IBM® Security Key Lifecycle Manager web interface and command line interface is included below. For more complete information, see Setup for SSL handshake between IBM Security Key Lifecycle Manager server and client device.

Follow the steps below to configure TLS with an ISKLM centralized KMIP key manager:

1. On the Content Manager OnDemand library server, create an TLS signer certificate.
    a. Create the certificate by issuing the gsk8capicmd_64 command.
        Example

        ```
        gsk8capicmd_64 –cert -create -db "clientkeydb.p12" -label "CMOD_signer_certificate"
          -dn "CN=weblinux.Raleigh.ibm.com,O=ibm,OU=IBM HTTP
           Server,L=RTP,ST=NC,C=US" -sig_alg SHA256_WITH_RSA -size 2048
        ```

    b. Extract the certificate to a file by issuing the gsk8capicmd_64 command.
        Example

        ```
        gsk8capicmd_64 -cert -extract -db "clientkeydb.p12" -label "CMOD_signer_certificate"
          -target "/path/to/CMOD_certificate_file.pem" -format ascii -fips
        ```

    c. Securely transmit the Content Manager OnDemand server certificate file to the centralized key manager.

2. On the centralized key manager, add the Content Manager OnDemand server certificate to the keystore.
    The following substeps describe how to add a certificate to IBM Security Key Lifecycle Manager using the web console.
        a. Create a device group:
            i. Select "Create" in the "Device Group" list of the "Advanced Configuration" tab.
            ii. Select the device family "General Parallel File System (GPFS)" and then enter "CMOD" as the new device group name.
            iii. Leave the "Enable machine affinity" check box unselected.
        b. Import the Content Manager OnDemand server certificate file:
            i. On the "Welcome" tab select your new group, "CMOD".

ii. From the "Go to" list, select "Manage Keys and Devices". This will bring you to the Advanced Configuration tab.
iii. Select "Certificates" from the "Add" list.
iv. Specify the certificate name and the file path when prompted.
v. In the "Advanced Configuration" window, select "Import" from the "Client Device Communication Certificates" menu.

3. On the centralized key manager, create a TLS signer certificate.
   The following substeps describe how to create a certificate and then extract it to a file using the IBM Security Key Lifecycle Manager web console and command-line interface.
   a. Create a self-signed certificate or obtain a certificate from a certificate authority.
   b. Extract the certificate to a file.
   c. Securely transmit the centralized key manager certificate file to the Content Manager OnDemand server.

4. On the Content Manager OnDemand server, add the centralized key manager certificate to the local keystore by issuing the gsk8capicmd_64 command.
   Example
   ```
   gsk8capicmd_64 -cert -add -db "clientkeydb.p12" -label "ISKLM_signer_certificate"
     -file "/path/to/ISKLM_certificate_file.pem"
   ```

Configuring TLS between a Content Manager OnDemand instance and a KeySecure centralized KMIP key manager

To store master keys in a centralized KMIP keystore with Content Manager OnDemand native encryption, you need to set up TLS (Transport Layer Security) communication between the Content Manager OnDemand instance and the centralized key manager. Consult the KeySecure documentation for general information about KeySecure.

After you create a local keystore to store your TLS certificates, the gsk8capicmd_64 command is used on the Content Manager OnDemand library server to create, extract, and add TLS certificates to the local keystore.

Follow the steps below to configure TLS with a KeySecure centralized KMIP key manager:

1. On KeySecure, create a CA and add it to the Trusted CA list.
    a. Verify that a CA certificate is created or installed. Make sure that the CA is added to the trusted CA list.
    b. Make sure that a server certificate request is created and signed with the CA certificate.
    c. Check that a Cryptographic Key Server is created. Also, verify that the appropriate authentication settings are configured.
        i. Ensure the appropriate Cryptographic Key Server Properties:
            - **Protocol:** Select KMIP.
            - **IP:** Select ALL or a specific IP address.
            - **Port:** Select a port number. The standard KMIP port number is 5696. In the centralized keystore configuration file, the value for the MASTER_SERVER_KMIP_PORT or CLONE_SERVER_KMIP_PORT parameter must be configured according to the value specified for the port number.
            - **Use SSL:** Select True
            - **Server Certificate:** Select the label of the server certificate.
        ii. Ensure the appropriate Authentication Settings:
            - **Password Authentication:** Select “Not Used”.
            - **Client Certification Authentication:** Select “Used for SSL session and username”.
            - **Trusted CA list Profile:** Select the profile that contains the Trusted CA list to which the CA was added.
            - **User name Field in Client Certificate:** Select either the CN or OU value from the dropdown list.
            - **Require Client Certificate to Contain Source IP:** Leave unchecked.
        iii. Create a Local User whose user name matches the *User name field in Client Certificate* field in the client certificate.
    d. Download the CA certificate to the client keystore.

2. On the Content Manager OnDemand library server, add the CA certificate that was previously downloaded to the local keystore.

   Example

   ```
   gsk8capicmd_64 -cert -add -db "clientkeydb.p12" -stashed -label "trustedCA" -file "trustedCA.crt"
   ```

3. On the Content Manager OnDemand library server, create a client certificate request.

   Example

   ```
   gsk8capicmd_64 -certreq -create -db "clientkeydb.p12" -stashed -label "clientCert" -dn
   "CN=CMODKeySecureUser,O=IBM,OU=CMOD,L=Boulder,ST=CO,C=US"
    -target "client_cert_request.arm"
   ```

4. At your CA, sign the client certificate request with the CA certificate, and then download the signed certificate.

5. On the Content Manager OnDemand library server, add the signed client certificate to the local keystore.

   Example

   ```
   gsk8capicmd_64 -cert -receive -db "clientkeydb.p12" -stashed  -file "client_cert_signed.arm"
   ```

Enabling encryption support for Content Manager OnDemand for z/OS

Content Manager OnDemand for z/OS depends on the z/OS Cryptographic Services Integrated Cryptographic Service Facility (ICSF) for cryptographic support. ICSF is an included element of z/OS that provides cryptographic services for the operating system. These services include:

- Application programming interfaces (APIs) for applications that need to perform crypto functions such as encryption and decryption of data, digital signatures, Message Authentication Codes (MACs), and key generation
- Basic key management
- Keystores for cryptographic key material
- Providing access to Hardware Cryptographic Coprocessors, Cryptographic Accelerators, and the CP Assist for Cryptographic Function
- Support for FIPS 140-2 mode

ICSF can interface with the System Authorization Facility (SAF, sometimes referred to as RACF) to restrict access to specific key labels in the Cryptographic Key Data Set (CKDS), ensuring that users cannot access the keys belonging to other users. Additionally, SAF can be used to restrict access to specific ICSF APIs. This prevents the misuse of any crypto hardware by unauthorized users. Detailed instructions for enabling these capabilities are documented in the ICSF Administrator's Guide.

The keystores are shared by all applications that use ICSF, and because of that, the necessary backup and recovery of the Content Manager OnDemand keys will be satisfied by the existing installation procedures for protecting the keystores.

Content Manager OnDemand for z/OS stores its master key (MK) in the CKDS keystore provided by ICSF. To allow Content Manager OnDemand to store its MK in the CKDS, ICSF requires the CKDS to be in a more recent variable format, either LRECL=1024 or LRECL=2048. If the CKDS is currently using the older fixed format CKDS, ICSF provides instructions to convert to variable format.

The following command can be used enable encryption by the Content Manager OnDemand instance:

arssockd -I <INSTANCE> -d "keystore_type=PKCS12,keystore_location=CKDS,keystore_mkl=*"

If using SAF to control access to key labels, the RACF user associated with the Content Manager OnDemand library server must have CONTROL access to those labels in the CSFKEYS class to allow it to read, write, create and delete keys. The Content Manager OnDemand keys in the CKDS have a label of the format:

ONDEMAND.*instance*.dbowner.*yyyy.mm.dd.hh.mm.ss.tttttt*

This allows SAF profiles to be created of the form ONDEMAND.*instance*.dbowner.**, restricting different instances to specific sets of key labels, and preventing other users from accessing those keys.

If using SAF to control access to ICSF APIs, the Content Manager OnDemand server needs READ access to the CSFKRW, CSFKRD, CSFKRC, CSFKRR, and CSFRNGL resources in the CSFSERV class.

All Content Manager OnDemand servers for a given instance must be using the same CKDS. If running in a sysplex environment, the CKDS must be shared among sysplex members running a Content Manager OnDemand server. See CKDS management in a sysplex for considerations about sharing the CKDS.

The IBM z/OS Knowledge Center topic titled Cryptographic Services ICSF: System Programmer's Guide provides information on how to initialize, customize, operate, and diagnose the z/OS Integrated Cryptographic Service Facility (ICSF).

These steps should only be done on the Content Manager OnDemand library server. Content Manager OnDemand object servers do not need the same CKDS, since they will communicate with the library server directly.

The steps to enable encryption for an instance are now complete. No encryption will be performed until it is enabled in a Content Manager OnDemand application group.

Enabling encryption support for Content Manager OnDemand for i

Content Manager OnDemand for i depends on the IBM i operating system for cryptographic support. IBM i has a repository for master keys which are used to encrypt other keys, and keystore files to store encrypted keys. See the IBM i Knowledge Center topic titled Cryptographic services key management for more information and instructions for creating and managing these files.

Content Manager OnDemand for i requires that a keystore file exist on your IBM i server. If you do not have a keystore file, a new one must be created. Since encryption must be turned on explicitly for an individual instance, it makes sense that the keystore file for an instance be in the instance library.

The following steps are required to enable encryption for an instance. Instance name QUSROND is used in all the examples, but the examples will work for any instance if you replace QUSROND with your instance name.

If you do not have a master key, you must create one and then set the master key.

To load a master key from the IBM Navigator for i interface, follow these steps:

1. Hover over the **Security** icon in the IBM Navigator for i window to display the **Security** menu.
2. Select **Cryptographic Services Key Management**.
3. Select **Manage Master Keys**.
4. Select the Master Key.
5. Select **Load part** from the **Actions** menu.
6. Specify the **Passphrase** and click **OK**.

You can also use the Add Master Key Part (ADDMSTPART) command on IBM i to load a key part for the specified master key. For example:

```
ADDMSTPART MSTKEY(1) PASSPHRASE('My Passphrase')
```

To set the master key from the IBM Navigator for i interface, follow these steps:

1. Hover over the **Security** icon in the IBM Navigator for i window to display the **Security** menu.
2. Select **Cryptographic Services Key Management**.
3. Select **Manage Master Keys**.
4. Select the Master Key.
5. Select **Set** from the **Actions** menu to set the master key.

You can also use the Set Master Key (SETMSTKEY) command on IBM i to set the specified master key that has parts already added. For example:

```
SETMSTKEY MSTKEY(1)
```

The next step is to create a keystore file. You can create as many keystore files as desired.

When you create a keystore file by using the IBM Navigator for i interface, it is automatically added to your list of managed keystore files. It is recommended that the keystore file for an instance be in that instance library. For example, the keystore file for the QUSROND instance would be in the QUSROND library.

To create a new keystore file by using the IBM Navigator for i interface, follow these steps:

1. Hover over the **Security** icon in the IBM Navigator for i window to display the **Security** menu.
2. Select **Cryptographic Services Key Management**.
3. Select **Manage Cryptographic Keystore Files**.
4. Click **Create Keystore** from the **Actions** menu.
5. Enter the Keystore name for the new keystore you want to create and specify the Library in which you want to create the new keystore.
6. Enter the Description of the new keystore that you want to create.
7. Enter the Master key that you want to be associated with the new keystore file.
8. Select the Public authority that you want to assign to the new keystore file.
9. Click OK.

You can also use the Create Keystore File (CRTCKMKSF) command on IBM i to create a database file for storing cryptographic key records. For example:

CRTCKMKSF KEYSTORE(QUSROND/KEYSTORE) MSTKEY(1) AUT(*EXCLUDE) TEXT('Keystore for instance QUSROND')

The instance user profile must be authorized to the keystore file with *ALL authority. You can use the Grant Object Authority command on IBM i to set the correct authority. For example:

GRTOBJAUT OBJ(QUSROND/KEYSTORE) OBJTYPE(*FILE) USER(QUSROND) AUT(*ALL)

To enable encryption support for an instance, you must run the following command in QSHELL on your IBM i server:

/qsys.lib/qrdars.lib/arssockd.pgm -I instanceName -d "keystore_type=PKCS12,keystore_location=instanceName/keyStoreFile,keystore_mkl=*"

For example, to enable encryption support for the QUSROND instance:

/qsys.lib/qrdars.lib/arssockd.pgm -I qusrond -d “keystore_type=PKCS12,keystore_location=QUSROND/KEYSTORE,keystore_mkl=*"

The output from this command should look like this:

```
keystore_type=PKCS12
keystore_location=QUSROND/KEYSTORE
keystore_mkl=ONDEMAND_QUSROND_QRDARS400_2017-01-27-16.50.23.063912
keystore_mkl_dt=2017-01-27 10:50:23.063912
```

These steps should only be done on the Content Manager OnDemand library server. Content Manager OnDemand object servers do not need the keystore file, since they will communicate with the library server directly.

The steps to enable encryption for an instance are now complete. No encryption will be performed until it is enabled in a Content Manager OnDemand application group.

Configuring Content Manager OnDemand application groups

You can determine and configure which Content Manager OnDemand application groups that you want to enable for use with encryption. Keep in mind that only new data will be encrypted; it is not possible to encrypt existing data without retrieving and reloading such data.

To enable encryption for an application group, follow these steps:

1. Log on to the OnDemand Administrator client.
2. Double-click **Application Groups** in the left panel.
3. Right-click the application group you wish to enable, then click **Update**.
4. On the **General** tab, click the **Advanced…** button.
5. Click the **Yes** radio button in the **Encryption** section under the **Encrypt physical documents at rest?** heading.
6. Click OK on the **Database Information** panel, but do not click OK to update the application group yet!

You then must add an Encryption field to the application group. The field must have data type Small Int (2), and the Encryption checkbox must be selected.

While still in update mode in the application group, add an Encryption field to the application group by following these steps:

1. Click the **Field Definition** tab.
2. Enter a name for your Encryption field in the **Database Field Name** field, such as ENCRYPT.
3. Click the **Add** button to add the Encryption field to the **Names List**.
4. Click the **Field Information** tab.
5. Click the down arrow of the Name field and select the Encryption field.
6. Set the **Type** field to **Filter** and the **Data Type** to **Small Int (2)**.
7. Click to select the **Encryption** checkbox.
8. Click OK to update the application group.

Encryption is now enabled for your application group.

Backing up your Content Manager OnDemand instance

You must backup your Content Manager OnDemand instance database, as well as the keystore database and stash file on Multiplatforms servers, any time you modify the master key. Failure to backup this data could prevent any data stored in the Content Manager OnDemand instance from being accessible.