Content Manager OnDemand for z/OS
Version 10 Release 5

Configuration Guide

SC19-3363-03

Note

Before using this information and the product it supports, read the information in “Notices” on page 303.

This edition applies to IBM® Content Manager OnDemand for z/OS®, Version 10 Release 5 Modification 0 (program number 5697-CM1) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

ibm.com® and related resources

Product support and documentation are available from ibm.com®.

Support and assistance

From ibm.com, click **Support & downloads** and select the type of support that you need. From the Support Portal, you can search for product information, download fixes, open service requests, and access other tools and resources.

IBM Knowledge Center

See your online product information in IBM Knowledge Center at https://www.ibm.com/support/knowledgecenter/SSQHWE.

PDF publications

See the following PDF publications for your product at https://www.ibm.com/support/pages/node/1079043.

Contacting IBM

For general inquiries, call 800-IBM-4YOU (800-426-4968). To contact IBM customer service in the United States or Canada, call 1-800-IBM-SERV (1-800-426-7378).

For more information about how to contact IBM, including TTY service, see the Contact IBM website at http://www.ibm.com/contact/us/.

Part 1. Before you begin

Review information before you begin installing the Content Manager OnDemand software and configuring the Content Manager OnDemand system.

Chapter 1. What you should know first

Before reviewing the installation and configuration information, verify that you have knowledge of or access to people that have knowledge of prerequisite and corequisite software, as well as access to that software.

Be familiar with or have access to resources knowledgeable of:

- Administering z/OS
- Administering UNIX System Services
- TCP/IP networking protocols that will be required for clients and servers to communicate
- RACF® (optional)
- DB2®
- OAM (optional)
- VSAM (optional)
- Any other file systems that Content Manager OnDemand will access
- How to run shell scripts using TSO (optional)
- Any other operational requirements for the system. For example, you might need to configure maintenance tasks to backup databases, cache storage, and archive storage
- Infoprint Server (optional)

Chapter 2. Installation and configuration overview

To install the Content Manager OnDemand software, you must configure the server control files, create and initialize the database, verify the installation, and prepare the system for use.

About this task

To make it easier to track your progress, print a copy of the list and mark the tasks you have completed.

Procedure

To install and configure the Content Manager OnDemand software, do the following tasks:

1. Review the *IBM Content Manager OnDemand for z/OS: Introduction and Planning Guide*.
2. Review all of the information in Part 1, “Before you begin,” on page 1
3. Review the current Preventive Service Planning (PSP) information.

   PSP buckets are identified by UPGRADEs, which identify product levels and SUBSETs, which specify the FMIDs for a product level. The UPGRADE value for Content Manager OnDemand is ODMP10. The SUBSET value for Content Manager OnDemand is H272A10.
4. Determine the type of system configuration that you need to install.

   To learn more about system configurations, see Chapter 3, “Server configurations,” on page 9.
5. Verify that your site has installed the required and optional hardware and software products.

   For a list of the Content Manager OnDemand prerequisites, see Chapter 4, “Hardware and software requirements,” on page 15.
6. Contact the IBM support center for the latest maintenance levels of the required and optional software, including z/OS, DB2, and Content Manager OnDemand.
7. Print, read and perform any required tasks in Part 2, “Installing the server software,” on page 17.
8. Install the Content Manager OnDemand software.

   For installation instructions, see Part 2, “Installing the server software,” on page 17.
9. Configure the server control files.

   For an overview of this task, see Part 3, “Configuring a single instance of Content Manager OnDemand,” on page 31.
   a) Copy the control files.

      For more information, see Chapter 11, “Copy server control files,” on page 33.
   b) Verify the `ARS.INI` file.

      For more information, see Chapter 12, “Verify the ARS.INI file,” on page 35.
   c) Verify the `ARS.CFG` file.

      For more information, see Chapter 13, “Verify the ARS.CFG file,” on page 39.
   d) Modify the `ARS.CACHE` file.

      For more information, see Chapter 14, “Modify the ARS.CACHE file,” on page 53.
   e) Optional: Verify the LDAP configuration files.

      For more information, see Chapter 15, “Configure LDAP (optional),” on page 55.
   f) Optional: Verify the `CLI.INI` file.

      For more information, see Chapter 16, “Verify the CLI.INI file,” on page 59.
   g) Modify the ARSSOCKD procedure.

      For more information, see Chapter 17, “Modify the ARSSOCKD procedure,” on page 61.
   h) Modify the ARSLOAD procedure.

      For more information, seeChapter 18, “Modify the ARSLOAD procedure,” on page 63.
10. Create and initialize the database.

For an overview of this task, see Part 4, “Creating and initializing the database,” on page 65.

a) Create the storage group and database.

For more information, see Chapter 19, “Creating the storage group and database,” on page 67.

b) Create a table space for the Content Manager OnDemand system tables.

For more information, see Chapter 20, “Creating the table spaces,” on page 69.

c) Create the Content Manager OnDemand system tables.

For more information, see Chapter 21, “Creating the Content Manager OnDemand system tables,” on page 81.

d) Initialize the system log component.

For more information, see Chapter 22, “Initializing the system log,” on page 83.

e) Optional: Initialize the system load logging facility by running the ARSSYSCR program.

For more information, see Chapter 23, “Initializing the system load logging facility,” on page 85.

f) Initialize the system migration component.

For more information, see Chapter 24, “Initializing system migration,” on page 87.

11. Prepare the system for use.

For an overview of this task, see Part 7, “Preparing the system for use,” on page 113.

a) Define storage sets.

You must define storage sets before you can add application groups or load data into the system. For more information, see Part 7, “Preparing the system for use,” on page 113.

b) Configure the System Log application group.

Before you define reports to the system, load data, or let users access the system, IBM recommends that you configure the System Log application group. For more information, see Chapter 36, “Configuring the System Log application group,” on page 117.

c) Optional: Configure the System Load application group.

For more information, see Chapter 37, “Configuring the System Load application group,” on page 119.

d) Configure the System Migration application group.

If you plan to migrate index data to archive storage, then you must configure the System Migration application group. For more information, see Chapter 38, “Configuring the System Migration application group,” on page 121.

e) Backup the Content Manager OnDemand database.

After configuring the system, IBM recommends that you create a full backup image of the Content Manager OnDemand database. For more information, see Chapter 39, “Back up the Content Manager OnDemand database,” on page 123.

12. Verify the installation.

For an overview of the steps in this task, see Part 8, “Verifying the installation,” on page 127.

a) Restart the system.

b) Start the server program (ARSSOCKD).

c) Install at least one of the Content Manager OnDemand client programs.

For information about installing the Content Manager OnDemand Windows client software, see the *IBM Content Manager OnDemand: Client Installation Guide*.

d) Log on to the library server with the Content Manager OnDemand client program.

13. Do any optional configuration tasks.

- Configure multiple instances of Content Manager OnDemand as described in Chapter 44, “Defining multiple instances,” on page 141.
- Configure the server print component as described in Chapter 45, “Server print,” on page 147.

- Configure server programs to start automatically and run on a regular schedule as described in Chapter 46, “Server tasks,” on page 155.
- Enable large object support as described in Chapter 47, “Migrated indexes,” on page 157.
- Configure the system to delete OAM and VSAM objects by using SMF records and the ARSEXPIR program as described in Chapter 48, “Storage manager expiration (deleting OAM and VSAM objects),” on page 165.
- Configure the following Content Manager OnDemand exits:
  - Configure the Content Manager OnDemand system log user exit as described in Chapter 49, “System log exit,” on page 173.
  - Configure the Content Manager OnDemand security system interface exit as described in Chapter 50, “User security exit,” on page 179.
  - Configure the Content Manager OnDemand command line programs (such as ARSLOAD) to run without specifying a userid and password as described in Chapter 51, “Unified login (ARS.PTGN) exit,” on page 201.
  - Configure the system to allow CICS® users to logon without a password as described in Chapter 52, “CICS® client unified login,” on page 205.
  - Configure the system to modify some of the parameters that are used by Content Manager OnDemand when document data is being captured (loaded) by the ARSLOAD program as described in Chapter 53, “Report specifications archive definition exit,” on page 207.
  - Configure the Content Manager OnDemand client preview user exit, an exit point that allows an installation to modify documents before they are presented to clients as described in Chapter 54, “Client preview exit,” on page 217.
  - Configure the exit point that is called when Content Manager OnDemand creates table spaces, tables, and indexes for the Content Manager OnDemand data tables as described in Chapter 55, “Table space creation exit,” on page 225.
- Do the following tasks to install and configure the Structured APIs:
  - Review the structured API installation requirements described in Chapter 56, “Structured APIs overview,” on page 231.
  - Install the structured APIs as described in Chapter 57, “Content Manager OnDemand structured API installation instructions,” on page 235.
  - Review the API request and response code messages described in Chapter 58, “Using the structured APIs,” on page 249.
  - Review the sample drivers described in Chapter 59, “Structured APIs sample drivers,” on page 283.

Chapter 3. Server configurations

Before you install and configure Content Manager OnDemand, learn about the different types of configurations and the various components that make up the system.

The following figure shows an overview of the Content Manager OnDemand system architecture:

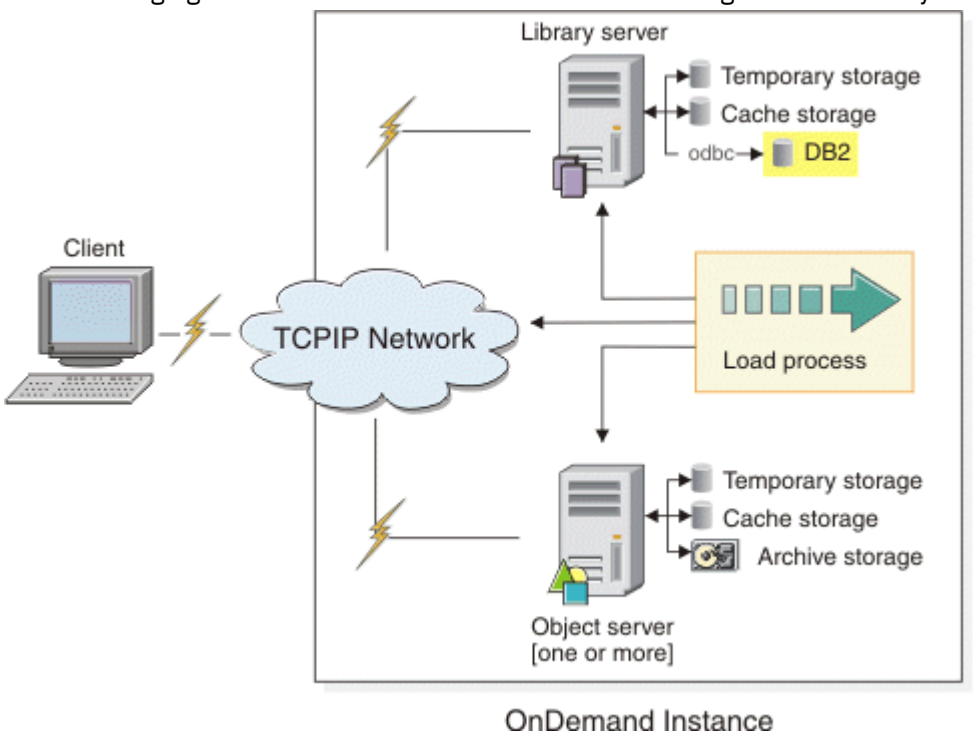

Figure 1. Overview of the Content Manager OnDemand system architecture

An operational Content Manager OnDemand system is referred to as a Content Manager OnDemand instance. The Content Manager OnDemand instance is composed of a single library server and one or more object servers. The library server manages a database of information about the various Content Manager OnDemand objects, such as users, groups, folders, and stored report index information. The object server manages the reports on disk, optical, and tape storage devices.

To the Content Manager OnDemand clients, the instance appears as a single entity. The clients connect to the instance through a single port and IP address, there is a single Content Manager OnDemand database (composed of system and AG data tables), and a single archive storage database that stores all reports and documents.

Reports are loaded into the Content Manager OnDemand system through a load process. The load process can be called locally (from the system on which the Content Manager OnDemand instance is installed) or remotely. Local and remote loading connectivity can be achieved by using TCP/IP. In the case of local loading, you can bypass TCP/IP and load data directly into the Content Manager OnDemand databases.

The Content Manager OnDemand server code for the library server and object servers is distributed as two separate modules:

ARSSOCKD
: This module provides a combined library and object server. You can use the ARSSOCKD module to create a single Content Manager OnDemand system that performs both the library server and object server functions.

ARSOBJD
This module is an object server only module. You can configure a system that contains one ARSSOCKD module and multiple ARSOBJD modules.

The object and library servers are composed of multiple modules, but the ARSSOCKD and ARSOBJD modules are the two modules that determine the server functionality.

Library server

The Content Manager OnDemand library server uses a relational database manager to manage objects and provide data integrity by maintaining index information and controlling access to objects stored on one or more object servers.

The library server directs requests from clients to query, retrieve, and print items in the database, which contains object indexes and other information. The library server routes requests to the appropriate object server to store, retrieve, and delete objects.

The library server is part of a Content Manager OnDemand instance. Each Content Manager OnDemand instance:

- Has its own folders, application groups, applications, users, groups, storage sets, and printers
- Must run in a single code page
- Has different security (users, groups, folder and application group permissions)
- Must have its name specified on commands if it is not the default instance
- Has its own system log

Some reasons to have multiple instances on the same system are:

- To have different test and production environments
- To have databases that use different code pages

The library server is the central component of the Content Manager OnDemand system. The library server maintains the central database of the objects on the system. The objects are: users, groups, storage sets, storage nodes, printers, application groups, applications, and folders. The library server uses DB2 to manage the database. The standard configuration for the library and object server includes the database and cache storage on one system or node. The library server processes the login requests from the clients, handles the queries from the clients, and maintains the database.

Object server

Content Manager OnDemand stores and retrieves objects that reside on an object server through requests routed by the library server. An object server is the repository for objects stored on the system. The object server maintains reports that are stored in cache or archive storage. If an object server is configured with archive storage, the object server works with OAM, Tivoli® Storage Manager, or VSAM to manage reports in cache storage and in archive storage. The object server processes data loading operations, document retrieval requests from the clients, and migration and expiration processing.

The object server manages storage resources that are defined through the Content Manager OnDemand administrative programs. A Content Manager OnDemand system can have many object servers distributed across multiple interconnected networks to provide convenient user access. Object servers can run on any supported Content Manager OnDemand environment including AIX®, Solaris, Windows, and z/OS systems.

Object servers work with the Content Manager OnDemand administrative programs to efficiently manage storage resources. This allows the Content Manager OnDemand administrator to specify the type of archive (HFS, VSAM, OAM, Tivoli Storage Manager), how long documents reside on specific device types before the documents are migrated to another device, and how long Content Manager OnDemand maintains documents on the system before expiring them.

An object server is the component of a Content Manager OnDemand system that holds the reports that are accessed by users. An object server belongs to a Content Manager OnDemand instance. An instance is a logical server environment consisting of a library server, one or more object servers, a database, and cache storage. A Content Manager OnDemand object server:

- Has its own storage nodes
- Must run in the same code page as the library server
- Uses the security from the library server

Some reasons to have more than one object server are:

- To distribute the storage of data across multiple systems or locations
- To be able to load data into more than one storage node at a time

Content Manager OnDemand installation scenarios

In a z/OS environment, there are three basic library and object servers installation scenarios. Other installation scenarios are possible, but such scenarios are typically customized to particular customer requirements and are beyond the scope of this information.

The selection of a particular installation configuration is based on your needs with respect to the following variables:

- Quantity of data to be stored
- Length of time for which the data and indexes need to be stored
- Archive and storage subsystems
- Storage and retrieval workloads (number of concurrent users)
- High availability and redundancy requirements
- Performance requirements
- Other customer specific requirements

Single LPAR configuration

The single LPAR configuration is composed of a single instance of the combined library and object server ARSSOCKD module.

The following figure shows how a single ARSSOCKD server provides the complete Content Manager OnDemand functionality.

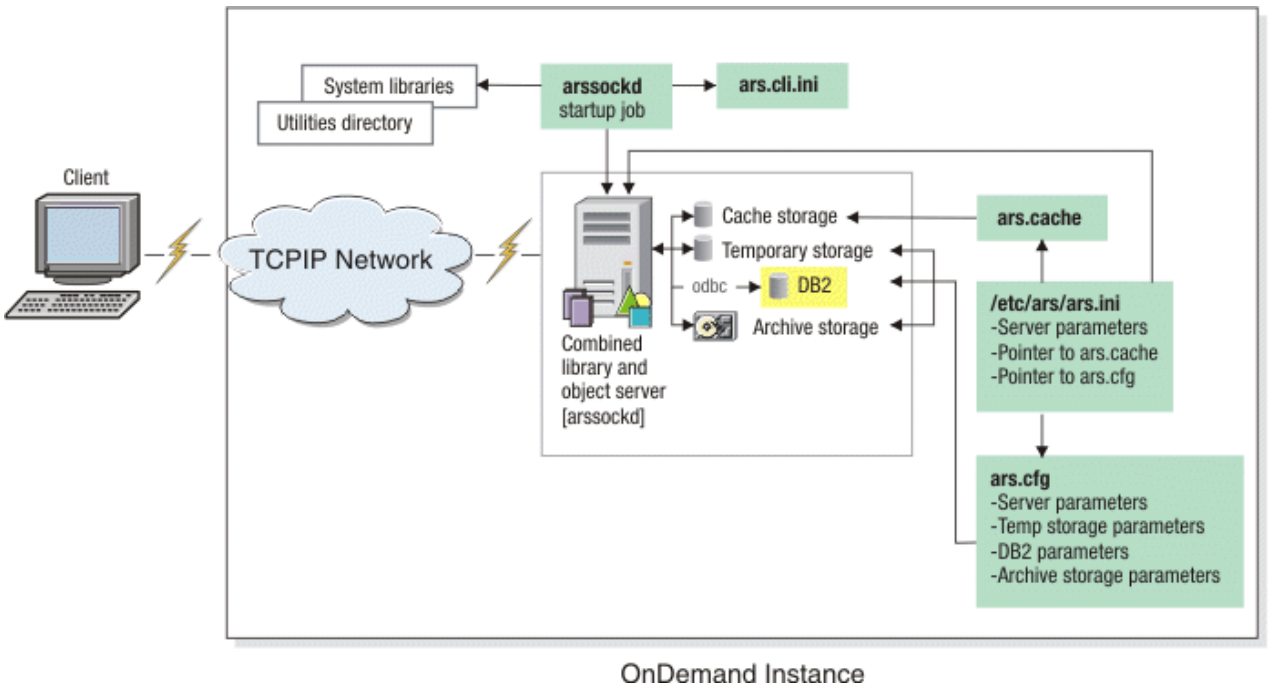

Figure 2. Single LPAR configuration

This configuration is set up using a single set of the following configuration files:

`ars.ini`
The primary configuration file that identifies the name of the Content Manager OnDemand server instance, defines basic server parameters, and contains pointers to the other configuration files. This file is common to all Content Manager OnDemand servers that are defined on the system or LPAR.

`ars.cfg`, `ars.cache`, `ars.cli.ini`, and arssockd startup job
Contain parameters that are specific to a particular instance of Content Manager OnDemand. The `ars.cache` file contains the cache storage system parameters. The `ars.cfg` file contains multiple sets of parameters that are related to the server, temporary storage, database, and archive storage interface.

Multiple LPAR configuration

The multiple LPAR configuration consists of a single ARSSOCKD library server modules and multiple ARSOBJD object server modules.

The object servers can be installed on separate LPARS or separate systems. These separate systems can be on z/OS, AIX or any other platform that is supported by Content Manager OnDemand. This setup allows for horizontal scalability with the ability to use multiple technologies as appropriate. The only constraint is that clients must have access to all systems through TCP/IP. Each Content Manager OnDemand server has its own set of configuration files. The parameters in all configuration files must be set such that all the servers are part of the same instance. The Content Manager OnDemand clients connect to the IP address listening port of the ARSSOCKD library server module. The documents are retrieved from the various object servers based on location information returned by the library server. The fact that there are multiple object servers, possibly at multiple locations, is not apparent to the client systems.

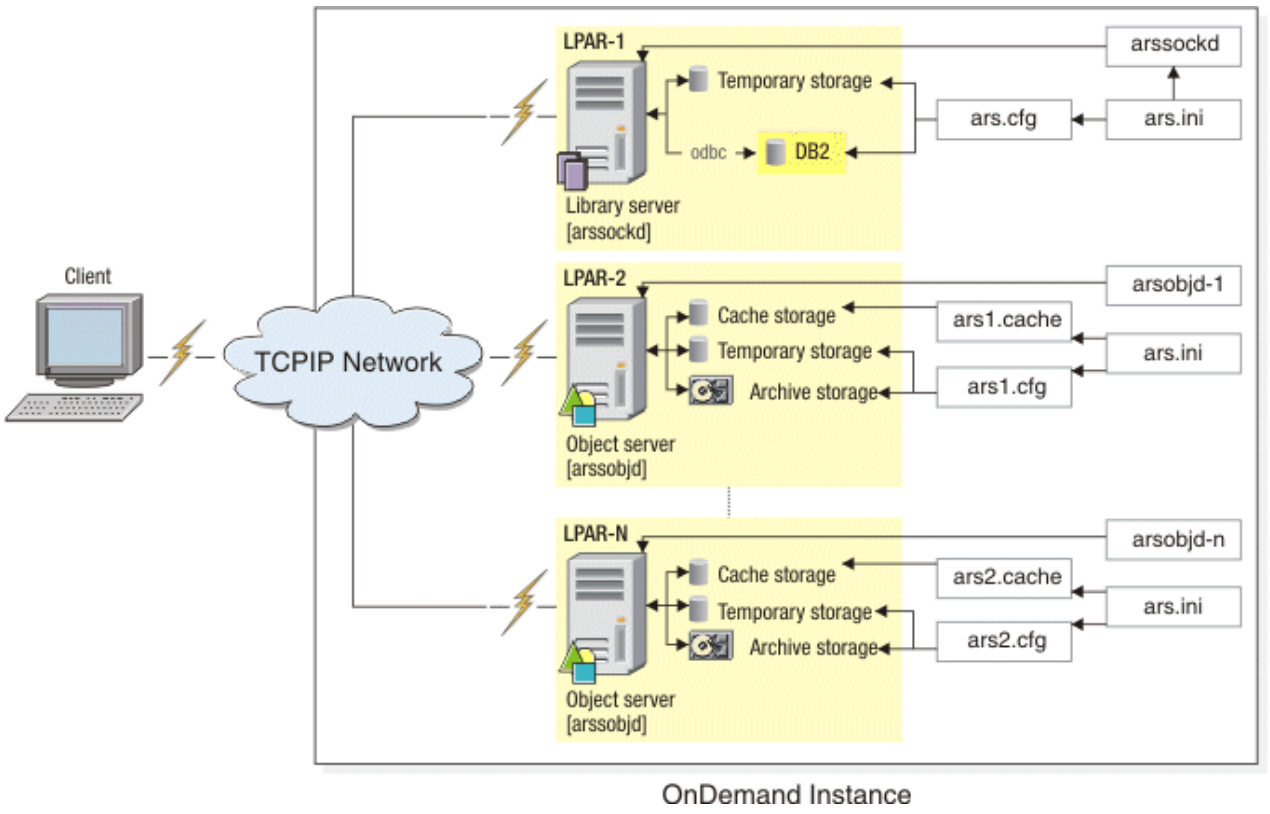

Figure 3. Multiple LPAR configuration

Parallel Sysplex, multiple LPAR configuration

The Parallel Sysplex® configuration is composed of multiple ARSSOCKD combined library and object servers modules. This type of installation is unique to z/OS systems and is based on Parallel Sysplex technology.

Although each combined library and object server is installed in a separate LPAR, the Parallel Sysplex technology enables the servers to share the same configuration files, database, JES, HFS, and archive. For performance reasons, all HFS read/write directories that are used for temporary storage of data are configured as being unique to each ARSSOCKD server.

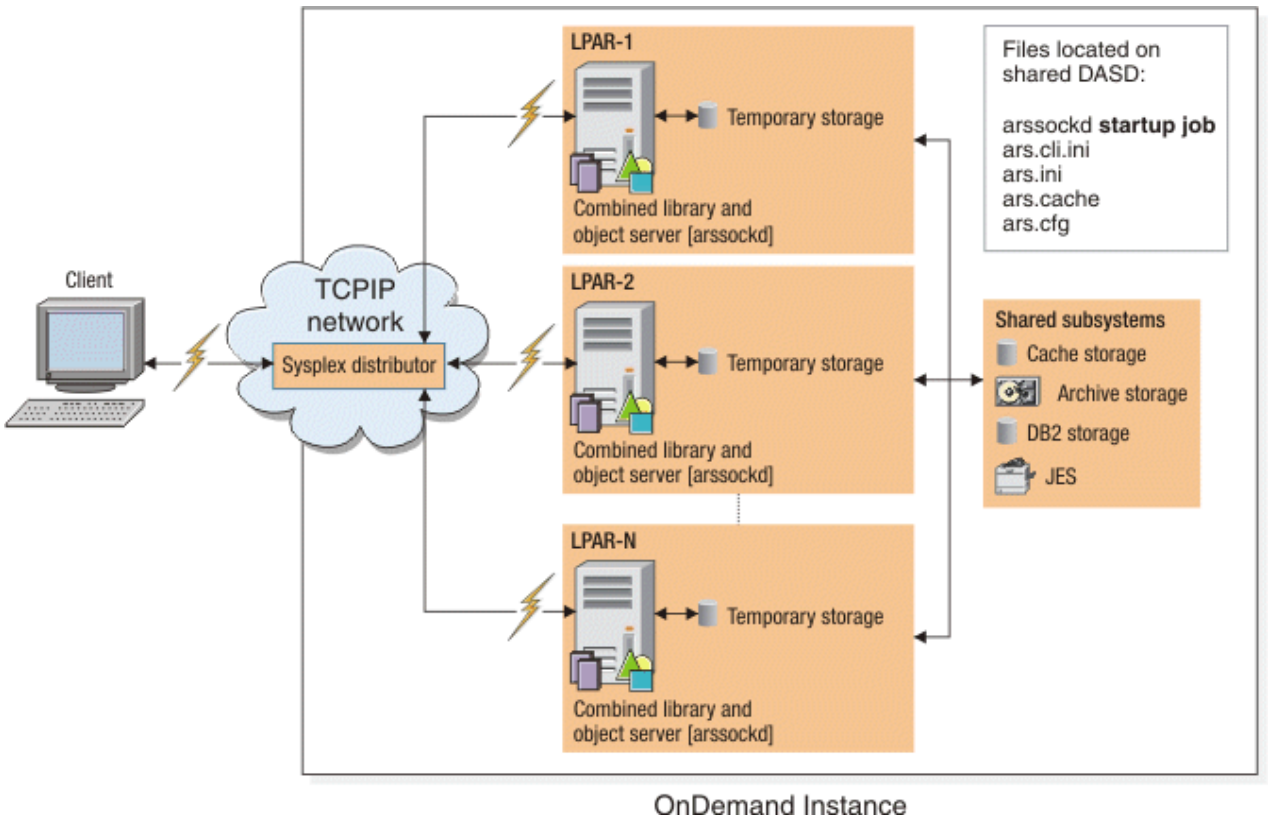

Figure 4. Parallel Sysplex configuration

This setup has the advantages of vertical scalability within the parallel sysplex as well as high availability because the failure of any single LPAR only affects the current transactions and a retry sent by the client is automatically routed to one of the active LPARs. The clients connect to the system using a single IP address and port that are associated with a device or program that routes the request to an appropriate ARSSOCKD server. Some systems can route the request according to the z/OS Work Load Manager suggestions, while other devices simply route the requests following a round robin routine.

Chapter 4. Hardware and software requirements

For all information about hardware and software prerequisites for Content Manager OnDemand, go to http://www.ibm.com/support/docview.wss?uid=swg27049168 .

Part 2. Installing the server software

Chapter 5. RACF, UNIX System Server IDs, Groups

To complete the installation, the user ID running the installation must have access to specific resources. You can manage the access with RACF profiles and groups.

The following keywords can be substituted at install time:

ODADMIN
The ID that is used for installing the Content Manager OnDemand product software.

ARS
A RACF owning group profile.

ARSSERVR
A server owning ID.

- All RACF groups and users must have OMVS segments assigned.
- A DS profile ARS.** to protect the HFS data sets, ARS group is given ALTER access.
- A server owning ID named ARSSERVR, connected to ARS group. Assign the OMVS home of `/tmp` to this ID. The `/tmp` directory is a working directory for log files, print data, and other temporary data.

The ARSSERVR ID is set as the owner of the DB2 instance. This ID will need the appropriate DB2 access, such as DBADM or SYSADM, for the creation of the storage group, database, table spaces, and so on. The DBA can adjust the access as required for the ARSSERVR ID, after the installation is completed.

Chapter 6. UNIX System Services profile

The system programmer must set and export the STEPLIB environment variable to the location of the latest Content Manager OnDemand code. This step is required if the user plans to run the ARSADMIN or ARSLOAD programs from a UNIX System Services shell.

Set and export the STEPLIB environment variable to the location of the latest Content Manager OnDemand code. This can be done in `/etc/profile` or in a `.profile` that gets executed when the user ID requiring this STEPLIB logs on. The following line shows an example:
`STEPLIB=ARS.V10R5M0.SARSLOAD:$STEPLIB`

Check with your system programmer to determine the best way to implement this requirement.

Chapter 7. Started tasks

The servers to be installed have STC (started task control) names of ARSSOCKD and ARSLOAD, so there are RACF STARTED profiles created for ARSSOCKD.* and ARSLOAD.*, which assign the user ID of ARSSERVR and the group of ARS.

Chapter 8. HFS data sets

The Content Manager OnDemand system requires several HFS data sets to store product code, report data, and temporary files (for example, logs).

The following HFS data sets are created:

- ARS.PRODUCT.HFS with a mount point at `/usr/lpp/ars/V10R5M0` for the product code. See the *Program Directory for IBM Content Manager OnDemand Servers for z/OS, Version 10.5* for the exact storage requirements.
- ARS.SERVER.HFS with a mount point at `/ars1` for the server data, with a size that is large enough to store the report data. The exact size is a function of the quantity of data that will be stored on the system. To improve storage and retrieval times, use multiple file systems, each with its own mount point. You can name these additional file as follows: ARS.SERVER2.HFS with a mount point at `/ars2`, ARS.SERVER3.HFS with a mount point at `/ars3`, and so on.
- ARS.TMP.HFS with a mount point at `/tmp` for temporary work space, including log files and server print data. Start with a minimum of 1000 tracks, although the exact size is a function of the number and type of logs, the amount of server print activity, and so on.

The HFS data sets are owned by the ARSSERVR ID and the ARS group, with a mode of 775.

Important:

Do not change the HFS directory structure `/usr/lpp/ars/V10R5M0`. If you want to use an alternate mount point, ensure that all the directories under the `/usr/lpp/ars/V10R5M0` directory are available at the location where the HFS is mounted. For example, if you take the HFS that is mounted at the `/usr/lpp/ars/V10R5M0` directory, and mount it at the directory `/xxx`, you get new directories, such as `/xxx/bin`, `/xxx/bin/exits`, and `/xxx/config`.

Chapter 9. ACIF

If you plan to convert or index reports with ACIF, then you must install the ACIF software on the system.

The ACIF product and program directory are included with the Content Manager OnDemand program package. If they are not, contact your IBM representative.

Important: The Loadlib that contains the enhanced ACIF program must be APF Authorized.

Chapter 10. Installation paths

You must install the Content Manager OnDemand software using the SMP installation process.

The following table describes the locations where Content Manager OnDemand software is installed.

Table 1. Component location

Component Location	Description
`ARS.V10R5M0.SARSINST`	Contains the installation jobs and the system-wide procedures such as ARSLOAD and ARSSOCKD.
`ARS.V10R5M0.SARSLOAD`	The z/OS executables (loadlib). The SARSLOAD data set must be APF Authorized.
`/ars1`, `/ars2`, `/ars3`, and so on	Cache storage file systems used for storage of the reports that you load into the system.
`/tmp`	Storage space for the error files and message files that are produced by the Content Manager OnDemand programs, including ARSSOCKD and ARSLOAD and temporary server print data and other temporary data. **Note:** The permissions for this file system must be drwxrwxrwt. Use the **CHMOD** command to set the permissions. For example: `chmod 1777 /ars/tmp`.
`/etc/ars`	The production configuration and JCL files, including `ars.cache`, `ars.cfg`, `ars.ini`, `arslog`, and `arsprtjcl`.
`/usr/lpp/ars/V10R5M0/bin`	The executable programs.
`/usr/lpp/ars/V10R5M0/bin/font1`	The AFP font files.
`/usr/lpp/ars/V10R5M0/bin/font1/maps`	The AFP font map files (`*.cp`).
`/usr/lpp/ars/V10R5M0/config`	A symbolic link to the `/etc/ars` directory.
`/usr/lpp/ars/V10R5M0/samples`	Sample configuration and JCL files. **Note:** In Chapter 11, “Copy server control files,” on page 33, you will copy the sample files from this location to the `/etc/ars` directory.

Part 3. Configuring a single instance of Content Manager OnDemand

You configure a single instance of Content Manager OnDemand by configuring the control files that are used to initialize and operate the server.

About this task

You can mount the Content Manager OnDemand installation directory at some point besides `/usr/lpp/ars/V10R5M0`. This support allows installations to run at different service levels. For example, SMP might be used to install into `SERVICE/usr/lpp/ars/V10R5M0`. `SERVICE/usr/lpp/ars/V10R5M0` might be copied into `/usr/lpp/ars/V10R5M0/alt` for testing. When testing is complete, the `/usr/lpp/ars/V10R5M0/alt` might be copied into `/usr/lpp/ars/V10R5M0` for production.

By default, programs started with JCL search `/usr/lpp/ars/V10R5M0` and any subdirectories for files that are needed. For example, `/usr/lpp/ars/V10R5M0/locale`. To use an alternate location to locate these file, an ARSBIN DD statement might be specified that has the path for the Content Manager OnDemand `bin` directory. For example, `//ARSBIN DD PATH='/usr/lpp/ars/V10R5M0/alt/bin'`. This command causes Content Manager OnDemand to use, for example, the `/usr/lpp/ars/V10R5M0/bin/alt/locale` locale instead of the `/usr/lpp/ars/V10R5M0` locale.

Programs started from a UNIX System Services shell locate necessary files relative to where they were invoked from. For example, running `/usr/lpp/ars/V10R5M0/bin/arsdoc` uses `/usr/lpp/ars/V10R5M0/alt/locale`.

Be careful if the Content Manager OnDemand `bin` directory is in the path. The relative location is determined based on where the UNIX System Services actually fetches the file. For example, if `PATH=/usr/lpp/ars/alt/V10R5M0/bin:/usr/lpp/ars/V10R5M0/bin` and an unqualified command like **arsdoc** is used, the copy of arsdoc from `/usr/lpp/ars/V10R5M0/alt/bin` is fetched, even if the current working directory is `/usr/lpp/ars/V10R5M0/bin`. This causes arsdoc to use `/usr/lpp/ars/V10R5M0/alt/locale`. If the arsdoc from the current directory should be used, specify `./arsdoc`.

This support requires that you copy the entire Content Manager OnDemand directory structure before you copy the executable files to random directories. For example, if `/usr/lpp/ars/V10R5M0/alt/bin/arsdoc` was copied to `/u/fred/arsdoc`, and `/u/fred/arsdoc` was executed, the `/usr/lpp/ars/V10R5M0` directory is used for any necessary files, and unpredictable results might occur.

It is strongly recommended that all configuration directories be linked to the same directory, and that they use the same `ars.ini` file. This setup reduces the possible duplication of definitions. For example, both `/usr/lpp/ars/V10R5M0/alt/config` and `/usr/lpp/ars/V10R5M0/config` point to the same directory. The `ars.ini` file has two instances defined, ARCHIVE and TEST. The normal **ARSSOCKD** command uses the `/usr/lpp/ars/V10R5M0` directory. The TEST ARSSOCKD PROC command has a PARM='TEST', and a `//ARSBIN DD PATH='/usr/lpp/ars/V10R5M0/alt/bin'`. Similarly, the ARSLOAD command used for production specifies a **-h ARCHIVE** in the parm or be omitted. The **ARSLOAD** command for test specifies a **-h TEST** parm, and have a `//ARSBIN DD PATH='/usr/lpp/ars/V10R5M0/alt/bin'`.

Chapter 11. Copy server control files

Each instance must have its own set of server configuration files. Make two copies of these files: one set for backup, the other set to configure the instance.

Procedure

Do the following steps:

1. Make an `/etc/ars` directory.
   For example: `mkdir /etc/ars` .
2. Change the access mode for the `/etc/ars` directory by entering the following command: `chmod 755 /etc/ars`
3. Create a symbolic link from the `/usr/lpp/ars/V10R5M0/config` directory to the `/etc/ars` directory.
   For example: `ln -s /etc/ars /usr/lpp/ars/V10R5M0/config`.
4. Make an `/ars/save` directory.
   For example: `mkdir /ars/save`.
5. Change the access mode for the `/ars/save` directory by entering the following command: `chmod 755 /ars/save`
6. Make two copies of the Content Manager OnDemand server control files (found in the `/usr/lpp/ars/V10R5M0/samples` directory) that are listed in Table 2 on page 33. Make one copy (the backup copy) to the `/ars/save` directory. Make a second copy (the working copy) to the `/etcm/ars` directory.

   Table 2. Content Manager OnDemand server control files to save

File	Purpose
`ars.cache`	List of the cache storage file systems
`ars.cfg`	Content Manager OnDemand server configuration file
`ars.ini`	Names of and configuration information for Content Manager OnDemandinstances
`arsprtjcl`	Infoprint server printing JCL. This file should be a copy of `/usr/lpp/ars/V10R5M0/samples/arsprtjcl1`. The installation needs to rename the copy to `/etc/ars/arsprtjcl`. See "Configuring Infoprint server printing" in Chapter 45, “Server print,” on page 147 for more information.
`arsodf.xml`	ODF email xml file. Only required if ODF email capabilities will be used.
`arsxform.xml`	UTF-8 transformation configuration file.

7. Run the **CHMOD** command to change the access mode for the working copy of the files.
   For example:`chmod 644 /etc/ars/*.*`

Chapter 12. Verify the ARS.INI file

The `ars.ini` file contains information about Content Manager OnDemand instances. Review the files to determine that the parameters specify information that is correct for your Content Manager OnDemand system

About this task

Note: The ARS.INI file must be in code page 1047. That is, the delimiters in the header line for the instances must be X'AD' (left bracket character) and X'BD' (right bracket character).

A Content Manager OnDemand instance is a logical server environment made up of a database, a library server, and one or more object servers. An instance is defined in the `ars.ini` file by naming the instance, identifying the name of the database used by the instance, and identifying the library server on which the database will be maintained. An instance has its own cache storage file systems. The cache storage file systems are defined in the `ars.cache` file on each object server.

When you install the Content Manager OnDemand software, the `ars.ini` file contains information about the default instance named ARCHIVE. The following shows an example of the `ars.ini` file.

```
[@SRV@_ARCHIVE]
HOST=
PROTOCOL=2
PORT=0
SRVR_INSTANCE=ARSDBASE
SRVR_INSTANCE_OWNER=ARSSERVR
SRVR_OD_CFG=/usr/lpp/ars/V10R5M0/config/ars.cfg
SRVR_SM_CFG=/usr/lpp/ars/V10R5M0/config/ars.cache
```

Verify that the parameters in the `ars.ini` file specify information that is correct for your Content Manager OnDemand system. To verify the file, log on as the super user or as any user set up with appropriate permissions. Change to the `/usr/lpp/ars/V10R5M0/config` directory. Use a standard text editor to modify the file.

Note:

1. For a distributed library / object server system, configure one copy of the `ars.ini` file on each server that is part of the Content Manager OnDemand system.
2. If you change the default instance from ARCHIVE to another value, it is recommended that you specify the instance on the **arsdb** command. Otherwise, you might receive this error: `arsdb unable to determine database engine`

Server parameters

[@SRV@_ARCHIVE]

Identifies the header line of an instance stanza. The header line contains a string that identifies the name of the instance. By default, the first or only instance in the `ars.ini` file is named ARCHIVE.

HOST parameter

When configuring the `ars.cfg` file on an object server, specifies the IP address, TCP/IP host name alias, or fully qualified TCP/IP host name of the library server. When configuring a library server, set the value of the **HOST** parameter to null (`HOST=`).

PORT parameter

Identifies the TCP/IP port number that the server monitors for client requests. A value of 0 means that the server monitors port number 1445. If you plan to use a port number other than 1445 on the library server, then enter that number instead of 0.

If you plan to run more than one instance of Content Manager OnDemand on the same system, then you must specify a different port number for each instance.

PROTOCOL parameter

Identifies the communications protocol that the server uses to communicate with clients. The only valid value is 2 for TCP/IP.

SRVR_INSTANCE parameter

The **SRVR_INSTANCE** parameter identifies the name of the Content Manager OnDemand instance. The name of the Content Manager OnDemand instance should match the name of the database that you specify for the ARSDB2 job and the ARSTSPAC job.

SRVR_INSTANCE_OWNER parameter

Identifies the name of the instance owner. This value should always be set to the super user or any other user that has the appropriate permissions. Also, the name of the Content Manager OnDemand instance owner should match the SQLID that you specify for the ARSDB2 job and the ARSTSPAC job.

SRVR_OD_CFG parameter

Identifies the location and name of the `ars.cfg` file that is used by the server.

SRVR_SM_CFG parameter

Identifies the location and name of the `ars.cache` file that is used by the server. The `ars.cache` file specifies the cache storage file systems that are used by the storage manager.

Secure Socket Layer (SSL) parameters

SSL_PORT parameter

Specifies the port number on the Content Manager OnDemand server dedicated to communicating with the SSL protocol. To set the Content Manager OnDemand server to only use the SSL protocol, also set the PORT parameter in the ARS.INI file to a value of -1 (PORT=-1). Specify 0 to indicate that no port on the Content Manager OnDemand server communicates with the SSL protocol.

SSL_KEYRING_FILE parameter

Identifies the HFS key database file, RACF key ring, or z/OS PKCS#11 token that is used by the server. For example, `SSL_KEYRING_FILE=/etc/ars/arssockd.kdb` specifies an HFS key database file, `SSL_KEYRING_FILE=ARSUSER/ARSSOCKD.SSLRING` specifies a RACF key ring named `ARSSOCKD.SSLRING` that is owned by user ARSUSER, and `SSL_KEYRING_FILE=*TOKEN*/SSLTOKEN` specifies a PKCS#11 token.

SSL_KEYRING_STASH parameter

Identifies the location and name of the SSL Keyring stash file that is used by the server. For example, `SSL_KEYRING_STASH=/usr/lpp/ars/V10R5M0/config/ondemand.sth`.

Restriction: Do not specify the **SSL_KEYRING_STASH** parameter if you use an SAF key ring or z/OS PKCS#11 token.

SSL_KEYRING_LABEL parameter

Specifies the label of the SSL key.

SSL_CLNT_USE_SSL parameter

Specifies whether the server-side clients (for example, ARSDOC, ARSMAINT, or ARSLOAD) must communicate with the SSL protocol. Specify 0 to indicate that the clients do not communicate with the SSL protocol. Specify 1 to indicate that the clients must communicate with the SSL protocol.

Exit parameters

SRVR_FLAGS_DOCUMENT_EXIT parameter

Determines if the Content Manager OnDemand security user exit will be invoked to check document permissions.

The following statement must exist in the ARS.INI file so that arsuperm will be invoked to check document permissions: SRVR_FLAGS_DOCUMENT_EXIT=1

Note: Enabling document permission checking can greatly decrease Content Manager OnDemand performance when performing a document query.

SRVR_FLAGS_FORCE_SECURITY parameter

When you set the **SRVR_FLAGS_FOLDER_APPLGRP_EXIT** to 1, the **SRVR_FLAGS_FORCE_SECURITY** parameter determines whether the Content Manager OnDemand security user exit is invoked to check folder or application group permissions for end user client functions even if the normal Content Manager OnDemand permissions grant permission.

For example, Content Manager OnDemand does not normally invoke the permission exit for a system administrator. Setting the **SRVR_FLAGS_FORCE_SECURITY** parameter to 1 causes the exit to be called if the system administrator attempts to open the folder for viewing or using the **ARSDOC get** function. This has no effect when you use the administrative interface such as the administrative client or the **ARSXML** command.

SRVR_FLAGS_FOLDER_APPLGRP_EXIT parameter

Determines if the Content Manager OnDemand security user exit will be invoked to check folder and application group permissions.

The following statement must exist in the ARS.INI file so that arsuperm will be invoked to check folder and application group permissions: SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1

SRVR_FLAGS_SECURITY_EXIT parameter

Determines if the Content Manager OnDemand security user exit will be invoked.

The Content Manager OnDemand security user exit allows an installation to augment the security related processing of the following activities or events:

- Logon
- Change Password
- Add User ID or Delete User ID by using the Content Manager OnDemand administrative functions

The following statement must exist in the ARS.INI file so that arsusec will be invoked: SRVR_FLAGS_SECURITY_EXIT=1.

SRVR_FLAGS_SQL_QUERY_EXIT parameter

Determines if the Content Manager OnDemand security user exit will be invoked for SQL query processing.

The following statement must exist in the ARS.INI file so that arsuperm will be invoked for SQL query processing: SRVR_FLAGS_SQL_QUERY_EXIT=1

Chapter 13. Verify the ARS.CFG file

The `ars.cfg` file contains parameters related to licenses, servers, temporary storage, archive manager, and the database. For a distributed library and object server system, you must configure a separate copy of the `ars.cfg` file on each server that is part of the system.

The following example shows the content of the `ars.cfg` file:

```
ARS_ORIGINAL_CODEPAGE=
ARS_DB_TABLESPACE=ARSTSPAC=
ARS_LOCAL_SRVR=
ARS_NUM_DBSRVR=4
ARS_NUM_LICENSE=1
ARS_NUM_OAMSRVR=4
ARS_OAM_DB2SSID=DSNA
ARS_OAM_PLAN=CBRIDBS
ARS_PRINT_PATH=/ars/tmp
ARS_SRVR=
DB_ENGINE=DB2
ARS_LDAP_ALLOW_ANONYMOUS=TRUE /* Allow anonymous bind connections */
ARS_LDAP_BASE_DN=foo /* Specifies 'foo' as base distinguished name */
ARS_LDAP_BIND_ATTRIBUTE=bar /* Specifies 'bar' as bound attribute */
ARS_LDAP_BIND_MESSAGES_FILE='$ONDEMAND/LDAP/msg_string.txt' /* Specifies location \
of LDAP message string file */
ARS_LDAP_MAPPED_ATTRIBUTE=foonly /* Specifies attribute 'foonly' returned to \
OnDemand as user ID */
ARS_LDAP_PORT=389 /* Specifies port on which LDAP listens */
ARS_LDAP_SERVER=127.0.0.1 /* Specifies IP address of LDAP server */
```

You should verify that the parameters in the `ars.cfg` file specify information that is correct for your Content Manager OnDemand system. To verify the file, log on as the super user or as any user set up with appropriate permissions. Change to the `/usr/lpp/ars/V10R5M0/config` directory. Use a standard text editor to modify the file.

The ARS.CFG file might contain parameters other than those described in the following topics. However, the parameters that are described in the following topics are the only parameters that apply to a Content Manager OnDemand system that runs on z/OS.

Content Manager OnDemand database parameter

ARS_ORIGINAL_CODEPAGE parameter

This parameter specifies the code page of the Content Manager OnDemand database and is mandatory.

You must run `arsdb -u` to determine the correct setting. Do not run any other Content Manager OnDemand command until you run `arsdb -u` to determine the **ARS_ORIGINAL_CODEPAGE** setting. For newly created instances, the parameter must be the CCSID associated with database. This is either the MBCS CCSID (1208) for databases that are created with CCSID UNICODE, or the SBCS CCSID for databases created with CCSID EBCDIC.

When you run the `arsdb -u` command, the **arsdb** program displays messages ARS4012E. You can ignore the part of the message that indicates there is an error. Review the instructions in the message regarding the correct value to assign to the **ARS_ORIGINAL_CODEPAGE** parameter.

For instances created prior to version 8.5 of Content Manager OnDemand, the parameter must be the CCSID used by the server, which could be different than the database CCSID if the previously used versions of Content Manager OnDemand did not set **ARS_CODEPAGE** to the DB2 EBCDIC SBCS CCSID.

The value to specify can be determined by running the 8.5 version (or later) of `arsdb -u` without `ARS_ORIGINAL_CODEPAGE=` in the `ars.cfg` file.

ACIF parameter

ARSMVS_ACIF_WARNING_LEVEL parameter

This parameter specifies the maximum return code that is allowed from ACIF. The range is from 0 to 16. Values that are less than 0 are treated as 0. Values that are greater than 16 are treated as 16. The default is 4.

`ARSMVS_ACIF_WARNING_LEVEL=0` requires a return code 0 from ACIF to allow the load to succeed. The ARSLOAD **-h** parameter must specify an instance name for this parameter to be honored.

Db2 parameters

ARS_DB_TABLESPACE parameter

The name of the table space for the Content Manager OnDemand system tables.

This parameter is only required if you are migrating from a previous version of Content Manager OnDemand and all of your system tables are in the same table space.

ARS_DB_TABLESPACE_USEREXIT parameter

Determines if the Content Manager OnDemand table space creation exit will be invoked.

The Content Manager OnDemand table space creation exit allows an installation to take action when Content Manager OnDemand creates a table space, table, or index tables that will be used to store application index data. The exit is not called for the Content Manager OnDemand system tables.

The following statement must exist in the `ARS.CFG` file that is associated with the instance so that the arsutbl DLL can be invoked: `ARS_DB_TABLESPACE_USEREXIT=`*`absolute path name`* For the sample arsutbl, you would specify the following statement in the `ARS.CFG` file:
`ARS_DB_TABLESPACE_USEREXIT=/usr/lpp/ars/V10R5M0/bin/exits/arsutbl`

ARS_NUM_DBSRVR parameter

Determines the number of task control blocks (TCBs) that the Content Manager OnDemand library server starts to handle connections to the database.

The **ARS_NUM_DBSRVR** parameter is ignored on object servers.

ARSMVS_BPOOL_INDEX parameter

Specifies the bufferpool to use for application group data table indexes.

If not specified, the bufferpool associated with the Content Manager OnDemand database is used.

For example: `ARSMVS_BPOOL_INDEX=BP3`

ARSMVS_BPOOL_TSPACE parameter

Specifies the bufferpool to use for application group data table spaces with a row length less than 1024.

If not specified or the row length is larger than or equals to 1024, the bufferpool associated with the Content Manager OnDemand database is used.

For example: `ARSMVS_BPOOL_TSPACE=BP2`

ARSMVS_DB_DSSIZE parameter

Specifies the value to use for the DSSIZE parameter for the application group data table tablespaces. This must be a power-of-two in the range of 1 – 256.

If not specified, the application group data table tablespaces are created with a MAXPARTITIONS 1, and the PRIQTY and SEQTY are chosen based on the max rows for the application group. The INDEX is created with a PRIQTY and SECQTY that are also based on the max rows.

If the ARSMVS_DB_DSSIZE parameter specifies a valid value, that value is used in the DSSIZE N G clause when creating the application group data table tablespace. Additionally, the PRIQTY and SECQTY default to -1, and the MAXPARTITIONS is set based on the application group max rows value. Any index that is created specifies a PIECESIZE based on the max rows, and defaults to a PRIQTY and SECQTY of -1.

For example: ARSMVS_BP00L_TSPACE=BP2

ARSMVS_EXIT31_DB2_SECPERM parameter

Specifies whether the arsusec and arsuperm user exit programs will be invoked with an ODBC Db2 connection.

Valid values are 0 and 1. If not specified or if specified as ARSMVS_EXIT31_DB2_SECPERM=0, the exit programs are not invoked with an ODBC Db2 connection. If specified as ARSMVS_EXIT31_DB2_SECPERM=1, the arsusec and arsuperm exits are invoked with an ODBC Db2 connection. These ODBC Db2 connections are additional Db2 batch connections above that specified by the ARS_NUM_DBSRVR parameter.

ARSMVS_MAXROWS_INDEX_PRIQTY parameter

For application groups with a maximum rows specified, this specifies the PRIQTY that is used for the CREATE INDEX statement.

If not specified, the value calculated from the maximum rows is used.

ARSMVS_MAXROWS_INDEX_SECQTY parameter

For application groups with a maximum rows specified, this parameter specifies the SECQTY that is used for the CREATE INDEX statement.

If not specified, the value calculated from the maximum rows is used.

Restriction: If you do not specify the **ARSMVS_MAXROWS_INDEX_PRIQTY** parameter, Content Manager OnDemand ignores this parameter.

ARSMVS_MAXROWS_PRIQTY parameter

For application groups with a maximum rows specified, this specifies the PRIQTY that is used for the CREATE TABLESPACE statement.

If not specified, the value calculated from the maximum rows is used.

ARSMVS_MAXROWS_SECQTY parameter

For application groups with a maximum rows specified, this specifies the SECQTY that is used for the CREATE TABLESPACE statement.

If not specified, the value calculated from the maximum rows is used.

Restriction: If you do not specify the **ARSMVS_MAXROWS_PRIQTY** parameter, Content Manager OnDemand ignores this parameter.

ARSMVS_NOMAXROWS_INDEX_PRIQTY parameter

For application groups with **Single table for all loads** checked, this specifies the PRIQTY that is used for the CREATE INDEX statement.

If not specified, a value is calculated as if a max rows of 10 million had been specified.

ARSMVS_NOMAXROWS_INDEX_SECQTY parameter

For application groups with **Single table for all loads** checked, this specifies the SECQTY that is used for the CREATE INDEX statement.

If not specified, a value is calculated as if a max rows of 10 million had been specified.

Restriction: If you do not specify the **ARSMVS_NOMAXROWS_INDEX_PRIQTY** parameter, Content Manager OnDemand ignores this parameter.

ARSMVS_NOMAXROWS_PRIQTY parameter

For application groups with **Single table for all loads** checked, this specifies the PRIQTY that is used for the CREATE TABLESPACE statement.

If not specified, a value is calculated as if a max rows of 10 million had been specified.

ARSMVS_NOMAXROWS_SECQTY parameter

For application groups with **Single table for all loads** checked, this specifies the SECQTY that is used for the CREATE TABLESPACE statement.

If not specified, a value is calculated as if a max rows of 10 million had been specified.

Restriction: If you do not specify the **ARSMVS_NOMAXROWS_PRIQTY** parameter, Content Manager OnDemand ignores this parameter.

ARSMVS_TABLESPACE_COMPRESS parameter

This parameter indicates that Content Manager OnDemand is to include a COMPRESS clause with the value YES or NO of the CREATE TABLESPACE statement for the application group data tables.

Valid values are `ARSMVS_TABLESPACE_COMPRESS=YES` or `ARSMVS_TABLESPACE_COMPRESS=NO`. The COMPRESS clause is omitted if an invalid value is specified or if the parameter is absent.

ARSMVS_TABLESPACE_TRACKMOD parameter

This parameter indicates that Content Manager OnDemand is to include a TRACKMOD clause with the value YES or NO of the CREATE TABLESPACE statement for the application group data tables.

Valid values are `ARSMVS_TABLESPACE_TRACKMOD=YES` or `ARSMVS_TABLESPACE_TRACKMOD=NO`. The COMPRESS clause is omitted if an invalid value is specified or if the parameter is absent.

DB_ENGINE parameter

Specifies the database manager. You must specify DB2.

Exits parameter

ARS_USER_EXITS_DIR parameter

This parameter specifies a directory from which the user exits are to be loaded from instead of the `/usr/lpp/ars/V10R5M0/bin/exits` directory.

If you specify this parameter, all the exits must be in this directory.

LDAP parameters

ARS_LDAP_ALLOW_ANONYMOUS parameter

This parameter specifies whether anonymous bind connections are allowed on this LDAP server.

Valid values are TRUE and FALSE. If the value is FALSE, you must also specify an LDAP user ID and password in the stash file.

ARS_LDAP_BASE_DN parameter

This parameter specifies the base distinguished name to use.

This parameter is required.

ARS_LDAP_BIND_ATTRIBUTE parameter

This parameter specifies the attribute being bound.

This parameter is required.

ARS_LDAP_BIND_MESSAGES_FILE parameter

This parameter specifies the location of a file that contains the LDAP message strings that the Content Manager OnDemand server looks for during login.

This is used for issuing messages when the user's password is about to expire, or their LDAP account is locked. The **ARS_LDAP_BIND_MESSAGES_FILE** parameter is used with the ARSLDAP.INI file to implement this functionality.

ARS_LDAP_GROUP_FILTER parameter

This parameter is used to query LDAP for groups that will be imported into Content Manager OnDemand.

For example: (objectclass=group)

This parameter is required when using the ARSLSYNC program.

ARS_LDAP_GROUP_MAPPED_ATTRIBUTE parameter

This parameter is used to identify the LDAP field that contains the name of the group that will be created in Content Manager OnDemand.

For example: ARS_LDAP_GROUP_MAPPED_ATTRIBUTE=CN will extract the value for the LDAP field named CN and create that group in Content Manager OnDemand.

This parameter is required when using the ARSLSYNC program.

ARS_LDAP_IGN_GROUPS parameter

This parameter specifies the group IDs that Content Manager OnDemand ignores when you sync group IDs between LDAP and Content Manager OnDemand by using the ARSLSYNC program.

You can specify up to 10 group IDs, delimited by a comma.

ARS_LDAP_IGN_USERIDS parameter

This parameter specifies the user IDs that Content Manager OnDemand ignores when you enable LDAP for authentication or when you sync user IDs between LDAP and Content Manager OnDemand by using the ARSLSYNC program.

If the parameter does not exist or you do not specify a value, Content Manager OnDemand defaults to ADMIN. You can specify up to 10 user IDs, delimited by a comma. If you specify a list of user IDs and you want to include ADMIN, you must specify it on the list.

ARS_LDAP_MAPPED_ATTRIBUTE parameter

This parameter specifies the attribute being returned to Content Manager OnDemand as a user ID.

This parameter is required.

ARS_LDAP_PORT parameter

This parameter specifies the port on which LDAP is listening.

The default value is 389. This parameter is optional.

ARS_LDAP_SERVER parameter

This parameter specifies the IP address or the fully qualified host name of the LDAP server.

This parameter is required.

ARS_LDAP_SERVER_TYPE parameter

This parameter specifies the type of LDAP repository that is being configured.

Valid values are AD, SUN, or OPEN. Only one server can be configured per Content Manager OnDemand instance.

This parameter is required when using the ARSLSYNC program.

ARS_LDAP_USER_FILTER parameter

This parameter is used to query LDAP for users that will be imported into Content Manager OnDemand.

For example: (&(objectclass=user)(objectclass=CMODUSER))

This parameter is required when using the ARSLSYNC program.

LDAP SSL parameters

ARS_LDAP_USE_SSL parameter

This parameter specifies whether LDAP uses SSL.

To use SSL, set this parameter to TRUE. If this parameter is set to TRUE, then you must specify the following LDAP parameters:

- **ARS_LDAP_ALLOW_ANONYMOUS**
- **ARS_LDAP_KEYRING_FILE**
- **ARS_LDAP_KEYRING_LABEL**
- **ARS_LDAP_OD_AUTHORITY_FALLBACK**

ARS_LDAP_KEYRING_FILE parameter

Specifies the name of the SSL key database, SAF key ring, or PKCS#11 token.

For a SAF key ring, specify the name in the format *USERID/KEYRING_Name*. The current user ID is used if you do not specify the user ID. The user must have READ access to the IRR.DIGTCERT.LISTRING resource in the FACILITY class when using a SAF key ring that is owned by the current user. The user must have UPDATE access to the IRR.DIGTCERT.LISTRING resource in the FACILITY class when using a SAF key ring that is owned by another user.

Restriction: Certificate private keys are not available when you use a SAF key ring that is owned by another user.

For a PKCS#11 token, specify the name in the format **TOKEN*/NAME*, where *NAME* is the name of the PKCS #11 token. The user must have READ access to the SO.NAME and USER.NAME resources in the CRYPTOZ class when using a PKCS #11 token.

If you use an SSL key database, you must specify the GSK_KEYRING_STASH environment variable to specify a stash.

ARS_LDAP_KEYRING_LABEL parameter

This parameter specifies the LDAP keyring label.

ARS_LDAP_OD_AUTHORITY_FALLBACK parameter

Valid values are TRUE and FALSE.

If the value is set to TRUE and the user ID does not exist on the LDAP server for the bind attribute, the LDAP server authentication is bypassed. The logon process is reverted back to the normal Content Manager OnDemand logon.

This parameter is optional and defaults to FALSE if not specified.

Example:

```
ARS_LDAP_OD_AUTHORITY_FALLBACK=TRUE
```

Logging parameters

ARS_DISABLE_ARSLOG parameter

This parameter specifies that the ARSLOG exit is to be disabled completely.

Without specifying this parameter, even though all user exit logging is disabled in the **System Parameters** window of the administrative client, an attempt is still made to call the ARSLOG exit for certain messages. By specifying `ARS_DISABLE_ARSLOG=1` in `ars.cfg`, the ARSLOG exit is not invoked at all. If you do not plan on using the ARSLOG exit, specify `ARS_DISABLE_ARSLOG=1` to minimize the overhead of attempting to call the ARSLOG exit.

Migration parameters

ARS_MIGR_SERVERS parameter

Used with the migration server (coexistence feature). This parameter accepts arguments for the CICS server.

The **ARS_MIGR_SERVERS** parameter accepts the following arguments. Separate each argument with a blank character.

- The name of the CICS Region.
- The TCP/IP address of the CICS server.
- The communications protocol used by the CICS server. This argument must be set to 2 (two), meaning TCP/IP.
- The TCP/IP port number on which the CICS server is listening.
- Specifies whether CICS unified login is to be used. Specify `0` (zero), meaning CICS unified login support is not to be used.
- Specifies the version of the server. Specify 1 (one), meaning Content Manager OnDemand for OS/390® Version 2 server.

For example: `ARS_MIGR_SERVERS=OD210dis 9.91.121.31 2 3017 0 1`

Where:

OD210dis
Is the name of the CICS Region.

9.91.121.31
Is the TCP/IP address of the CICS server.

2
Means that the CICS server is using the TCP/IP communications protocol.

3017
Is the TCP/IP port number on which the server is listening.

0
Means that CICS unified login support is not being used.

1
Means that the server is Content Manager OnDemand for OS/390 Version 2.

ARS_MIGRATE_PLAN parameter

Specifies the DB2 plan to use when retrieving documents that are captured by Content Manager OnDemand for z/OS Version 2.1.

This plan is created by either the ARSZIAC2 or ARSCTBL1 members of SARSINST. If the value for this parameter is not specified, ARSZDOCG is used. This parameter needs to be specified only if Content Manager OnDemand for z/OS Version 2.1 documents are retrieved directly by the server and if the default plan is not acceptable. For example: `ARS_MIGRATE_PLAN=ARSBDOCG`

ARS_MIGRATE_SSID parameter

Specifies the DB2 subsytem to connect to when retrieving documents captured by Content Manager OnDemand for z/OS Version 2.1. If not specified, a default of DSN is used.

This parameter needs to be specified only if Content Manager OnDemand for z/OS Version 2.1 documents are retrieved directly by server, and the default DB2 subsystem is not acceptable. For example: `ARS_MIGRATE_SSID=DSNA`

ARS_V2_IAFC_OBJ_DEL parameter

This parameter specifies whether Content Manager OnDemand Version 2.1 and earlier versions, R/DARS and IAFC OAM objects get deleted due to an ARSADMIN UNLOAD process.

A value of 1 indicates that these objects get deleted. A value of 0 indicates the objects do not get deleted. This parameter defaults to 0.

ARS_V2_IAFC_OBJ_DEL_COMMIT parameter

This parameter specifies how often the Content Manager OnDemand Version 2.1 and earlier versions and IAFC OAM object deletion process issues SQL COMMITs.

Valid values range from 1 to 9999. This parameter defaults to 100.

OAM parameters

ARS_NUM_OAMSRVR parameter

Determines the number of task control blocks (TCBs) that the Content Manager OnDemand server starts to handle connections to OAM.

This parameter applies to all object servers. If the value specified is 0, no TCBs will be started and the server will not be able to access OAM objects. The parameter is limited to a maximum value of 30.

ARS_NUM_OAMSRVR_SLOW_RETRIEVE parameter

Determines the number of task control blocks (TCBs) that the Content Manager OnDemand server starts to handle connections to OAM for retrievals from objects with a slow retrieval time as defined by the **ARS_OAM_SLOW_RETRIEVE_THRESHOLD** parameter.

The **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** parameter applies to all object servers. If the value specified for this parameter is 0, no TCBs are dedicated for slow retrievals and all retrievals are processed by the TCBs that are associated with the **ARS_NUM_OAMSRVR** parameter. The default is 0. The **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** TCBs are in addition to the **ARS_NUM_OAMSRVR** TCBs and use additional DB2 connections.

ARS_OAM_DB2SSID parameter

Specifies the DB2 subsystem that is to be used for OAM access.

This parameter only needs to be specified if objects are to be stored using OAM.

ARS_OAM_PLAN parameter

Specifies the DB2 plan that is used for OAM access.

This parameter only needs to be specified if objects are to be stored using OAM.

ARS_OAM_SLOW_RETRIEVE_THRESHOLD parameter

Specifies the threshold at which OAM retrievals are processed by the TCBs that are associated with the **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** parameter.

If the estimated retrieval time for an object (as indicated by QELQERRT) is greater than or equal to the value of the **ARS_OAM_SLOW_RETRIEVE_THRESHOLD** parameter, the OSREQ RETRIEVE is processed by a **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** TCB. The default value is 12000. Refer to the Object Access Method Application Programmer's Reference manual for other valid QELQERRT values. An **ARS_OAM_SLOW_RETRIEVE_THRESHOLD** value of 0 along with a non-zero **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** value causes all OAM retrieve requests to be processed by the **ARS_NUM_OAMSRVR_SLOW_RETRIEVE** TCBs, while the **ARS_NUM_OAMSRVR** TCBs will process store, query, and delete requests.

Print parameters

ARS_PRINT_PATH parameter

Specifies the location where the Content Manager OnDemand server print function temporarily stores print data.

You must make sure that there is enough space in the specified location to hold the print files for the maximum number of concurrent print requests that the server will handle. The default value is `/tmp`. The **ARS_PRINT_PATH** parameter is ignored on object servers.

IBM recommends that you dedicate a file system on a separate mount point to hold the print files.

The permissions for the file system must be drwxrwxrwt. You can use the **CHMOD** command to set the permissions.

Note: By default, the installation process creates the `ARS.TMP.HFS` data set with a mount point at `/tmp` for temporary print data. See Part 2, "Installing the server software," on page 17 for more information.

ARSMVS_JESPRT_SUBMIT parameter

This parameter specifies that a job is submitted to invoke the direct server printer (JES:).

Normally, the server creates a program to send the print file to the JES SPOOL. If you specify `ARSMVS_JESPRT_SUBMIT=1`, a job is submitted to invoke that program. The job skeleton that is used is specified by ARSMVS_JESPRTJCL or by `/usr/lpp/ars/V10R5M0/config/arsprtjcl`.

ARSMVS_JESPRTJCL parameter

Specifies the full path name to a file. This path name contains the JCL used to invoke the direct server printer (JES: printers).

If the path is not specified, `/usr/lpp/ars/V10R5M0/config/arsprtjcl` is used.

ARSMVS_PRTJCL parameter

Specifies the full path name to a file. This path name contains the JCL used to invoke the InfoPrint server printer (arsprt script).

If the path is not specified, `/usr/lpp/ars/V10R5M0/config/arsprtjcl` is used.

ARSMVS_PRT_SPAWN parameter

Specifying ARSMVS_PRT_SPAWN=1 causes the arsprt script to be spawned.

Normally, when a server-based print is requested, the JCL in arsprtjcl is submitted to run the arsprt script. Since these jobs all have the same job name, the print requests are single threaded. If a large number of print requests are received, the UNIX System Services MAXPROCSYS limit can be reached.

ARSMVS_PRT_SPAWN_USERJOB parameter

This parameter specifies that when you spawn a process to perform server based printing (either direct server or InfoPrinter), the server attempts to set the job name based on the user who makes the print request.

Normally, the spawned address space inherits a job name based on the job name of the server. If ARSMVS_PRT_SPAWN_USERJOB=1 is specified, and the server has READ permission to the BPX.JOBNAME profile in the RACF FACILITY class, the spawned process has a job name that is based on the user who makes the print request. If the user name is less than 8 characters, the job name is the user name appended with a number from 0 to 9. The number is randomly picked.

Security parameters

ARSMVS_ALTERNATE_UNIFIED_LOGIN parameter

Enables protected user IDs to be authenticated.

If unified login is used, Passtickets provided by ARS.PTGN for user IDs protected with RACF cannot be used because protected user IDs cannot be authenticated with RACROUTE REQUEST=VERIFY.

To enable protected user IDs to run unified login, ARSMVS_ALTERNATE_UNIFIED_LOGIN=1 must be specified. If ARSMVS_ALTERNATE_UNIFIED_LOGIN=1 is specified, user IDs protected with RACF use the same method for authentication as the CICS client does for unified login. That is, the RACF UTOKEN is passed from the client program (for example, arsdoc) to the server. The server runs a RACROUTE REQUEST=TOKENMAP on the token. This allows the user IDs protected with RACF to run unified login.

ARSMVS_ARSUSEC_USERPROXY parameter

Enables the **ARSUSEC** parameter to check the ARSPROXY field of the CSDATA user segment.

If ARSMVS_ARSUSEC_USERPROXY=1, then the value as the Content Manager OnDemand user proxy is returned. The default is 0. See Chapter 50, “User security exit,” on page 179 for more information about user proxy processes.

ARSMVS_USE_ACEE_USERID parameter

Specifies that Content Manager OnDemand use an alternate method of determining the user ID when it runs commands and uses unified login.

When Content Manager OnDemand needs to determine a user ID, it extracts the UNIX System Services uid associated with the process, then determines the user ID associated with that uid. If the same uid is assigned to multiple users, Content Manager OnDemand cannot determine which user ID to choose or might choose the wrong user ID. Set the **ARSMVS_USE_ACEE_USERID** parameter to 1 (ARSMVS_USE_ACEE_USERID=1) to indicate that Content Manager OnDemand determine the user ID by the ACEE associated with the task or, if no task-level ACEE exists, by address space.

Server parameters

ARS_LOCAL_SRVR parameter

Identifies the object server.

Important: If you are configuring a library server, you must either omit this parameter from the ARS.CFG file or set this parameter to a blank value (that is, ARS_LOCAL_SRVR=).

If you are configuring an object server, set the **ARS_SRVR** parameter to the TCP/IP host name alias, fully qualified host name, or IP address of the object server.

Note: When you define a Content Manager OnDemand storage node for an object server, you must use the value of the **ARS_LOCAL_SRVR** parameter to name the storage node.

ARS_MESSAGE_OF_THE_DAY parameter

Use to show the message of the day. Set to the full path name of a file that contains the message that you want to show. For example: ARS_MESSAGE_OF_THE_DAY=/*dir*/tmp/message.txt

The contents of the message file can contain a maximum of 1024 characters of text. The administrative client and the Windows client show the message after the user logs on to the server. To close the message box and continue, the user must click **OK**. If you do not specify a message file, then the normal client processing occurs.

ARS_SRVR parameter

Identifies the library server.

Important: If you are configuring a library server, you must either omit this parameter from the ARS.CFG file or set this parameter to a blank value (that is, ARS_SRVR=).

If you are configuring an object server, then set the **ARS_SRVR** parameter to the TCP/IP host name alias, fully qualified host name, or IP address of the library server.

ARS_TMP parameter

The location where Content Manager OnDemand programs temporarily store data.

You must allocate sufficient free space in the specified file system to support tasks such as migrating and importing index data. The default value is: /tmp. You must specify the **ARS_TMP** parameter on the library server and on all object servers.

The permissions for the file system must be drwxrwxrwt. You can use the **CHMOD** command to set the permissions.

Note: By default, the installation process creates the ARS.TMP.HFS data set with a mount point at tmp for temporary data. See Part 2, “Installing the server software,” on page 17 for more information.

ARSMVS_UPPERCASE_WTO

This parameter determines whether WTO messages are changed to uppercase letters. If the value of this parameter is set to 1, WTO messages are changed to uppercase letters. Otherwise, the messages remain in mixed case format.

TZ environment variable

Specifies the mapping from the universal reference time to local time.

The **TZ** environment variable is typically specified in the ars.cfg file, and can also be specified as an ENVAR LE runtime option in the PARM= value in the ARSSOCKD STC.

The ars.cfg contains this **TZ** value: TZ=EST5EDT

The value of the **TZ** environment variable has the following fields:

standard
An alphabetic abbreviation for the local standard time zone. For example: GMT, EST, MSEZ.

***HH*[:*MM*[:*SS*]]**
The time offset westwards from the universal reference time. A leading minus sign (-) means that the local time zone is east of the universal reference time. An offset of this form must follow standard and can also optionally follow `daylight`. An optional colon (:) separates hours from optional minutes and seconds.

If `daylight` is specified without a `daylight` offset, daylight savings time is assumed to be one hour ahead of the standard time.

[*daylight*]
The abbreviation for your local daylight savings time zone. If the daylight field is missing, Daylight Saving Time conversion is disabled. The number of hours, minutes, and seconds your local Daylight Savings Time is offset from UTC when Daylight Savings Time is in effect. If the Daylight Savings Time abbreviation is specified, and the offset omitted, the offset of one hour is assumed.

[,*startdate*[/*starttime*],*enddate*[/*endtime*]]
A rule that identifies the start and end of Daylight Savings Time, specifying when Daylight Savings Time should be in effect. Both the *startdate* and *enddate* must be present, and must either take the form `J`*n*, *n*, or `M`*m*`.`*n*`.`*d*..

- `J`*n* is the Julian day *n* (1 <= *n* <=365) and does not account for leap days.
- *n* is the zero-based Julian day (0 <= *n* <= 365). Leap days are counted; therefore, you can refer to February 29th.
- `M`*m*`.`*w*`.`*d* defines the day (0 <= *d* <= 6) of week w (1 <= *w* <= 5) of month *m* (1 <= 12) of the year. Week 5 has the last day (*d*) in month *m*, which might occur in either the fourth or fifth week. Week 1 is the first week in which the *d*th day occurs. Day zero is Sunday.

Neither *starttime* nor *endtime* are required. If they are omitted, their values default to 02:00:00. If this Daylight Savings Time rule is omitted altogether, the values in the rule default to the standard American Daylight Savings Time rules starting at 02:00:00 the first Sunday in April and ending at 02:00:00 the last Sunday in October.

When the **TZ** environment variable is not set, time conversions behave as if TZ were set to GMT0 (`TZ=GMT0`).

TCP/IP parameters

ARSMVS_PORT_BIND_IPADDR parameter

Specifies the IP address to be used by the server when it must communicate with itself. This parameter should be used only if a TCPIP PORT BIND statement is used to restrict the IP address that ARSSOCKD can receive requests on.

```
ARSMVS_PORT_BIND_IPADDR=192.168.1.100
```

zEDC parameters

Available with PI41677, Content Manager OnDemand supports using the zEnterprise Data Compression (zEDC) available with z/OS 2.1 for OD77 and OD77Lite compression methods. The following ARS.CFG parameters allow control beyond what it provided by the zEDC zlib support.

ARS_USE_ZLIB_HW parameter

This parameter can be specified to prevent Content Manager OnDemand from using the zEDC zlib for compression.

For example, `ARS_USE_ZLIB_HW=0` specifies that Content Manager OnDemand should not use the zEDC zlib for OD77 and OD77Lite compression.

ARS_ZLIB_HW_COMPRESS_BUF parameter

Specifies that the zEDC zlib is only invoked when the input compression buffer exceeds the specified size in bytes. This should be at least as large as the size specified by the **DEFMINREQSIZE** parameter of the IQPPRMxx member of PARMLIB.

ARS_ZLIB_HW_DECOMPRESS_BUF parameter

Specifies that the zEDC zlib is only invoked when the input decompression buffer exceeds the specified size in bytes. This should be at least as large as the size specified by the **INFMINREQSIZE** parameter of the IQPPRMxx member of PARMLIB.

Chapter 14. Modify the ARS.CACHE file

The `ars.cache` file contains a list of the file systems on the server that will be used by Content Manager OnDemand for cache storage. The file systems reside in the HFS storage defined for use by UNIX System Services.

About this task

The following shows an example of an `ars.cache` file that specifies two cache storage file systems.

```
/ars1
/ars2
```

Note:

1. The first line in the `ars.cache` file identifies the base cache storage file system. Content Manager OnDemand maintains control information in the base cache storage file system. After you define the base cache storage file system to Content Manager OnDemand, you should not add or remove it from Content Manager OnDemand or change it in any way; otherwise, the system might fail.
2. You should modify the `ars.cache` file to specify information that is correct for your Content Manager OnDemand system. To modify the file, log on as the super user or as any user set up with appropriate permissions. Change to the `/usr/lpp/ars/V10R5M0/config` directory. Use a standard text editor to modify the file. Insert one line in the file for each file system on the server that Content Manager OnDemand can use for cache storage.
3. For a distributed library / object server system, configure one copy of the `ars.cache` file on each server that is part of the Content Manager OnDemand system. Also, if you plan to run more than one instance of Content Manager OnDemand on the same system, create one copy of the `ars.cache` file for each instance and specify separate cache storage file systems for each instance.

Chapter 15. Configure LDAP (optional)

You can optionally use LDAP (Light Weight Directory Access Protocol) as a method for controlling logon access to a Content Manager OnDemand server. After your LDAP server is operational and configured, you can configure Content Manager OnDemand to access the LDAP server.

About this task

To configure Content Manager OnDemand to access the LDAP server, modify the following files:

- `ARS.CFG`
- `ARSLDAP.INI`
- One or more user-configurable message text files

Configuring LDAP parameters in the ARS.CFG file

When you bind to an LDAP server, you can use anonymous binds or non-anonymous binds. With anonymous binds, you can gain access to the LDAP server without logging on. With non-anonymous binds, you must log on to the LDAP server to access the server. In both cases, you must provide a user ID and password after a connection is established with the LDAP server before you can access a particular record from the LDAP server.

About this task

The following table indicates the parameters in the `ARS.CFG` file that must be set depending on whether you use anonymous binds or non-anonymous binds.

Table 3. LDAP parameters in the ARS.CFG file that must be set for anonymous binds or non-anonymous binds

	Settings for anonymous binds	**Settings for non-anonymous binds**
ARS_LDAP_SERVER	Required	Required
ARS_LDAP_PORT	Optional	Optional
ARS_LDAP_BASE_DN	Required	Required
ARS_LDAP_BIND_ATTRIBUTE	Required	Required
ARS_LDAP_MAPPED_ATTRIBUTE	Required	Required
ARS_LDAP_ALLOW_ANONYMOUS	TRUE	FALSE
ARS_LDAP_BIND_MESSAGES_FILE	Optional	Optional

The **ARS_LDAP_BIND_MESSAGES_FILE** parameter identifies the bind message file, such as the `ARSLDAP.INI` file.

For detailed descriptions of the LDAP parameters, see “LDAP parameters” on page 42.

Configuring the ARSLDAP.INI file

Use the ARSLDAP.INI file to specify the message text that Content Manager OnDemand displays when status codes are returned from the LDAP server. You can use this file to enable Content Manager OnDemand to alert users when their LDAP password expired, or the LDAP account is locked.

About this task

The ARSLDAP.INI file contains three sections:

```
[BIND MESSAGES]
PASSWORD_EXPIRED="/usr/lpp/ars/V10R5M0/config/password_expired.txt"
ACCOUNT_LOCKED="/usr/lpp/ars/V10R5M0/config/account_locked.txt"

[PASSWORD_EXPIRED]
TDS6="Password has expired"
AD="data 532"
UDEF1=
UDEF2=
UDEF3=

[ACCOUNT_LOCKED]
TDS6="Account is locked"
AD="data 775"
UDEF1=
UDEF2=
UDEF3=
```

The BIND_MESSAGES section specifies the path to the files that contain the user-configurable message text that is displayed to the user when the LDAP password is about to expire or the LDAP account is locked. Generic files are provided and you must customize these files for your Content Manager OnDemand environment.

The following text is an example of a message that is displayed to a user:

```
Your LDAP password has expired and needs to be changed.
Log into http://www.example.com/password for instructions.
```

The PASSWORD_EXPIRED and ACCOUNT_LOCKED sections of the ARSLDAP.INI file enable the LDAP server to determine when a user's password has expired or the account is locked, respectively. The entries in these sections are for Tivoli Directory Server Version 6 or later (TDS6), and Microsoft Active Directory (AD). These sections also contain three user-defined entries (UDEF*n*) that you can use to enter your own pattern strings for LDAP servers that are not directly supported.

The LDAP server might return additional information when the user's bind operation fails. When the LDAP server returns an error, Content Manager OnDemand searches for user-defined text that matches the LDAP server error code in the [PASSWORD_EXPIRED] and [ACCOUNT_LOCKED] sections of the message file that is referenced by the **ARS_LDAP_BIND_MESSAGES_FILE** parameter of the ARSLDAP.INI file. If matching text is found, Content Manager OnDemand displays the user-defined text. If no matching text is found, such as if the **ARS_LDAP_BIND_MESSAGES_FILE** parameter is not defined or the PASSWORD_EXPIRED or ACCOUNT_LOCKED files do not exist, Content Manager OnDemand returns the message `The server failed while attempting to log on.`

Currently only two error conditions can be handled: PASSWORD_EXPIRED and ACCOUNT_LOCKED. The section titles for these two conditions cannot be changed, but you can change the pattern strings and message text that is presented to the user to define any two error conditions.

Creating user-configurable message text files

You must manually create `.txt` files that contain the message text and store the files in the appropriate directory.

About this task

For example, if you specify `PASSWORD_EXPIRED="/usr/lpp/ars/V10R5M0/config/password_expired.txt"` in the `ARSLDAP.INI` file, you must create a text file with the name `password_expired.txt` in the `/usr/lpp/ars/V10R5M0/config` directory. In the file, specify one or more lines of message text to display. For example, specify the text `Your LDAP password expired and needs to be changed`.

Chapter 16. Verify the CLI.INI file

About this task

Note: The `CLI.INI` file contains the configuration information for ODBC that ARSSOCKD uses to connect to the correct DB2 subsystem. This file is pointed to by the value of DSNAOINI. This value could be already set within your installation by a DSANOINI DD statement in JCL or as a HFS file as in the example. See *DB2® V10 for z/OS ODBC Guide and Reference* for more information about configuring the `CLI.INI` file.

The following shows an example of the `CLI.INI` file with the minimum keywords required; however, other keywords might be present.

```
[COMMON]
MVSDEFAULTSSID=DSNA
[DSNA]
PLANNAME=DSNACLI
```

Where:

MVSDEFAULTSSID=DSNA
Identifies to ODBC the DB2 subsystem name or group attachment name in a data sharing group.

PLANNAME=DSNACLI
If the plan for ODBC is not the default plan DSNACLI, you must specify the plan with PLANNAME=*plan*

Verify that the ARSSOCKD owner has execute privilege on the ODBC plan (for example, DSNACLI). If the ID does not have access, the customer needs to grant the access. One way to do that is

```
GRANT EXECUTE ON PLAN planname TO arssockd owner
```

Note: Content Manager OnDemand does not support the RRSAF value of the MVSATTACHTYPE keyword in the `CLI.INI` file. If the MVSATTACHTYPE keyword is specified, the value must be CAF (that is, `MVSATTACHTYPE=CAF`).

Chapter 17. Modify the ARSSOCKD procedure

About this task

The ARSSOCKD procedure is used to start the server as a started task.

A sample ARSSOCKD procedure is provided in the SARSINST library. You must customize the procedure for your environment and copy the customized procedure into the PROCLIB concatenation.

The following is the sample ARSSOCKD procedure.

```
//ARSSOCKD PROC
//ARSSOCKD EXEC PGM=ARSSOCKD,REGION=0M,TIME=NOLIMIT
//STEPLIB  DD DISP=SHR,DSN=ARS.V10R5M0.SARSLOAD
//         DD DISP=SHR,DSN=DB2.V10R5M0.SDSNEXIT
//         DD DISP=SHR,DSN=DB2.V10R5M0.SDSNLOAD
//*DSNAOINI DD PATH='/usr/lpp/ars/V10R5M0/config/cli.ini'
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
```

Figure 5. Sample ARSSOCKD procedure.

The following list explains the statements highlighted in Figure 5 on page 61:

//STEPLIB DD
Specifies the program libraries, which must be customized for the local naming conventions.

//*DSNAOINI DD
Specifies the location and name of the ODBC parameter file. This DD statement must be un-commented and customized if you need to configure ODBC. For more information about configuring ODBC, see the *DB2 V10 for z/OS ODBC Guide and Reference*.

Note: Environment variables may be specified by using the ENVAR Language Environment runtime option on the PARM= JCL keyword. See *z/OS Language Environment Programming Reference* for more information. For example, to set the locale (and the associated code page) for the server, you could specify: `PARM='ENVAR("LC_ALL=De_DE.IBM-273")/'`

If you wish to specify more environment variables than will fit in the 100-character PARM= keyword, additional environment variables can be set from an MVS™ data set or HFS file by using the **_CEE_ENVFILE** environment variable. See *z/OS XL C/C++ Programming Guide* for details about the **_CEE_ENVFILE** environment variable. Note that for an MVS data set, the data set must be allocated with RECFM=V or RECFM=VB.

Chapter 18. Modify the ARSLOAD procedure

About this task

The ARSLOAD procedure starts the load daemon (long running task) that captures SYSOUT directly from spool and loads it into Content Manager OnDemand. See"Automating the data loading process" on page 155 for details and other requirements.

A sample ARSLOAD procedure is provided in the SARSINST library. You must customize the procedure for your environment and copy the customized procedure into the PROCLIB concatenation. The STEPLIB DDs should be customized to your local naming conventions.

The following is the sample ARSLOAD procedure.

```
//ARSLOAD  PROC
//ARSLOAD  EXEC PGM=ARSLOAD,REGION=0M,TIME=NOLIMIT,
//  PARM=('/-h ARCHIVE -C Q')
//STEPLIB  DD DISP=SHR,DSN=ARS.V10R5M0.SARSLOAD
//         DD DISP=SHR,DSN=DB2.V10R5M0.SDSNEXIT
//         DD DISP=SHR,DSN=DB2.V10R5M0.SDSNLOAD
//         DD DISP=SHR,DSN=ACIF.V4R5M0.SAPKMOD1
//*******************************************
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=121,BLKSIZE=6050
//SYSOUT   DD SYSOUT=*
```

Figure 6. Sample ARSLOAD procedure.

The following list explains the highlighted JCL statements:

//ARSLOAD EXEC PGM=ARSLOAD ... PARM=('/-h ARCHIVE -C Q')

Specifies the parameters for the ARSLOAD program. The parameters are documented in the ARSLOAD section of the *IBM Content Manager OnDemand for z/OS: Administration Guide*. The example specifies the following parameters:

-h ARCHIVE

Specifies the name of the Content Manager OnDemand instance to process. The system locates the specified instance name in the ARS.INI file to determine the IP address, host name alias, or fully-qualified host name of the library server. The example **-h** parameter specifies an instance name of ARCHIVE.

Note: In the JCL, the / character to the left of the **-h** parameter ensures that everything to the right will be treated as program parameters and not as Language Environment® runtime options.

-C Q

Specifies the JOB CLASS to monitor for SYSOUT to load into Content Manager OnDemand. The example **-C** parameter specifies that CLASS Q be monitored.

You can also specify the following parameters:

-F *form*

Specify the FORM to monitor.

-W *external writer*

Specify the external writer name to be used.

Note: A Content Manager OnDemand administrative user ID and password must be specified to the ARSLOAD procedure, or the ARS.PTGN exit might be used to allow ARSLOAD to run without needing to specify a user ID and password. You can specify the **-u** *userid* and **-p** *password* parameters to the PARM keyword on the EXEC statement or you can use the ARS.PTGN exit. See Chapter 51, "Unified login (ARS.PTGN) exit," on page 201 for details and requirements about the ARS.PTGN exit. In Figure 6 on page 63, because the **-u** and **-p** parameters are omitted from the PARM keyword on the EXEC statement, the ARS.PTGN exit must be used; otherwise, the ARSLOAD procedure will fail.

//STEPLIB DD ... DSN=ACIF.V4R5M0.SAPKMOD1
Specifies the ACIF library, which must be customized for the local naming conventions.

Part 4. Creating and initializing the database

Before you begin to define reports to Content Manager OnDemand, load data on the system, or use the system, you must create the Content Manager OnDemand database and initialize the system tables. The database resides on the Content Manager OnDemand library server.

This part describes the tasks required to create the storage group, create the Content Manager OnDemand database, create a table space for the Content Manager OnDemand system tables, create the Content Manager OnDemand system tables, initialize the Content Manager OnDemand system log component, initialize the system load logging facility, and initialize the Content Manager OnDemand system migration component.

Chapter 19. Creating the storage group and database

Procedure

To run the ARSDB2 job to create the storage group and the Content Manager OnDemand database, do the following steps:

1. Modify the JCL for your environment and replace the symbolic fields.

   For more information, see Chapter 19, “Creating the storage group and database,” on page 67 . You can find the ARSDB2 member in the SARSINST library.
2. Submit the job.
3. Check job completion status in the SDSF output.

Example

The following shows an example of the ARSDB2 job.

```
//ARSDB2   JOB (account),userid,MSGCLASS=A,REGION=0M
//*  FUNCTION: CREATE THE DB2 DATA BASE AND STORAGE GROUP
//*            TO INSTALL AND RUN OD/MP
//*  NOTES:    ON MULTIPLE CPU SYSTEMS, THIS JOB MUST BE RUN
//*            ON THE SYSTEM WHERE DB2 EXECUTES.
//*************************************************************
//*
//* TO CREATE THE OD/MP DB2 DATABASE AND STORAGE GROUP:
//* 1. PROVIDE A SUITABLE JOBCARD FOR YOUR ENVIRONMENT.
//* 2. REPLACE THE FOLLOWING SYMBOLIC FIELDS:
//*    db2ssid     - DB2 SUBSYSTEM ID
//*    catalog     - EXISTING CATALOG NAME OR CATALOG ALIAS
//* 3. (OPTIONAL) CHANGE THE FOLLOWING CREATE PARAMETERS:
//*    STOGROUP    - DEFAULT STORAGE GROUP IS ARSSGRP
//*    DATABASE    - DEFAULT DATABASE IS ARSDBASE
//*    PLAN        - DEFAULT DB2 PLAN NAME IS DSNTIA91
//*    LIB         - DEFAULT DB2 RUNTIME LIBRARY IS
//*                  'DB2.RUNTIME.LOADLIB'
//*    SQLID       - DEFAULT CURRENT SQLID='ARSUSER'
//*
//* example entries
//* account                    =  acct
//* userid                     =  USER
//* db2ssid                    =  DSNA
//* db2.runtime.loadlib        =  DB2.V10R5M0.DSNA.RUNLIB.LOAD
//* arsuser                    =  ARSSERVR
//* catalog                    =  DSNACAT
//*
//* 4. SUBMIT THE JOB.
//*
//************************ Sample 1 ************************************
//STEP1    EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DISP=SHR,DSN=DB2.V10R5M0.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(db2ssid)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA10) -
      LIB('DB2.RUNTIME.LOADLIB')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
 SET CURRENT SQLID='arsuser';
 CREATE STOGROUP ARSSGRP
        VOLUMES ('*','*')
        VCAT  catalog;
 CREATE DATABASE ARSDBASE           .
        STOGROUP ARSSGRP;
```

Figure 7. Sample ARSDB2 job.

The JCL statements in Figure 7 on page 67 are explained as follows.

//ARSDB2 JOB (*account*),*userid*
Provide a suitable JOB statement for your environment.

//STEPLIB DD
Specifies the DB2 load library, which must be customized for the version and local naming conventions.

//SYSTSIN DD
Specifies the parameters for the ARSDB2 job. Verify and replace the following parameter values:

DSN SYSTEM(*db2ssid*)
Specify the DB2 subsystem ID.

PLAN(DSNTIA10)
Specify the PLAN.

LIB ('DB2.RUNTIME. LOADLIB')
Specify the DB2 runtime library.

//SYSIN DD
Specifies the parameters for the ARSDB2 job. Verify and replace the following parameter values:

SQLID='*arsuser*'
The SQLID that is specified for the database should be the same as that specified for the **SRVR_INSTANCE_OWNER** parameter in the `/usr/lpp/ars/V10R5M0/config/ars.ini` file. The default value in the `ars.ini` file is ARSSERVR (see Chapter 12, “Verify the ARS.INI file,” on page 35).

STOGROUP ARSSGRP
Specify the name for the storage group.

DATABASE ARSDBASE
Specify the name for the Content Manager OnDemand database.

Important: The database name that you specify must match the name that you specified for the **SRVR_INSTANCE** parameter in the `/usr/lpp/ars/V10R5M0/config/ars.ini` file. The default value in the `ars.ini` file is ARSDBASE (see Chapter 12, “Verify the ARS.INI file,” on page 35).

Chapter 20. Creating the table spaces

About this task

After you create the storage group and the database by running the ARSDB2 job, run the ARSTSPAC job to create table spaces that are used for the Content Manager OnDemand system tables.

Procedure

To create the table spaces, do the following steps:

1. Modify the JCL for your environment and replace the symbolic fields.

   For more information, see Chapter 20, “Creating the table spaces,” on page 69. The ARSTSPAC member can be found in the ODADMIN.V10R5M0.SARSINST library.
2. Submit the job.
3. Check job completion status in the SDSF output.

Example

Figure 8 on page 69 shows an example of the ARSTSPAC job.3

Figure 8. Sample ARSTSPAC job.

```
//ARSTBSPC JOB (account),userid,MSGCLASS=A
//*************************************************************
//*
//*  FUNCTION: CREATE THE DB2 TABLESPACES REQUIRED TO INSTALL
//*            AND RUN ONDEMAND. THE FOLLOWING IS A LIST OF THE
//*            TABLESPACES THAT WILL BE CREATED.
//*
//*  FMID:     H272950
//*
//*  NOTES:    ON MULTIPLE CPU SYSTEMS, THIS JOB MUST BE RUN
//*            ON THE SYSTEM WHERE DB2 EXECUTES.
//*
//*************************************************************
//*
//* TO CREATE THE ONDEMAND TABLESPACE:
//*
//* 1. PROVIDE A SUITABLE JOBCARD FOR YOUR ENVIRONMENT.
//*
//* 2. REPLACE THE FOLLOWING SYMBOLIC FIELDS:
//*    db2ssid  - DB2 SUBSYSTEM NAME
//*    LIB      - DB2 RUNTIME LOAD LIBRARY
//*    PLAN     - DEFAULT DB2 PLAN NAME IS DSNTIA10
//*    ARSUSER  - DEFAULT SQLID IS ARSUSER
//*
//* 3. (OPTIONAL) CHANGE THE FOLLOWING CREATE PARAMETERS:
//*    DATABASE   - DEFAULT DATABASE IS ARSDBASE
//*    STOGROUP   - DEFAULT STORAGE GROUP IS ARSSGRP
//*    PRIQTY     - PRIMARY TABLESPACE ALLOCATION
//*    SECQTY     - SECONDARY TABLESPACE ALLOCATION
//*    BP32K      - 32K BUFFERPOOL NAME
//*    BP0        - 4K BUFFERPOOL NAME
//*    STEPLIB    - SPECIFIES THE SDSNLOAD DATA SET
//*
//* 4. SUBMIT THE JOB.
//*
//*************************************************************
//* STEP1: CREATE THE ONDEMAND TABLESPACES
//*************************************************************
//STEP1    EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=?db2qual?.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(db2ssid)
 RUN  PROGRAM(DSNTIAD) PLAN(?DB2PLAN?) -
      LIB('?db2run?')
//SYSPRINT DD SYSOUT=*
```

```
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
 SET CURRENT SQLID='ARSUSER';
 CREATE TABLESPACE ARSAGT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 220
      SECQTY 110
      SEGSIZE 64
      LOCKSIZE ROW
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSAGFLT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 292
      SECQTY 146
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSAGFAT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 184
      SECQTY 92
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSAG2FT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 100
      SECQTY 50
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSAGPET
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 320
      SECQTY 120
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSAGIDT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 320
      SECQTY 120
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSANNT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 2656
      SECQTY 1328
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSAPPT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 3648
      SECQTY 1824
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSAPPUT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 288
      SECQTY 144
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSCABT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 122
      SECQTY 61
      SEGSIZE 64
      LOCKSIZE ANY
```

```
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSCABFT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 100
     SECQTY 50
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSCABNT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 210
     SECQTY 55
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSCABPT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 160
     SECQTY 80
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSCFSWT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 160
     SECQTY 80
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSCMFWT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 210
     SECQTY 55
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSFOLT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 122
     SECQTY 61
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSFOLFT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 200
     SECQTY 100
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSFOLNT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 210
     SECQTY 55
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSFOLUT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 100
     SECQTY 50
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSFOLPT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 40
     SECQTY 20
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSFTIWT
```

```
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 210
     SECQTY 55
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSGROUT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 16
     SECQTY 8
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSHLDT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 16
     SECQTY 8
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSHLDMT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 16
     SECQTY 8
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSHLDNT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 210
     SECQTY 55
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSHLDPT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 16
     SECQTY 8
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSHLDWT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 16
     SECQTY 8
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSLOADT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 348
     SECQTY 174
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSLDWKT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 16
     SECQTY 8
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSNAMET
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 224
     SECQTY 112
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSNODET
     IN ARSDBASE
     USING STOGROUP ARSSGRP
```

```
     PRIQTY 12
     SECQTY 12
     SEGSIZE 4
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSPRTT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 20
     SECQTY 10
     SEGSIZE 4
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSPRTOT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 12
     SECQTY 12
     SEGSIZE 4
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSPRUST
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 24
     SECQTY 12
     SEGSIZE 4
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSREST
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 188
     SECQTY 94
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSSEGT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 172
     SECQTY 86
     SEGSIZE 64
     LOCKSIZE ROW
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSSETT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 12
     SECQTY 12
     SEGSIZE 4
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSSYST
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 96
     SECQTY 96
     SEGSIZE 8
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSUSERT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 344
     SECQTY 172
     SEGSIZE 64
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSUSRGT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 24
     SECQTY 12
     SEGSIZE 4
     LOCKSIZE ANY
     BUFFERPOOL BP0;
CREATE TABLESPACE ARSUSGIT
     IN ARSDBASE
     USING STOGROUP ARSSGRP
     PRIQTY 24
     SECQTY 12
```

```
      SEGSIZE 4
      LOCKSIZE ANY
      BUFFERPOOL BP0;
 CREATE TABLESPACE ARSSLUT
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 16
      SECQTY 8
      SEGSIZE 64
      LOCKSIZE ANY
      BUFFERPOOL BP32K;
```

The following list describes the JCL statements in Figure 8 on page 69:

//ARSTSPAC JOB (*account*),*userid*
Provide a suitable JOB statement for your environment.

//STEPLIB DD
Specifies the DB2 load library, which must be customized for the version and local naming conventions.

//SYSTSIN DD
Specifies the parameters for the ARSTSPAC job. Verify the following parameter values:

DSN SYSTEM(*db2ssid*)
Specify the DB2 subsystem ID.

PLAN(DSNTIA91)
Specify the PLAN.

LIB ('DB2.RUNTIME. LOADLIB')
Specify the DB2 runtime library.

//SYSIN DD
Specifies the parameters for the ARSTSPAC job. Verify the following parameter values:

SQLID='*arsuser*'
Specify the SQLID for the database. The value for *arsuser* must be the same as the value specified for the **SRVR_INSTANCE_OWNER** parameter in the `/usr/lpp/ars/V10R5M0/config/ars.ini` file. The default value in the `ars.ini` file is ARSSERVR. For information about specifying values in the `ars.ini` files, see Chapter 12, “Verify the ARS.INI file,” on page 35.

TABLESPACE *name_of_table_space*
Specify the name of the table space where the ARSTPAC job stores the Content Manager OnDemand system tables it creates. If you want to use the default names, use the values specified in the ARSTPAC member. If you want to create custom names, specify the custom names for each table space in *name_of_table_space* and see “Building tables and indexes into custom table spaces” on page 76.

IN ARSDBASE
Specify the name of the Content Manager OnDemand database.

Note:

1. The database name must match the value that you specified for the **SRVR_INSTANCE** parameter in the `/usr/lpp/ars/V10R5M0/config/ars.ini` file. The default value in the `ars.ini` file is ARSDBASE. For information about specifying values in the `ars.ini` files, see Chapter 12, “Verify the ARS.INI file,” on page 35.
2. The database name must match the value that you specified when you created the database in Chapter 19, “Creating the storage group and database,” on page 67.

STOGROUP ARSSGRP
Specify the name for the storage group. This value must match the value that you specified when you created the storage group in Chapter 19, “Creating the storage group and database,” on page 67).

PRIQTY
Specify the primary table space allocation.

SECQTY
Specify the secondary table space allocation.

BUFFERPOOL
Specify the name of the bufferpool.

What to do next

Existing installations that are using a single tablespace are not required to migrate to multiple table spaces. However, if you choose to, you can migrate using the following steps:

1. Shutdown the ARSSOCKD server.
2. Backup the ARSDBASE database.
3. Run `/usr/lpp/ars/V10R5M0/bin/arsdb` to export the Content Manager OnDemand system tables.
4. DROP TABLESPACE ARSTSPAC.
5. Run the ARSTSPAC job.
6. Run `/usr/lpp/ars/V10R5M0/bin/arsdb -trs` to create the tables in the new table spaces.
7. Run `arsdb -i` to import the Content Manager OnDemand system tables.
8. Restart the ARSSOCKD server.

Estimating storage values for ARSTSPAC member

About this task

The PRIQTY and SECQTY values specified in the sample ARSTSPAC member are an example. In the example, it was assumed that there would be a total of 1000 application groups and 1000 applications, with 4 index fields per application. It was also assumed that user IDs would be at most 8 characters. To calculate the values for your site, review "System Control Tables" of the *Administration Guide* and include the following factors:

- Size of user IDs and group names. Content Manager OnDemand allows these to be up to 120 characters. You might choose to create user IDs with a smaller length, particularly if you are going to have Content Manager OnDemand interface with RACF or another external security manager.
- Sizes of names and descriptions for various entities such as application groups and folders. Content Manager OnDemand allows 60 character names and 120 character descriptions.
- Size of the MIGR_SRVR_STR column in the ARSAG table. Content Manager OnDemand allows 254 characters. You might not be using the V2 migration capability, or you might be using significantly shorter migration server names.
- Size of the INTERNAL and EXTERNAL columns of the ARSAGFLDALIAS table. Content Manager OnDemand allows 254 bytes for each. You might decide to limit the names to 16 bytes, for example.
- Number of fields per application group (ARSAGFLD). There is a maximum of 128 fields per application group, but your site might average fewer.
- Number of application group fields per folder (ARSAG2FOL).
- Number of distinct permissions per application group, and the size of the Q_RESTRICT column for the ARSAGPERMS table. You can expect at least 2 rows per application group. The Q_RESTRICT column provides up to 32K for a query restriction SQL string. However, your site might choose to limit the size of the string or not use this capability.
- Number of annotations. There will be one row in ARSANN for each annotation. Content Manager OnDemand allows up to 32K bytes of text data for each annotation. You might choose to not use annotations, or expect annotations to be, on average of smaller length.
- Are you going to post-process the application indexes before loading them. If not, the size of the PREPROCESSOR column of ARSAPP will be zero.

- The size of the indexer parameters in the INDEXER column of ARSAPP. This size is dependant on the size of the parameters that will get passed to the indexer during load processing.
- The size of the application identifier within the application (ALIAS column of ARSAPP). Content Manager OnDemand allows for 254 characters. You might restrict yourself to smaller identifier sizes.
- The size of the FIXED_VIEW column of ARSAPP. It would be unlikely to exceed 1000 bytes.
- Are you going to exploit giving individual users and/or groups logical views of the application (ARSAPPUSR)?
- The number of folders you expect to create (ARSFOL). There will be one row per folder. Are you going to create folder fields for individual users/groups (ARSFOLFLDUSR). There will be at least 1 per folder field.
- Number of users and groups with distinct permissions to a folder (ARSFOLPERMS).
- Number of groups (ARSGROUP). There will be one row per group.
- Number of documents loaded with expiration type LOAD (ARSLOAD).
- Number or named queries (ARSNAMEQ) and the size of the query (BUF column of ARSNAMEQ).
- Number of storage nodes. There will be one row in ARSNODE per storage node.
- Number of defined server based printers (ARSPRT).
- Number of individual permissions assigned to a printer (ARSPRTUSR).
- Total number of AFP resources expected to be loaded (ARSRES). There will be 1 row per loaded resource.
- Total number of application group data tables you expect Content Manager OnDemand to have active at any given time (ARSSEG).
- Total number of storage sets (ARSSET). There will be one row per storage set.
- The number of users you expect to define to Content Manager OnDemand (ARSUSER). There will be one row per user defined to Content Manager OnDemand.

Building tables and indexes into custom table spaces

About this task

When you create the Content Manager OnDemand tables or indexes, the **ARSDB** command can build the tables and indexes in the default table space or in table spaces that you create. When you run the **ARSDB** command, Content Manager OnDemand validates the existence of the table space you specified. If the table space does not exist, the **ARSDB** command creates the Content Manager OnDemand system tables into the default table space.

Procedure

If you want the **ARSDB** command to build the Content Manager OnDemand system tables and indexes into tables spaces that you create (user-defined table spaces), do the following tasks before you run the **ARSDB** command:

1. Create the table spaces by following the directions in Chapter 20, “Creating the table spaces,” on page 69. If you choose the default table space names, which are listed in “Default names for system tables and table spaces” on page 78, you can skip the next step.
2. If you want to create your own table space names, modify the Content Manager OnDemand configuration file. Specify the names of the tables spaces that you created by adding the parameters listed in “Parameters to specify names for table spaces containing system tables” on page 76.

Parameters to specify names for table spaces containing system tables

The following list describes the parameters you can use to specify a name for each Content Manager OnDemand system table space:

- **ARS_ARSAG_TABLESPACE**

- **ARS_ARSAG2FOL_TABLESPACE**
- **ARS_ARSAGFLD_TABLESPACE**
- **ARS_ARSAGFLDALIAS_TABLESPACE**
- **ARS_ARSAGINDEX_TABLESPACE**
- **ARS_ARSAGPERMS_TABLESPACE**
- **ARS_ARSANN_TABLESPACE**
- **ARS_ARSAPP_TABLESPACE**
- **ARS_ARSAPPUSR_TABLESPACE**
- **ARS_ARSCAB_TABLESPACE**
- **ARS_ARSCAB2FOL_TABLESPACE**
- **ARS_ARSCABPERMS_TABLESPACE**
- **ARS_ARSCFSODWORK_TABLESPACE**
- **ARS_ARSCMFEDWORK_TABLESPACE**
- **ARS_ARSFOL_TABLESPACE**
- **ARS_ARSFOLFLD_TABLESPACE**
- **ARS_ARSFOLFLDUSR_TABLESPACE**
- **ARS_ARSFOLPERMS_TABLESPACE**
- **ARS_ARSFTIWORK_TABLESPACE**
- **ARS_ARSGROUP_TABLESPACE**
- **ARS_ARSHOLD_TABLESPACE**
- **ARS_ARSHOLDMAP_TABLESPACE**
- **ARS_ARSHOLDPERMS_TABLESPACE**
- **ARS_ARSHOLDWORK_TABLESPACE**
- **ARS_ARSLOAD_TABLESPACE**
- **ARS_ARSNAMEQ_TABLESPACE**
- **ARS_ARSNODE_TABLESPACE**
- **ARS_ARSPRT_TABLESPACE**
- **ARS_ARSPRTOPTS_TABLESPACE**
- **ARS_ARSPRTUSR_TABLESPACE**
- **ARS_ARSRES_TABLESPACE**
- **ARS_ARSSEG_TABLESPACE**
- **ARS_ARSSET_TABLESPACE**
- **ARS_ARSSYS_TABLESPACE**
- **ARS_ARSUSER_TABLESPACE**
- **ARS_ARSUSRGRP_TABLESPACE**
- **ARS_ARSUSRGRPID_TABLESPACE**
- **ARS_ARSCABNT_TABLESPACE**
- **ARS_ARSCABNAMES_TABLESPACE**
- **ARS_ARSFOLNT_TABLESPACE**
- **ARS_ARSFOLNAMES_TABLESPACE**
- **ARS_ARSHOLNT_TABLESPACE**
- **ARS_ARSHOLDNAMES_TABLESPACE**
- **ARS_ARSLDWKT_TABLESPACE**

- **ARS_ARSLOADWORK_TABLESPACE**

The following list describes the parameters you can use to specify a name for each Content Manager OnDemand Distribution Facility table space:

- **ARS_ARSDFBDT_TABLESPACE**
- **ARS_ARSDFCRT_TABLESPACE**
- **ARS_ARSDFDCT_TABLESPACE**
- **ARS_ARSDFDRT_TABLESPACE**
- **ARS_ARSDFDST_TABLESPACE**
- **ARS_ARSDFEML_TABLESPACE**
- **ARS_ARSDFLIS_TABLESPACE**
- **ARS_ARSDFPPT_TABLESPACE**
- **ARS_ARSDFUOT_TABLESPACE**

Default names for system tables and table spaces

Table 4. Default names for Content Manager OnDemand system tables and table spaces

Table name	Table space names
ARSAG	ARSAGT
ARSAG2FOL	ARSAG2FT
ARSAGFLD	ARSAGFLT
ARSAGFLDALIAS	ARSAGFAT
ARSAGINDEX	ARSAGIDT
ARSAGPERMS	ARSAGPET
ARSANN	ARSANNT
ARSAPP	ARSAPPT
ARSAPPUSR	ARSAPPUT
ARSCAB	ARSCABT
ARSCAB2FOL	ARSCABFT
ARSCABPERMS	ARSCABPT
ARSCFSODWORK	ARSCFSWT
ARSFOL	ARSFOLT
ARSFOLFLD	ARSFOLFT
ARSFOLFLDUSR	ARSFOLUT
ARSFOLPERMS	ARSFOLPT
ARSFTIWORK	ARSFTIWT
ARSGROUP	ARSGROUT
ARSHOLD	ARSHLDT
ARSHOLDMAP	ARSHLDMT
ARSHOLDPERMS	ARSHLDPT
ARSHOLDWORK	ARSHLDWT

Table 4. Default names for Content Manager OnDemand system tables and table spaces (continued)	
Table name	**Table space names**
ARSLOAD	ARSLOADT
ARSNAMEQ	ARSNAMET
ARSNODE	ARSNODET
ARSPRT	ARSPRTT
ARSPRTOPTS	ARSPRTOT
ARSPRTUSR	ARSPRUST
ARSRES	ARSREST
ARSSEG	ARSSEGT
ARSSET	ARSSETT
ARSSYS	ARSSYST
ARSUSER	ARSUSERT
ARSUSRGRP	ARSUSRGT
ARSUSRGRPID	ARSUSGIT

Table 5. Default names for ODF system tables and table spaces	
Table name	**Table space names**
ARSDFBDT	ARSDFBDT
ARSDFCRT	ARSDFCRT
ARSDFDCT	ARSDFDCT
ARSDFDRT	ARSDFDRT
ARSDFDST	ARSDFDST
ARSDFEML	ARSDFEML
ARSDFLIS	ARSDFLIS
ARSDFPPT	ARSDFPPT
ARSDFUOT	ARSDFUOT

Chapter 21. Creating the Content Manager OnDemand system tables

About this task

After you have successfully created a table space for the Content Manager OnDemand system tables, run the ARSDB program to create the Content Manager OnDemand system tables.

Procedure

To run the ARSDB program, do the following steps:

1. Set up the ODBC environment:
   a) Log on to OMVS.
   b) Switch to the super user (su)
   c) Export the environment variable.
   For example: `export DSNAOINI="/etc/ars/cli.ini"`
   d) Run the **set** command to display the environment settings.
2. Move to the Content Manager OnDemand executable directory.
   For example: `cd /usr/lpp/ars/V10R5M0/bin`
3. Run the ARSDB program.
   For example: `arsdb -trs`
   The ARSDB program will generate the following series of messages:

```
Creating the DB2 ARSDBASE database
Creating table ARSSERVR.arsag
Creating index ARSSERVR.arsag_name_idx
Creating index ARSSERVR.arsag_agid_idx
..... .. . .  . . . . . . . . . . . . . . . . . .
..... .. . .  . . . . . . . . . . . . . . . . . .
..... .. . .  . . . . . . . . . . . . . . . . . .
Updating runstat statistics for table ARSSERVR.arsusrgrp
Creating table ARSSERVR.arsusrgrpid
Creating index ARSSERVR.arsusrgrpid_idx
Updating runstat statistics for table ARSSERVR.arsusrgrpid
```

Chapter 22. Initializing the system log

About this task

After you have successfully created the Content Manager OnDemand system tables, run the ARSSYSCR program to initialize the Content Manager OnDemand system log component.

Procedure

To run the ARSSYSCR program, do the following steps:

1. Move to the Content Manager OnDemand executable directory.
   For example:`cd /usr/lpp/ars/V10R5M0/bin`
2. Run the ARSSYSCR program.
   For example: `./arssyscr -l`.

   If you change the default instance from ARCHIVE to another value, specify the instance on the **ARSSYSCR** command. Otherwise, you might receive the following error: `arssyscr unable to determine database engine`.

   The ARSSYSCR program will generate a series of messages similar to the following:

```
arssyscr:       Updating ARSSERVR.ARSSYS
arssyscr:       Adding to ARSSERVR.ARSAG with Storage Set Id = 0
arssyscr:       Adding to ARSSERVR.ARSAGPERMS
arssyscr:       Adding to ARSSERVR.ARSAGFLD
arssyscr:       Adding to ARSSERVR.ARSAGFLDALIAS
arssyscr:       Adding to ARSSERVR.ARSAG2FOL
arssyscr:       Adding to ARSSERVR.ARSAPPUSR
arssyscr:       Adding to ARSSERVR.ARSAPP
arssyscr:       Adding to ARSSERVR.ARSFOL
arssyscr:       Adding to ARSSERVR.ARSFOLPERMS
arssyscr:       Adding to ARSSERVR.ARSFOLFLD
arssyscr:       Adding to ARSSERVR.ARSFOLFLDUSR
arssyscr:       Creation of System Log information was successful
```

Results

If you encounter SQL errors after you perform this configuration step, check for the existence of the **MVSATTACHTYPE** parameter in the `cli.ini` file. The MVSATTACHTYPE=RRSAF parameter value is not supported by Content Manager OnDemand. Only the CAF option is allowed. Modify **MVSATTACHTYPE** from RRSAF to CAF, or remove the **MVSATTACHTYPE** parameter because CAF is the default.

Chapter 23. Initializing the system load logging facility

About this task

Content Manager OnDemand provides an optional logging facility to enable tracking Content Manager OnDemand loading activity. When you enable load logging, Content Manager OnDemand stores the messages that are generated by Content Manager OnDemand load programs in the system load log. You use one of the Content Manager OnDemand client programs to search for and filter messages by load date, application group name, load ID, input file name, and other parameters.

Procedure

To initialize the Content Manager OnDemand system load logging facility, do the following steps:

1. Change to the Content Manager OnDemand executable directory.
   For example: `cd /usr/lpp/ars/V10R5M0/bin`
2. Run the ARSSYSCR program
   For example: `./arssyscr -a`.

   If you change the default instance from ARCHIVE to another value, specify the instance on the **ARSSYSCR** command. Otherwise, you might receive the following error: `arssyscr unable to determine database engine`.

   The ARSSYSCR program generates the following series of messages:

```
arssyscr: Updating ARSSERVR.ARSSYS
arssyscr: Adding to ARSSERVR.ARSAG with Storage Set Id = 0
arssyscr: Adding to ARSSERVR.ARSAGPERMS
arssyscr: Adding to ARSSERVR.ARSAGFLD
arssyscr: Adding to ARSSERVR.ARSAGFLDALIAS
arssyscr: Adding to ARSSERVR.ARSAG2FOL
arssyscr: Adding to ARSSERVR.ARSAPPUSR
arssyscr: Adding to ARSSERVR.ARSAPP
arssyscr: Adding to ARSSERVR.ARSFOL
arssyscr: Adding to ARSSERVR.ARSFOLPERMS
arssyscr: Adding to ARSSERVR.ARSFOLFLD
arssyscr: Adding to ARSSERVR.ARSFOLFLDUSR
arssyscr: Creation of System Load information was successful
```

Results

If you encounter SQL errors after you perform this configuration step, check for the existence of the **MVSATTACHTYPE** parameter in the `cli.ini` file. The MVSATTACHTYPE=RRSAF parameter value is not supported by Content Manager OnDemand. Only the CAF option is allowed. Modify **MVSATTACHTYPE** from RRSAF to CAF, or remove the **MVSATTACHTYPE** parameter because CAF is the default.

Chapter 24. Initializing system migration

About this task

Note: You must initialize the system migration facility only if you plan to migrate application group index data from the database to archive storage.

After you have successfully created the Content Manager OnDemand system tables, you can run the ARSSYSCR program to initialize the Content Manager OnDemand system migration component:

Procedure

To run the ARSSYSCR program, do the following steps:

1. Move to the Content Manager OnDemand executable directory.
   For example: `cd /usr/lpp/ars/V10R5M0/bin`
2. Run the ARSSYSCR program.
   For example: `arssyscr -m`.

   If you change the default instance from ARCHIVE to another value, specify the instance on the **ARSSYSCR** command. Otherwise, you might receive the following error: `arssyscr unable to determine database engine`.

   The ARSSYSCR program will generate the following series of messages:

```
arssyscr:       Updating ARSSERVR.ARSSYS
arssyscr:       Adding to ARSSERVR.ARSAG with Storage Set Id = 0
arssyscr:       Adding to ARSSERVR.ARSAGPERMS
arssyscr:       Adding to ARSSERVR.ARSAGFLD
arssyscr:       Adding to ARSSERVR.ARSAGFLDALIAS
arssyscr:       Adding to ARSSERVR.ARSAG2FOL
arssyscr:       Adding to ARSSERVR.ARSAPPUSR
arssyscr:       Adding to ARSSERVR.ARSAPP
arssyscr:       Adding to ARSSERVR.ARSFOL
arssyscr:       Adding to ARSSERVR.ARSFOLPERMS
arssyscr:       Adding to ARSSERVR.ARSFOLFLD
arssyscr:       Adding to ARSSERVR.ARSFOLFLDUSR
arssyscr:       Creation of System Migration information was successful
```

Results

If you encounter SQL errors after you perform this configuration step, check for the existence of the **MVSATTACHTYPE** parameter in the `cli.ini` file. The MVSATTACHTYPE=RRSAF parameter value is not supported by Content Manager OnDemand. Only the CAF option is allowed. Modify **MVSATTACHTYPE** from RRSAF to CAF, or remove the **MVSATTACHTYPE** parameter because CAF is the default.

Part 5. Configuring other external storage solutions

Content Manager OnDemand supports external cloud storage managers such as Amazon Simple Storage Service (S3), Apache Hadoop Distributed File System (HDFS), Hitachi Content Platform, IBM Cloud Object Storage, Microsoft Azure, and OpenStack Swift. Content Manager OnDemand can also store data to external file systems.

Cloud storage options

The Content Manager OnDemand server can be configured to maintain copies of its stored data in both cache storage, managed by Content Manager OnDemand, and in archive storage, now referred to as external storage. Historically, OAM and VSAM have been the only options used by Content Manager OnDemand to maintain data stored in external storage. The addition of Amazon S3, Apache HDFS, Hitachi Content Platform, IBM Cloud Object Storage, Microsoft Azure, and OpenStack Swift support augments the storage capabilities of Content Manager OnDemand by providing multiple external cloud storage solution options. Cloud storage solutions allow Content Manager OnDemand users to leverage the advantages that such storage provides such as cost savings, data replication, and disaster recovery. This functionality is configured in Content Manager OnDemand and behaves much in the same way that communicating with OAM and VSAM does. This means that data in Content Manager OnDemand can be stored in cache as well as stored in Amazon S3, Apache HDFS, Hitachi Content Platform, IBM Cloud Object Storage, Microsoft Azure, or OpenStack Swift. The storing of data to any external cloud storage manager can take place at the same time that data is written to the Content Manager OnDemand cache or can be scheduled to migrate at a later date.

Amazon S3, Apache HDFS, Hitachi Content Platform, IBM Cloud Object Storage, Microsoft Azure, and OpenStack Swift storage options complement the functionality provided by OAM and VSAM. Content Manager OnDemand servers can be configured to use any combination of Amazon S3, Apache HDFS, Hitachi Content Platform, IBM Cloud Object Storage, Microsoft Azure, OpenStack Swift, OAM, and VSAM.

Additional information for each cloud storage solution can be found on the web at the following locations:

Amazon S3
https://aws.amazon.com/s3/

Apache HDFS
https://hadoop.apache.org/

Hitachi Content Platform
https://www.hitachivantara.com/en-us/products/cloud-object-platform/content-platform.html

IBM Cloud Object Storage
https://www.ibm.com/cloud-computing/infrastructure/object-storage/

Microsoft Azure
https://azure.microsoft.com/

OpenStack Swift
http://docs.openstack.org/developer/swift/

Using a file system for external storage

The Content Manager OnDemand server can be configured to maintain copies of its stored data in a file system accessible to Content Manager OnDemand. As with the cloud storage options, the storing of data to an external file system can take place at the same time that data is written to the Content Manager OnDemand cache or can be scheduled to migrate at a later date.

Chapter 25. Configuring an Amazon S3 external storage manager

Content Manager OnDemand supports data storage in an Amazon Simple Storage Service (S3) repository. More information on Amazon S3 can be found at: https://aws.amazon.com/s3/

Updating the ARS.CFG file

Perform these steps to configure Amazon S3 on a z/OS server.

1. Two new entries must be added to the ARS.CFG file.

   ```
   ARS_S3_CONFIG_FILE=/usr/lpp/ars/V10R5M0/config/ars.s3
   ARS_S3_CONFIG_DIR=/usr/lpp/ars/V10R5M0/config
   ```

   The ARS_S3_CONFIG_FILE entry specifies an existing Amazon S3 configuration file which the server uses by default.

   The ARS_S3_CONFIG_DIR entry specifies the directory in which any alternate configuration files are kept. This directory is used if additional Amazon S3 configuration files are defined. The names of these additional configuration files can be specified when defining storage nodes in Content Manager OnDemand. If no configuration file is specified in the storage node, the default configuration file is used.

   The configuration file name and directory path shown in the examples are the recommended values for these entries.

2. The ARS_STORAGE_MANAGER entry in the ARS.CFG file might also need to be changed. If you specify ARS_STORAGE_MANAGER=CACHE_ONLY, this disables all storage managers supported by Content Manager OnDemand.

   To configure the Content Manager OnDemand server to use Amazon S3 as a storage manager, the value must be set to the following:

   ARS_STORAGE_MANAGER=NO_TSM
   This setting will enable all external storage managers supported by Content Manager OnDemand except Tivoli Storage Manager, which is not supported on z/OS. This setting is used when the additional software to support Tivoli Storage Manager is not installed and Tivoli Storage Manager is not required as an external storage manager.

Creating an Amazon S3 configuration file

An Amazon S3 configuration file for Content Manager OnDemand contains entries specific to your Amazon S3 implementation. You specify the location and name of the default configuration file in the ARS.CFG entry. Required entries must be specified. Optional entries are not required in the configuration file unless those values need to be changed.

The following list describes the entries that can be specified in an Amazon S3 configuration file.

ARS_S3_SERVER
Specifies the Amazon S3 server name. Do not include `http://` or `https://` in the name. This entry is required.

ARS_S3_REGION
Specifies the Amazon S3 region. This entry is required.

ARS_S3_USE_SSL
Indicates whether or not to use SSL in server communications. The possible values are:

- `0` - SSL will not be used
- `1` - SSL will be used

The default value is 1. This entry is optional.

ARS_S3_CONNECT_TIMEOUT
Specifies the maximum number of seconds that Content Manager OnDemand waits for a response from the storage manager. The default is 60. This entry is optional. **Warning:** Setting this value too low might cause connection failures.

ARS_S3_HLD
Specifies the high-level directory name. This attribute is available to group sets of Content Manager OnDemand data together which might be needed if sharing external storage among multiple Content Manager OnDemand servers. **Warning:** Once this value is set, it must not be changed. If it is changed, any data that is already stored will not be retrievable. There is no default value. This entry is optional.

As an example, for a URL such as `https://s3-us-west-2.amazonaws.com/`, the Amazon S3 configuration file contains:

```
ARS_S3_SERVER=s3.amazonaws.com
ARS_S3_REGION=us-west-2
```

Defining an Amazon S3 storage node with the Administrator client

You can define the settings for using the Amazon S3 access method on the **Add a Primary Node** dialog of the OnDemand Administrator client.

The Storage Node field becomes Bucket Name when the Access Method is set to Amazon S3. The bucket name must exist in your repository or access to Amazon S3 will fail.

The Logon and Password fields contain the Amazon S3 access key and password that Content Manager OnDemand needs to access your Amazon S3 repository.

The Access Method radio button is set to Amazon S3. The Configuration File Name defaults to the value specified by the ARS_S3_CONFIG_FILE parameter in the ARS.CFG file if no value is entered. Otherwise, Content Manager OnDemand looks for the configuration file in the directory defined by the ARS_S3_CONFIG_DIR parameter specified in the ARS.CFG file.

Chapter 26. Configuring an Apache HDFS external storage manager

Content Manager OnDemand supports data storage in an Apache Hadoop Distributed File System (HDFS).

The Apache® Hadoop® project develops a variety of open-source software for reliable, scalable, distributed computing. The project includes Apache HDFS, which is a distributed file system that provides high-throughput access to application data. More information on Apache HDFS can be found at: https://hadoop.apache.org/

Updating the ARS.CFG file

Perform these steps to configure Apache HDFS on a z/OS server.

1. Two new entries must be added to the ARS.CFG file.

   ```
   ARS_HDFS_CONFIG_FILE=/usr/lpp/ars/V10R5M0/config/ars.hdfs
   ARS_HDFS_CONFIG_DIR=/usr/lpp/ars/V10R5M0/config
   ```

   The `ARS_HDFS_CONFIG_FILE` entry specifies an existing Apache HDFS configuration file which the server uses by default.

   The `ARS_HDFS_CONFIG_DIR` entry specifies the directory in which any alternate configuration files are kept. This directory is used if additional Apache HDFS configuration files are defined. The names of these additional configuration files can be specified when defining storage nodes in Content Manager OnDemand. If no configuration file is specified in the storage node, the default configuration file is used.

   The configuration file name and directory path shown in the examples are the recommended values for these entries.

2. The `ARS_STORAGE_MANAGER` entry in the ARS.CFG file might also need to be changed. If you specify `ARS_STORAGE_MANAGER=CACHE_ONLY`, this disables all storage managers supported by Content Manager OnDemand.

   To configure the Content Manager OnDemand server to use Apache HDFS as a storage manager, the value must be set to the following:

   ARS_STORAGE_MANAGER=NO_TSM
   This setting will enable all external storage managers supported by Content Manager OnDemand except Tivoli Storage Manager, which is not supported on z/OS. This setting is used when the additional software to support Tivoli Storage Manager is not installed and Tivoli Storage Manager is not required as an external storage manager.

Creating an Apache HDFS configuration file

An Apache HDFS configuration file for Content Manager OnDemand contains entries specific to your Apache HDFS implementation. You specify the location and name of the default configuration file in the ARS.CFG entry. Required entries must be specified. Optional entries are not required in the configuration file unless those values need to be changed.

The following list describes the entries that can be specified in an Apache HDFS configuration file.

ARS_HDFS_SERVER
Specifies the Apache HDFS server name. Do not include `http://` or `https://` in the name. This entry is required.

ARS_HDFS_PORT
Specifies the Apache HDFS server port number. This entry is optional if using a standard port. Content Manager OnDemand assumes port 80 for HTTP or port 443 for HTTPS communications.

ARS_HDFS_TLD
Specifies the Apache HDFS top-level directory name. This is any additional path information after the server name and port in the URL. This entry is optional.

ARS_HDFS_USE_SSL
Indicates whether or not to use SSL in server communications. The possible values are:

- 0 - SSL will not be used
- 1 - SSL will be used

The default value is 0. This entry is optional.

ARS_HDFS_AUTH_TYPE
Specifies the user authentication type. The possible values are:

- NONE - Open system
- KNOX - Access and authenticate through Apache Knox

The default value is NONE. This entry is optional.

ARS_HDFS_CONNECT_ TIMEOUT
Specifies the maximum number of seconds that Content Manager OnDemand waits for a response from the storage manager. The default is 60. This entry is optional. **Warning:** Setting this value too low might cause connection failures.

ARS_HDFS_FILE_PERMS
Specifies the permissions for new files. The default is 440. This entry is optional.

ARS_HDFS_HLD
Specifies the high-level directory name. This attribute is available to group sets of Content Manager OnDemand data together which might be needed if sharing external storage among multiple Content Manager OnDemand servers. **Warning:** Once this value is set, it must not be changed. If it is changed, any data that is already stored will not be retrievable. There is no default value. This entry is optional.

As an example, for a URL such as `http://hdfs.example.com/webhdfs/v1`, the Apache HDFS configuration file contains:

```
ARS_HDFS_SERVER=hdfs.example.com
ARS_HDFS_TLD=/webhdfs/v1
```

Defining an Apache HDFS storage node with the Administrator client

You can define the settings for using the Apache HDFS access method on the **Add a Primary Node** dialog of the OnDemand Administrator client.

The Storage Node field is not used for communication with the Apache HDFS server and can be set to any name you choose.

The Logon field is the user name from the Apache HDFS system which Content Manager OnDemand uses to store and retrieve data. A password might not be required for open Apache HDFS systems, so this field is optional.

The Access Method radio button is set to Apache HDFS. The Configuration File Name defaults to the value specified by the ARS_HDFS_CONFIG_FILE parameter in the ARS.CFG file if no value is entered. Otherwise, Content Manager OnDemand looks for the configuration file in the directory defined by the ARS_HDFS_CONFIG_DIR parameter specified in the ARS.CFG file.

Chapter 27. Configuring a Hitachi Content Platform external storage manager

Content Manager OnDemand supports data storage in a Hitachi Content Platform repository. More information on Hitachi Content Platform can be found at: https://www.hitachivantara.com/en-us/products/cloud-object-platform/content-platform.html

Updating the ARS.CFG file

Perform these steps to configure Hitachi Content Platform on a z/OS server.

1. Two new entries must be added to the ARS.CFG file.

   ```
   ARS_HITACHI_CONFIG_FILE=/usr/lpp/ars/V10R5M0/config/ars.hcp
   ARS_HITACHI_CONFIG_DIR=/usr/lpp/ars/V10R5M0/config
   ```

   The ARS_HITACHI_CONFIG_FILE entry specifies an existing Hitachi Content Platform configuration file which the server uses by default.

   The ARS_HITACHI_CONFIG_DIR entry specifies the directory in which any alternate configuration files are kept. This directory is used if additional Hitachi Content Platform configuration files are defined. The names of these additional configuration files can be specified when defining storage nodes in Content Manager OnDemand. If no configuration file is specified in the storage node, the default configuration file is used.

   The configuration file name and directory path shown in the examples are the recommended values for these entries.

2. The ARS_STORAGE_MANAGER entry in the ARS.CFG file might also need to be changed. If you specify ARS_STORAGE_MANAGER=CACHE_ONLY, this disables all storage managers supported by Content Manager OnDemand.

   To configure the Content Manager OnDemand server to use Hitachi Content Platform as a storage manager, the value must be set to the following:

   ARS_STORAGE_MANAGER=NO_TSM
   This setting will enable all external storage managers supported by Content Manager OnDemand except Tivoli Storage Manager, which is not supported on z/OS. This setting is used when the additional software to support Tivoli Storage Manager is not installed and Tivoli Storage Manager is not required as an external storage manager.

Creating a Hitachi Content Platform configuration file

A Hitachi Content Platform configuration file for Content Manager OnDemand contains entries specific to your Hitachi Content Platform implementation. You specify the location and name of the default configuration file in the ARS.CFG entry. Required entries must be specified. Optional entries are not required in the configuration file unless those values need to be changed.

The following list describes the entries that can be specified in a Hitachi Content Platform configuration file.

ARS_HITACHI_SERVER
Specifies the Hitachi Content Platform server name. Do not include `http://` or `https://` in the name. This entry is required.

ARS_HITACHI_USE_SSL
Indicates whether or not to use SSL in server communications. The possible values are:

- `0` - SSL will not be used
- `1` - SSL will be used

The default value is 0. This entry is optional.

ARS_HITACHI_CONNECT_ TIMEOUT
Specifies the maximum number of seconds that Content Manager OnDemand waits for a response from the storage manager. The default is 60. This entry is optional. **Warning:** Setting this value too low might cause connection failures.

ARS_HITACHI_HLD
Specifies the high-level directory name. This attribute is available to group sets of Content Manager OnDemand data together which might be needed if sharing external storage among multiple Content Manager OnDemand servers. **Warning:** Once this value is set, it must not be changed. If it is changed, any data that is already stored will not be retrievable. There is no default value. This entry is optional.

As an example, for a URL such as `http://sample.hitachi-hcp.com/`, the Hitachi Content Platform configuration file contains:

```
ARS_HITACHI_SERVER=sample.hitachi-hcp.com
```

Defining a Hitachi Content Platform storage node with the Administrator client

You can define the settings for using the Hitachi Content Platform access method on the **Add a Primary Node** dialog of the OnDemand Administrator client.

The Storage Node field becomes Namespace when the Access Method is set to Hitachi Content Platform. The namespace must exist in your repository or access to Hitachi Content Platform will fail.

The Logon and Password fields contain the Hitachi Content Platform logon and password that Content Manager OnDemand needs to access your Hitachi Content Platform repository.

The Access Method radio button is set to `Hitachi Content Platform`. The Configuration File Name defaults to the value specified by the `ARS_HITACHI_CONFIG_FILE` parameter in the ARS.CFG file if no value is entered. Otherwise, Content Manager OnDemand looks for the configuration file in the directory defined by the `ARS_HITACHI_CONFIG_DIR` parameter specified in the ARS.CFG file.

Chapter 28. Configuring an IBM Cloud Object Storage external storage manager

Content Manager OnDemand supports data storage in an IBM Cloud Object Storage repository. More information on IBM Cloud Object Storage can be found at: https://www.ibm.com/cloud-computing/infrastructure/object-storage/

Updating the ARS.CFG file

Perform these steps to configure IBM Cloud Object Storage on a z/OS server.

1. Two new entries must be added to the ARS.CFG file.

   ```
   ARS_ICOS_CONFIG_FILE=/usr/lpp/ars/V10R5M0/config/ars.icos
   ARS_ICOS_CONFIG_DIR=/usr/lpp/ars/V10R5M0/config
   ```

   The `ARS_ICOS_CONFIG_FILE` entry specifies an existing IBM Cloud Object Storage configuration file which the server uses by default.

   The `ARS_ICOS_CONFIG_DIR` entry specifies the directory in which any alternate configuration files are kept. This directory is used if additional IBM Cloud Object Storage configuration files are defined. The names of these additional configuration files can be specified when defining storage nodes in Content Manager OnDemand. If no configuration file is specified in the storage node, the default configuration file is used.

   The configuration file name and directory path shown in the examples are the recommended values for these entries.

2. The `ARS_STORAGE_MANAGER` entry in the ARS.CFG file might also need to be changed. If you specify `ARS_STORAGE_MANAGER=CACHE_ONLY`, this disables all storage managers supported by Content Manager OnDemand.

   To configure the Content Manager OnDemand server to use IBM Cloud Object Storage as a storage manager, the value must be set to the following:

   ARS_STORAGE_MANAGER=NO_TSM
   This setting will enable all external storage managers supported by Content Manager OnDemand except Tivoli Storage Manager, which is not supported on z/OS. This setting is used when the additional software to support Tivoli Storage Manager is not installed and Tivoli Storage Manager is not required as an external storage manager.

Creating an IBM Cloud Object Storage configuration file

An IBM Cloud Object Storage configuration file for Content Manager OnDemand contains entries specific to your IBM Cloud Object Storage implementation. You specify the location and name of the default configuration file in the ARS.CFG entry. Required entries must be specified. Optional entries are not required in the configuration file unless those values need to be changed.

The following list describes the entries that can be specified in an IBM Cloud Object Storage configuration file.

ARS_ICOS_SERVER
Specifies the IBM Cloud Object Storage server name. Do not include `http://` or `https://` in the name. This entry is required.

ARS_ICOS_USE_SSL
Indicates whether or not to use SSL in server communications. The possible values are:

- `0` - SSL will not be used
- `1` - SSL will be used

The default value is 0. This entry is optional.

ARS_ICOS_CONNECT_ TIMEOUT
Specifies the maximum number of seconds that Content Manager OnDemand waits for a response from the storage manager. The default is 60. This entry is optional. **Warning:** Setting this value too low might cause connection failures.

ARS_ICOS_HLD
Specifies the high-level directory name. This attribute is available to group sets of Content Manager OnDemand data together which might be needed if sharing external storage among multiple Content Manager OnDemand servers. **Warning:** Once this value is set, it must not be changed. If it is changed, any data that is already stored will not be retrievable. There is no default value. This entry is optional.

As an example, for a URL such as `http://sample.cleversafe.com/`, the IBM Cloud Object Storage configuration file contains:

```
ARS_ICOS_SERVER=sample.cleversafe.com
```

Defining an IBM Cloud Object Storage storage node with the Administrator client

You can define the settings for using the IBM Cloud Object Storage access method on the **Add a Primary Node** dialog of the OnDemand Administrator client.

The Storage Node field becomes Vault Name when the Access Method is set to IBM Cloud Object Storage. The vault name must exist in your repository or access to IBM Cloud Object Storage will fail.

The Logon and Password fields contain the IBM Cloud Object Storage logon and password that Content Manager OnDemand needs to access your IBM Cloud Object Storage repository.

The Access Method radio button is set to `IBM Cloud Object Storage`. The Configuration File Name defaults to the value specified by the `ARS_ICOS_CONFIG_FILE` parameter in the ARS.CFG file if no value is entered. Otherwise, Content Manager OnDemand looks for the configuration file in the directory defined by the `ARS_ICOS_CONFIG_DIR` parameter specified in the ARS.CFG file.

Chapter 29. Configuring a Microsoft Azure external storage manager

Content Manager OnDemand supports data storage in a Microsoft Azure repository. Microsoft Azure is a highly available, distributed, eventually consistent object/blob store. You can use Microsoft Azure to store lots of data efficiently, safely, and inexpensively. More information on Microsoft Azure can be found at: https://azure.microsoft.com/

Updating the ARS.CFG file

Perform these steps to configure Microsoft Azure on a z/OS server.

1. Two new entries must be added to the ARS.CFG file.

   ```
   ARS_AZURE_CONFIG_FILE=/usr/lpp/ars/V10R5M0/config/ars.az
   ARS_AZURE_CONFIG_DIR=/usr/lpp/ars/V10R5M0/config
   ```

   The ARS_AZURE_CONFIG_FILE entry specifies an existing Azure configuration file which the server uses by default.

   The ARS_AZURE_CONFIG_DIR entry specifies the directory in which any alternate configuration files are kept. This directory is used if additional Azure configuration files are defined. The names of these additional configuration files can be specified when defining storage nodes in Content Manager OnDemand. If no configuration file is specified in the storage node, the default configuration file is used.

   The configuration file name and directory path shown in the examples are the recommended values for these entries.

2. The ARS_STORAGE_MANAGER entry in the ARS.CFG file might also need to be changed. If you specify ARS_STORAGE_MANAGER=CACHE_ONLY, this disables all storage managers supported by Content Manager OnDemand.

   To configure the Content Manager OnDemand server to use Azure as a storage manager, the value must be set to the following:

   ARS_STORAGE_MANAGER=NO_TSM
   This setting will enable all external storage managers supported by Content Manager OnDemand except Tivoli Storage Manager, which is not supported on z/OS. This setting is used when the additional software to support Tivoli Storage Manager is not installed and Tivoli Storage Manager is not required as an external storage manager.

Creating a Microsoft Azure configuration file

A Microsoft Azure configuration file for Content Manager OnDemand contains entries specific to your Azure implementation. You specify the location and name of the default configuration file in the ARS.CFG entry. Required entries must be specified. Optional entries are not required in the configuration file unless those values need to be changed.

The following list describes the entries that can be specified in an Azure configuration file.

ARS_AZURE_SERVER
Specifies the Microsoft Azure server name. Do not include `http://` or `https://` in the name. This entry is required.

ARS_AZURE_USE_SSL
Indicates whether or not to use SSL in server communications. The possible values are:

- `0` - SSL will not be used
- `1` - SSL will be used

The default value is 0. This entry is optional.

ARS_AZURE_CONNECT_ TIMEOUT
Specifies the maximum number of seconds that Content Manager OnDemand waits for a response from the storage manager. The default is 60. This entry is optional. **Warning:** Setting this value too low might cause connection failures.

ARS_AZURE_HLD
Specifies a high-level directory name. This attribute is available to group sets of Content Manager OnDemand data together which might be needed if sharing external storage among multiple Content Manager OnDemand servers. **Warning:** Once this value is set, it must not be changed. If it is changed, any data that is already stored will not be retrievable. There is no default value. This entry is optional.

As an example, the Azure configuration file might contain:

```
ARS_AZURE_SERVER=blob.core.windows.net
ARS_AZURE_USE_SSL=1
```

Defining a Microsoft Azure storage node with the Administrator client

You can define the settings for using the Microsoft Azure access method on the **Add a Primary Node** dialog of the OnDemand Administrator client.

The Storage Node field becomes Container Name when the Access Method is set to `Microsoft Azure`. The Container Name field is used with the Microsoft Azure server and determines the storage hierarchy for objects stored to this node. Containers are not created. They must exist before you add the storage node.

The Logon field is the Microsoft Azure user name which is used to store and retrieve data from the Microsoft Azure system. The password is also required.

The Access Method radio button is set to `Microsoft Azure`. The Configuration File Name defaults to the value specified by the `ARS_AZURE_CONFIG_FILE` parameter in the ARS.CFG file if no value is entered. Otherwise, Content Manager OnDemand looks for the configuration file in the directory defined by the `ARS_AZURE_CONFIG_DIR` parameter specified in the ARS.CFG file.

Chapter 30. Configuring an OpenStack Swift external storage manager

Content Manager OnDemand supports data storage in an OpenStack Swift repository. OpenStack Swift is a highly available, distributed, eventually consistent object/blob store. You can use OpenStack Swift to store lots of data efficiently, safely, and inexpensively. More information on OpenStack Swift can be found at: http://docs.openstack.org/developer/swift/

Updating the ARS.CFG file

Perform these steps to configure OpenStack Swift on a z/OS server.

1. Two new entries must be added to the ARS.CFG file.

   ```
   ARS_SWIFT_CONFIG_FILE=/usr/lpp/ars/V10R5M0/config/ars.swift
   ARS_SWIFT_CONFIG_DIR=/usr/lpp/ars/V10R5M0/config
   ```

   The ARS_SWIFT_CONFIG_FILE entry specifies an existing Swift configuration file which the server uses by default.

   The ARS_SWIFT_CONFIG_DIR entry specifies the directory in which any alternate configuration files are kept. This directory is used if additional Swift configuration files are defined. The names of these additional configuration files can be specified when defining storage nodes in Content Manager OnDemand. If no configuration file is specified in the storage node, the default configuration file is used.

   The configuration file name and directory path shown in the examples are the recommended values for these entries.

2. The ARS_STORAGE_MANAGER entry in the ARS.CFG file might also need to be changed. If you specify ARS_STORAGE_MANAGER=CACHE_ONLY, this disables all storage managers supported by Content Manager OnDemand.

   To configure the Content Manager OnDemand server to use Swift as a storage manager, the value must be set to the following:

   ARS_STORAGE_MANAGER=NO_TSM
   This setting will enable all external storage managers supported by Content Manager OnDemand except Tivoli Storage Manager, which is not supported on z/OS. This setting is used when the additional software to support Tivoli Storage Manager is not installed and Tivoli Storage Manager is not required as an external storage manager.

Creating an OpenStack Swift configuration file

An OpenStack Swift configuration file for Content Manager OnDemand contains entries specific to your Swift implementation. You specify the location and name of the default configuration file in the ARS.CFG entry. Required entries must be specified. Optional entries are not required in the configuration file unless those values need to be changed.

The following list describes the entries that can be specified in a Swift configuration file.

ARS_SWIFT_SERVER
Specifies the OpenStack Swift server name. Do not include `http://` or `https://` in the name. This entry is required.

ARS_SWIFT_PORT
Specifies the OpenStack Swift server port number. This entry is optional if using a standard port. Content Manager OnDemand assumes port 80 for HTTP or port 443 for HTTPS communications.

ARS_SWIFT_TLD
Specifies the OpenStack Swift top-level directory name. This contains any additional path information after the server name and port in the URL. This entry is optional but usually necessary.

ARS_SWIFT_USE_SSL
Indicates whether or not to use SSL in server communications. The possible values are:

- 0 - SSL will not be used
- 1 - SSL will be used

The default value is 0. This entry is optional.

ARS_SWIFT_CONNECT_ TIMEOUT
Specifies the maximum number of seconds that Content Manager OnDemand waits for a response from the storage manager. The default is 60. This entry is optional. **Warning:** Setting this value too low might cause connection failures.

ARS_SWIFT_HLD
Specifies a high-level directory name. This attribute is available to group sets of Content Manager OnDemand data together which might be needed if sharing external storage among multiple Content Manager OnDemand servers. **Warning:** Once this value is set, it must not be changed. If it is changed, any data that is already stored will not be retrievable. There is no default value. This entry is optional.

As an example, for a URL such as `https://swift.example.com:8088/v1/account/`, the Swift configuration file contains:

```
ARS_SWIFT_SERVER=swift.example.com
ARS_SWIFT_PORT=8088
ARS_SWIFT_TLD=/v1/account
ARS_SWIFT_USE_SSL=1
```

Defining an OpenStack Swift storage node with the Administrator client

You can define the settings for using the OpenStack Swift access method on the **Add a Primary Node** dialog of the OnDemand Administrator client.

The Storage Node field becomes Container Name when the Access Method is set to `OpenStack Swift`. The Container Name field is used with the OpenStack Swift server and determines the storage hierarchy for objects stored to this node. Containers are created if they do not already exist.

The Logon field is the OpenStack Swift user name which is used to store and retrieve data from the OpenStack Swift system. The password is also required.

The Access Method radio button is set to `OpenStack Swift`. The Configuration File Name defaults to the value specified by the `ARS_SWIFT_CONFIG_FILE` parameter in the ARS.CFG file if no value is entered. Otherwise, Content Manager OnDemand looks for the configuration file in the directory defined by the `ARS_SWIFT_CONFIG_DIR` parameter specified in the ARS.CFG file.

Chapter 31. Using a file system for external storage

Content Manager OnDemand supports data storage in a file system repository. The file system must be locally accessible to the Content Manager OnDemand library or object server.

Updating the ARS.CFG file

Perform these steps to configure Content Manager OnDemand to use a file system as external storage on a z/OS server.

1. Two new entries must be added to the ARS.CFG file.

   ```
   ARS_FILESYSTEM_CONFIG_FILE=/usr/lpp/ars/V10R5M0/config/ars.fs
   ARS_FILESYSTEM_CONFIG_DIR=/usr/lpp/ars/V10R5M0/config
   ```

   The ARS_FILESYSTEM_CONFIG_FILE entry specifies an existing file system configuration file which the server uses by default.

   The ARS_FILESYSTEM_CONFIG_DIR entry specifies the directory in which any alternate configuration files are kept. This directory is used if additional file system configuration files are defined. The names of these additional configuration files can be specified when defining storage nodes in Content Manager OnDemand. If no configuration file is specified in the storage node, the default configuration file is used.

   The configuration file name and directory path shown in the examples are the recommended values for these entries.
2. The ARS_STORAGE_MANAGER entry in the ARS.CFG file might also need to be changed. If you specify ARS_STORAGE_MANAGER=CACHE_ONLY, this disables all storage managers supported by Content Manager OnDemand.

   To configure the Content Manager OnDemand server to use a file system as external storage, the value must be set to the following:

   ARS_STORAGE_MANAGER=NO_TSM
   This setting will enable all external storage managers supported by Content Manager OnDemand except Tivoli Storage Manager, which is not supported on z/OS. This setting is used when the additional software to support Tivoli Storage Manager is not installed and Tivoli Storage Manager is not required as an external storage manager.

Creating a file system configuration file

A file system configuration file for Content Manager OnDemand contains entries specific to your file system implementation. You specify the location and name of the default configuration file in the ARS.CFG entry. Required entries must be specified. Optional entries are not required in the configuration file unless those values need to be changed.

The following list describes the entries that can be specified in a file system configuration file.

ARS_FILESYSTEM_NAME
Specifies the name of the file system for Content Manager OnDemand to use as a storage location. The file system permissions and ownership must be set to allow Content Manager OnDemand to read, write, and delete data to and from this location. This entry is required.

ARS_FILESYSTEM_HLD
Specifies the high-level directory name. This attribute is available to group sets of Content Manager OnDemand data together which might be needed if sharing this external storage among multiple Content Manager OnDemand servers. **Warning:** Once this value is set, it must not be changed. If it is changed, any data that is already stored may not be retrievable. There is no default value. This entry is optional.

As an example:

```
ARS_FILESYSTEM_NAME=/ondemand/fs
```

Defining a file system storage node with the Administrator client

You can define the settings for using the file system access method on the **Add a Primary Node** dialog of the OnDemand Administrator client.

The Storage Node field can be set to any name you choose. It is only used internally by Content Manager OnDemand.

The Logon and Password fields are not used.

The Access Method radio button is set to `Use a file system`. The Configuration File Name defaults to the value specified by the `ARS_FILESYSTEM_CONFIG_FILE` parameter in the ARS.CFG file if no value is entered. Otherwise, Content Manager OnDemand looks for the configuration file in the directory defined by the `ARS_FILESYSTEM_CONFIG_DIR` parameter specified in the ARS.CFG file.

Part 6. Enabling native encryption

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 a PKCS#12-compliant keystore, which is a storage object for encryption keys that exists at the operating system level. The keystore is only needed with the Content Manager OnDemand library server; it is not needed with object server(s).

Chapter 32. Configuring encryption support

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.

The cryptographic services include:

- Application programming interfaces (APIs) for applications that need to perform cryptographic 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 IBM z/OS Knowledge Center in the *z/OS Cryptographic Services Integrated Cryptographic Service Facility 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 instancename -d "keystore_location=CKDS,keystore_mkl=*"
```

where `instancename` is the name of your Content Manager OnDemand instance.

If you are 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.instancename.dbowner.yyyy.mm.dd.hh.mm.ss.tttttt
```

where `instancename` is the name of your Content Manager OnDemand instance.

This allows SAF profiles to be created of the form ONDEMAND.instancename.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 the IBM z/OS Knowledge Center topic titled *CKDS management in a sysplex* in the *z/OS Cryptographic Services Integrated Cryptographic Service Facility Administrator's Guide* for considerations about sharing the CKDS.

The *Cryptographic Services ICSF: System Programmer's Guide* in the IBM z/OS Knowledge Center 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 communicate with the library server directly.

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

Chapter 33. Enabling encryption in 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 of 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.

Chapter 34. Backing up your Content Manager OnDemand instance

You must backup your Content Manager OnDemand instance database, as well as the ICSF CKDS keystore, 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.

Part 7. Preparing the system for use

This section of the book describes the set of tasks that you should complete before you allow other users to begin using the system.

- Create storage sets. You must add storage sets to the system before you can create application groups or assign the system-defined application groups to a storage set. Depending on the storage management characteristics of the reports that you plan to store on the system, you may need to add more than one storage set.
- Configure the System Log application group. IBM recommends that you assign the System Log application group to a storage location in archive storage (optionally VSAM, OAM or Tivoli Storage Manager) so that the system can maintain a permanent copy of the system log data.
- Configure the System Migration application group. If you plan to migrate index data from the database to archive storage, then you must create a storage set that specifies a storage location in archive storage. After you add the storage set to the system, you can assign the System Migration application group to the storage set.
- Create a backup copy of the database. After installing and configuring Content Manager OnDemand, IBM recommends that you create a backup copy of the Content Manager OnDemand database.
- Install the Content Manager OnDemand administration interface.

Chapter 35. Defining storage sets

About this task

You must define storage sets before you can define reports to Content Manager OnDemand or load data into the system.

You can define storage sets that store data to cache storage or archive storage (or both). The storage management attributes of the application groups that you add to the system will determine the types of media that you need and how you configure storage sets on the system.

A storage set must contain at least one primary storage node. Primary storage can be cache storage (the default) or archive storage (or both). The administrative client online help provides details about defining storage sets and storage nodes.

If you plan to migrate index data to archive storage, then you must assign the System Migration application group to archive storage. Also, we recommend that you assign the System Log application group to archive storage so that the system can maintain a permanent copy of the data that is written to the system log. The following topics provide additional details:

- Chapter 36, “Configuring the System Log application group,” on page 117
- Chapter 38, “Configuring the System Migration application group,” on page 121

Chapter 36. Configuring the System Log application group

About this task

When you install and configure Content Manager OnDemand, you initialize the system log. The system log comprises the System Log application group, a set of system log applications, and the System Log folder. The System Log application group contains the storage management information that Content Manager OnDemand uses to maintain the data written to the system log. When you initialize the system, the application group is not assigned to a storage set. Because the application group is not assigned to a storage set, the system does not maintain a permanent copy of the system log data.

Before you begin defining reports to Content Manager OnDemand, loading data on the system, or allowing users to access the system, we recommend that you configure the System Log application group to maintain a permanent copy of the data that is written to the system log. You can do this by first defining a storage set to archive storage and then by updating the System Log application group and assigning it to the storage set. For more information, see Part 7, “Preparing the system for use,” on page 113 and Chapter 36, “Configuring the System Log application group,” on page 117.

If your system does not use archive storage, then you should assign the System Log application group to a cache-only storage set and change the length of time that Content Manager OnDemand maintains the system log data to the maximum permitted value. For more information, see “Maintaining system log data in archive storage” on page 117.

Maintaining system log data in archive storage

About this task

We recommend that you create a storage set that specifies a storage location in archive storage. You must add at least one primary storage node to the storage set. The primary storage node must identify a storage location in archive storage that maintains data indefinitely.

After you create the storage set, you must update the System Log application group and assign it to the storage set. After assigning the application group to the storage set and restarting the system, the system automatically maintains a copy of the system log data in archive storage.

Procedure

To assign the System Log application group to a storage set, do the following steps:

1. Start the administrative client.
2. Log on to the server with a userid that has system administrator authority.
   The built-in user ID admin has system administrator authority.
3. Click Application Groups.
4. Point to the System Log application group and click the right mouse button.
5. From the pop-up menu, select **Update** to open the **Update an Application Group** window.
6. Click the **Storage Management** tab.
7. In the **Storage Set Name** list, select the name of the storage set. The storage set that you select should specify archive storage policy domain that maintains data indefinitely.
8. Click **Advanced** to open the **Advanced Storage Management** dialog box.

9. Select **Next Cache Migration** under **Migrate Data from Cache**. This causes Content Manager OnDemand to copy the system log data to archive storage the next time that the **ARSMAINT** command runs.
10. Click **OK** to close the **Advanced Storage Management** dialog box.
11. Click **OK** to save your changes and close the **Update an Application Group** window.

    The administrative client online help provides information about the options on the **Storage Management** page.

Maintaining system log data in cache storage

About this task

If your system does not use archive storage, then you should assign the System Log application group to a cache-only storage set and change the length of time that Content Manager OnDemand maintains the data to the maximum permitted value. Doing so ensures that Content Manager OnDemand does not delete the data from cache storage for a very long time.

Procedure

To configure the System Log application group, do the following steps:

1. Start the administrative client.
2. Log on to the server with a userid that has system administrator authority.

   The built-in user ID admin has system administrator authority.
3. Click **Application Groups**.
4. Point to the System Log application group and click the right mouse button.
5. From the pop-up menu, select **Update** to open the **Update an Application Group** window.
6. Click the **Storage Management** tab.
7. In the **Storage Set Name** list, select the name of the storage set. The storage set named Cache Only - Library Server is a cache-only storage set created on the library server when you initialized the system.
8. Replace the contents of the **Cache Data for ___ Days** field with 99999.

   This value causes Content Manager OnDemand to maintain data for approximately 273 years.
9. Click **OK**.

   The administrative client online help provides information about the options on the Storage Management page.

Chapter 37. Configuring the System Load application group

About this task

When you install and configure Content Manager OnDemand, you can optionally initialize the system load logging facility. The system load logging facility comprises the System Load application group, a set of system load applications, and the System Load folder. The System Load application group contains the storage management information that Content Manager OnDemand uses to maintain the data written to the system load logging facility. When you initialize the system, the application group is not assigned to a storage set. Because the application group is not assigned to a storage set, the system does not maintain a permanent copy of the system load data.

Before you begin defining reports to Content Manager OnDemand, loading data on the system, or allowing users to access the system, you should configure the System Load application group to maintain a permanent copy of the data that is written to the system load logging facility. You can do this by first defining a storage set that specifies a client node in storage managed by a storage manager (Tivoli Storage Manager or OAM) and then by updating the System Load application group and assigning it to the storage set. For more information, see Part 7, “Preparing the system for use,” on page 113 and Chapter 37, “Configuring the System Load application group,” on page 119.

If your system does not use an archive manager, you should assign the System Load application group to a cache-only storage set and change the length of time that Content Manager OnDemand maintains the system load data to the maximum permitted value. For more information, see “Maintaining system load data in archive storage” on page 119.

Maintaining system load data in archive storage

About this task

You should create a storage set that specifies a client node in storage managed by the archive manager (Tivoli Storage Manager or OAM). You must add at least one primary storage node to the storage set. The primary storage node must be set up to maintain data indefinitely.

If you create the storage set in OAM, create a storage set that specifies a node in storage that is managed by OAM. You must add at least one primary storage node to the storage set. The primary storage node must be configured to maintain data indefinitely.

If you create a storage set in Tivoli Storage Manager, create a storage set that specifies a client node in storage that managed by Tivoli Storage Manager. You must add at least one primary storage node to the storage set. The primary storage node must identify a client node in Tivoli Storage Manager that maintains data indefinitely. The Logon Name and Password in the primary storage node must be identical to the client node and password in Tivoli Storage Manager.

When you register the client node in Tivoli Storage Manager, you must specify the name of the Tivoli Storage Manager policy domain that maintains data on the required media for the required length of time. If you do not specify the name of a domain, Tivoli Storage Manager assigns the client node to the default domain.

After you create the storage set, you must update the System Load application group and assign it to the storage set. After assigning the application group to the storage set and restarting the system, the system automatically maintains a copy of the system load data in archive storage.

Procedure

To assign the System Load application group to a storage set, do the following steps:

1. Start the administrative client.
2. Log on to the server with a user ID that has system administrator authority.

   The built-in user ID admin has system administrator authority.
3. Click **Application Groups**.
4. Point to the System Load application group and click the right mouse button.
5. From the pop-up menu, select **Update** to open the **Update an Application Group** window.
6. Click the **Storage Management** tab.
7. In the **Storage Set Name** list, select the name of the storage set. The storage set that you select should specify a policy that maintains data indefinitely.
8. Click **Advanced** to open the **Advanced Storage Management** dialog box.
9. Select **Next Cache Migration** under **Migrate Data from Cache**.

   This causes Content Manager OnDemand to copy the system load data to archive storage the next time that the **ARSMAINT** command runs.
10. Click **OK** to close the **Advanced Storage Management** dialog box.
11. Click **OK** to save your changes and close the **Update an Application Group** window.

    The adminstrative client online help provides information about the options on the **Storage Management** page.

Maintaining system load data in cache storage

About this task

If your system does not use an archive storage manager, you should assign the System Load application group to a cache-only storage set and change the length of time that Content Manager OnDemand maintains the data to the maximum permitted value. Doing so ensures that Content Manager OnDemand does not delete the data from cache storage for a very long time.

Procedure

To configure the System Load application group, do the following steps:

1. Start the administrative client.
2. Log on to the server with a user ID that has system administrator authority.

   The built-in userid admin has system administrator authority.
3. Click **Application Groups**.
4. Point to the System Load application group and click the right mouse button.
5. From the pop-up menu, select **Update** to open the **Update an Application Group** window.
6. Click the **Storage Management** tab.
7. In the **Storage Set Name** list, select the name of the storage set. The storage set named Cache Only - Library Server is a cache-only storage set created on the library server when you initialized the system.
8. Replace the contents of the **Cache Data for ___ Days** field with 99999.

   This value causes Content Manager OnDemand to maintain data for approximately 273 years.
9. Click **OK**.

   The administrative client online help provides information about the options on the **Storage Management** page.

Chapter 38. Configuring the System Migration application group

About this task

System migration is the process by which Content Manager OnDemand moves index data from the database to archive storage. This process optimizes database storage space while allowing you to maintain index data for a very long time. You typically migrate index data after users no longer need to access the reports, but for legal or other requirements, you still need to maintain the data for some number of years or months. Content Manager OnDemand uses the storage management settings in application groups to determine whether or not to migrate index data to archive storage. All migrated data is managed through the System Migration application group.

When you install and configure Content Manager OnDemand, you initialize the system migration function. The system migration function comprises the System Migration application group, a set of system migration applications, and the System Migration folder. The System Migration application group contains the storage management information that Content Manager OnDemand uses to maintain index data migrated to archive storage. Until you assign the application group to a storage set that specifies a storage location in archive storage, Content Manager OnDemand cannot migrate index data from the database to archive storage. For more information, see Part 7, “Preparing the system for use,” on page 113 and Chapter 38, “Configuring the System Migration application group,” on page 121.

Note: Depending on the Expiration Type that is specified for an application group, it may be necessary for the indexes to be online (in the database) before index expiration processing can complete successfully.

Assign the System Migration application group to a storage set

About this task

If you need the system to maintain index data in archive storage, then you must assign the System Migration application group to a storage set that identifies a storage location in archive storage.

After you define the storage set, you must update the System Migration application group and assign it to the storage set. After assigning the application group to the storage set and restarting the system, the system automatically migrates index data to archive storage, whenever migration processing (the ARSMAINT command) runs.

Procedure

To assign the System Migration application group to a storage set, do the following steps:

1. Start the administrative client.
2. Log on to the server with a userid that has administrator authority.

   The built-in userid `admin` has system administrator authority.
3. Click **Application Groups**.
4. Point to the System Migration application group and click the right mouse button.
5. From the pop-up menu, select **Update** to open the **Update an Application Group** window.
6. Click the **Storage Management** tab.
7. In the **Storage Set Name** list, select the name of the storage set. The storage set that you select should identify an archive storage policy domain that maintains data indefinitely.
8. Click **OK**.

The administrative client online help provides information about the options on the **Storage Management** page.

Chapter 39. Back up the Content Manager OnDemand database

A full backup image of the Content Manager OnDemand database is required to rebuild the database, in the event that you need to do so. You cannot rebuild the database unless you have a full database backup. For information on how to make a full backup, refer to the manuals provided with the DB2 product. Also see the data administrators for your installation.

Chapter 40. Installing the Content Manager OnDemand administration interface

About this task

The XML interface is a batch administrative interface. You can create administrative objects in XML format and import them into a Content Manager OnDemand system. You can also use the XML interface to export and import administrative objects from and into another Content Manager OnDemand system. This interface enables you to export all administrative objects into a single XML file, and later import the objects into the same Content Manager OnDemand system or another system.

For more information about importing and exporting administrative objects, see "Importing and exporting administrative objects through an XML interface" in the *IBM Content Manager OnDemand for z/OS: Administration Guide*.

Provided files

The following files are created in the `-PathPrefix-/usr/lpp/ars/V10R5M0` directory:

- `/bin/arsxml`
- `/bin/xml/ondemand.xsd` – Content Manager OnDemand schema definition in UTF-8
- `/bin/xml/ondemand.ebcdic.xsd` – Content Manager OnDemand schema definition in IBM-1047
- `/bin/xml/samples/addgroups.xml` – Sample to add groups in UTF-8
- `/bin/xml/samples/addgroups.ebcdic.xml` – Sample to add groups in IBM-1047
- `/bin/xml/samples/addusers.xml` – Sample to add users in UTF-8
- `/bin/xml/samples/addusers.ebcdic.xml` – Sample to add groups in IBM-1047
- `/bin/xml/samples/deletegroups.xml` – Sample to delete groups in UTF-8
- `/bin/xml/samples/deletegroups.ebcdic.xml` – Sample to delete groups in IBM-1047
- `/bin/xml/samples/deleteusers.xml` – Sample to delete users in UTF-8
- `/bin/xml/samples/deleteusers.ebcdic.xml` – Sample to delete users in IBM-1047
- `/bin/xml/samples/exportgroups.xml` – Sample to export groups in UTF-8
- `/bin/xml/samples/exportgroups.ebcdic.xml` – Sample to export groups in IBM-1047
- `/bin/xml/samples/exportusers.xml` – Sample to export users in UTF-8
- `/bin/xml/samples/exportusers.ebcdic.xml` – Sample to delete groups in IBM-1047
- `/bin/xml/samples/updategroups.xml` – Sample to update groups in UTF-8
- `/bin/xml/samples/updategroups.ebcdic.xml` – Sample to update groups in IBM-1047
- `/bin/xml/samples/updateusers.xml` – Sample to update users in UTF-8
- `/bin/xml/samples/updateusers.ebcdic.xml` – Sample to update users in IBM-1047

Running the arsxml IVP

About this task

The samples create and delete users SampleUser, SampleUser1, SampleUser2, and SampleUser3; printer LabPrinter; and groups SampleGroup, SampleGroup1, SampleGroup2, SampleGroup3, SampleGroup4. Verify that these do not exist before you run the samples.

Procedure

To verify that the batch administration interface has been properly installed, do the following actions:

1. Go into a UNIX System Services shell.
2. Change to the directory containing the samples.
   For example, `cd /usr/lpp/ars/V10R5M0/bin/xml/samples`.
3. Run this command: `/usr/lpp/ars/V10R5M0/bin/arsxml add -h` *`instname`* `-u` *`user`* `-p` *`password`* `-i addusers.ebcdic.xml -v`

   The following list describes the values specified for each parameter in the command:

   instname
   Content Manager OnDemand Instance name.

   user
   Specifies a Content Manager OnDemand user which has authority to add users and printers.

   password
   Specifies the password for the Content Manager OnDemand user. If the user does not have a password, leave this blank.

   Note:

   a. The XML file contains user passwords that are between 6 and 8 characters long. If the Content Manager OnDemand system where these users are created has different password restrictions, these passwords might need to be changed before running the command.

   b. If you use the Content Manager OnDemand security exit to verify that user IDs are in the RACF data base, the user IDs need to be changed to conform with RACF requirements.

   After running the command, you should receive several messages stating that a printer, and five users have been added successfully.

   If the command runs properly, you can use the Content Manager OnDemand administrative client to view the newly added users and printer.
4. To remove the newly added object, run the following command: `/usr/lpp/ars/V10R5M0/bin/arsxml delete -h` *`instname`* `-u` *`user`* `-p` *`password`* `-i deleteusers.xml -v`

Results

There are several other sample XML files that can be used. However, they all depend on the users that are created with the `addusers.xml` file.

Part 8. Verifying the installation

The IVP uses the storage set “Cache only – Library Storage” and user ID ADMIN that you created during product installation.

You might need to perform some customization of the sample commands and jobs to match local naming conventions. See the IVP instructions for each task to be performed.

The documentation and samples use `/usr/lpp/ars/V10R5M0` as the default mount point. If Content Manager OnDemand is installed in an alternate mount point, change the **PATH** environment variable to match the mount point.

Chapter 41. IVP results

The IVP generates the following files:

Definitions for the sample folders, application groups, and applications

- ARSIVPR1 Checking account statements
- ARSIVPR2 Loan delinquency report
- ARSIVPR3 Baxter Bay report (ACIF sample)
- ARSIVPR4 TIFF report

These files are created by the `/usr/lpp/ars/V10R5M0/bin/xml/samples/ARSIVPadd.ebcdic.xml` file during "IVP task: Loading the IVP definitions" on page 134.

Output file that checks account statements line data report
`/usr/lpp/ars/V10R5M0/samples/ARSIVPR1.out`

Loan delinquency line data report
`/usr/lpp/ars/V10R5M0/samples/ARSIVPR2.out`

TIFF report
`/usr/lpp/ars/V10R5M0/samples/ARSIVPR4.ind` and `/usr/lpp/ars/V10R5M0/samples/ARSIVPR4.out`

Chapter 42. IVP requirements

To run the IVPs, you must have the following software installed:

- z/OS Version 1.13 or later

Chapter 43. IVP tasks

IVP task: Installing and validating the Content Manager OnDemand administrative client and Content Manager OnDemand client

Procedure

to install and validate the Content Manager OnDemand administrative client and Content Manager OnDemand client, do the following:

1. Install the Content Manager OnDemand administrative client and Content Manager OnDemand client. For installation instructions, see *IBM Content Manager OnDemand: Client Installation Guide* .
2. Start the Content Manager OnDemand client.
3. On the **Logon to Server** panel, select **Update Servers**.
   a) In the **Server** section, enter a name for the server.
   b) In the **Host Name** section, enter an IP address or host name.
   c) Click **TCP/IP** for the Protocol.
   d) Do not select **Unified Logon**.
   e) Click **Add** to add this server to the list of servers.
   f) Click **Close** to return to the **Logon** panel
4. Make sure that the name of the server that was added is visible in the **Server** section in the **Logon** panel. If not, position the cursor to the **Server** section, click to display the list of servers, and select the server added from the list.
5. In the fields provided, enter a Content Manager OnDemand user ID and password. Use the product supplied Content Manager OnDemand user ID ADMIN for the initial log on. The first ADMIN log on instance does not require a password, but prompts to set and verify the initial password. This password is used in the remainder of the validation process.
6. Press Enter or select **OK** to initiate the logon.
7. From the list of folders, highlight **System Log** , and select **Open**.
8. The search screen is displayed with today's date in the **Time Stamp** field. Select **Search** to display the default hit list. If any message in the list has Yes in the View column, you might highlight that message and click **View All Selected** to view the detail. Press F3 to exit the view.
9. Select **Close Folder** to return to the folder list.
10. To exit the Content Manager OnDemand client, select **Cancel** to close the list of folders, select **File** from the window tabs and select **Exit OnDemand** from the list of options.

Results

If the messages stored in the system log can be viewed, the installation of Content Manager OnDemand was successful.

If the client program does not start, check the drive, path name, and program name values that are used to start the program, then try the command again.

If the client program issues a message indicating a problem, follow the instructions in the message window. See *IBM Content Manager OnDemand: Messages and Codes*. If the problem persists, contact the IBM support center for help with resolving the problem.

IVP task: Loading the IVP definitions

Procedure

To load the IVP definitions, do the following steps:

1. Logon to TSO and activate an OMVS session.
2. Enter the following command on the OMVS command line, changing *password* to the ADMIN password: `/usr/lpp/ars/V10R5M0/bin/arsxml add -h ARCHIVE -u ADMIN -p password -v -d /usr/lpp/ars/V10R5M0/bin/xml/samples -i ARSIVPadd.ebcdic.xml`

   If an alternate mount point is being used, substitute the mount point for `/usr/lpp/ars/V10R5M0`.
3. While the command is processed, the message `Running` appears to right of the OMVS command line. If the message changes to `Input`, press the function key that is associated with Refresh.

Results

The command should complete without errors.

IVP task: Capturing report ARSIVPR1 – Checking Account Statements

Procedure

To capture report ARSIVPR1 – checking account statements, do the following steps:

1. Copy the JCL member ARSIVPJ1 from the SARSINST data set to a local data set.
2. Add a JOB card.
3. Change the SET statements ARS and DB2 to match the naming conventions of the installed Content Manager OnDemand system.
4. Change <password> in the EXEC PARM to match the ADMIN password.
5. If Content Manager OnDemand is installed in an alternate mount point, change the PATH for OBJINPT to match the mount point.
6. Submit the ARSIVPJ1 job.

   The ARSIVPJ1 job completes with the following messages:

```
2014-08-27 14:51:36.924746: ARS1144I OnDemand Load Id =
>7174-1-0-2FAA-19921218000000-19921218000000-7178<
2014-08-27 14:51:36.941299: ARS1146I Loaded 33 rows into the database
2014-08-27 14:51:36.946861: ARS1405I The data that was loaded was not a part of a distribution in ODF.
2014-08-27 14:51:36.946918: ARS1175I Document compression type used - None.  Bytes Stored = >1165113<
Rows = >33<
2014-08-27 14:51:36.946942: ARS4310I Loading completed
2014-08-27 14:51:36.948699: ARS4317I Processing successful for file >/ars/tmp//tempname
(DD:OBJINPT-...PATH=.SPECIFIED...)<
```

   Some variation of the messages is acceptable.
7. Start the Content Manager OnDemand client, log on and select **ARSIVPR1** from the list of folders.
8. Use the default search options to select **Search**.
9. Select and view one or more documents from the hit list.

   Press F3 to exit from each view.
10. After reviewing several documents, select **Close folder** to return to the Folder List.
11. Select **Cancel** to exit the list, then select **File** > **Exit OnDemand** to end the client.

IVP task: Capturing report ARSIVPR3 – ACIF sample baxter bay bank

Procedure

To capture report ARSIVPR3 – ACIF sample baxter bay bank, do the following steps:

1. Copy the JCL member ARSIVPJ3 from the SARSINST data set to a local data set.
2. Add a JOB card.
3. Change the SET statements ARS, DB2 and ACIF to match local naming conventions.
4. Change <password> in the EXEC PARM to match the ADMIN password.
5. If ACIF uses a data set naming convention other then ACIF.V4R3M0, change the application ARSIVPR3 indexing parameters.
   a) Start the Content Manager OnDemand administrative client.
   b) Right-click the server that is used for the IVP.
   c) Click **Logon**.
   d) Enter user ID ADMIN and the ADMIN password in the **Logon** panel and click **OK**.
   e) In the left pane, right-click **Applications**.
   f) From the list of applications in the right frame, click application ARSIVPR3.
   g) Select **Update**.
   h) Click **Indexer Information**.
   i) In the **Parameter Source** area, select **Keyboard** and then select **Modify**.
   j) Change the ACIF.V4R5M0.SAPKULIB value for USERLIB= to match local naming conventions.
   k) Click the **X** in the **Edit index parameters** window.
   l) Select **Yes** in response to the `Save changes to Indexer parameters?` message.
   m) Select **OK** to complete the update.
6. Submit the ARSIVPJ3 job.
   The ARSIVPJ3 job completes with the following messages:

```
2014-08-26 10:01:54.116159: ARS1144I OnDemand Load Id =
>34796-0-0-1FAA-20140826170154-20140826170154-34797<
2014-08-26 10:01:55.261672: ARS1146I Loaded 8 rows into the database
2014-08-26 10:01:55.266556: ARS1405I The data that was loaded was not a part of a distribution in
ODF.
2014-08-26 10:01:55.266595: ARS1175I Document compression type used - OD77.  Bytes Stored = >18204< Rows
= >8<
2014-08-26 10:01:55.266614: ARS4310I Loading completed
2014-08-26 10:01:55.269276: ARS4317I Processing successful for file >/ars/tmp//tempname (DD:OBJINPT-
APK.V4R5M0.SAPKSAM3(APKBANK))<
```

   The message you see might vary somewhat from this example.
7. Start the Content Manager OnDemand client, log on and select ARSIVPR3 from the list of folders.
8. Use the default search options to select **Search**.
9. Select and view one or more documents from the hit list.
   Press F3 to exit from each view.
10. After reviewing several documents, select **Close folder** to return to the folder list.
11. Select **Cancel** to exit the list, then select **File** > **Exit OnDemand** to close the client.

IVP task: Capturing report ARSIVPR4 – TIFF sample

Procedure

To capture report ARSIVPR4 – TIFF sample, do the following steps:

1. Copy the JCL member ARSIVPJ4 from the SARSINST data set to a local data set.
2. Add a JOB card.
3. Change the SET statements ARS and DB2 to match the naming conventions of the installed Content Manager OnDemand system.
4. Change <password> in the EXEC PARM to match the ADMIN password.
5. Copy files /usr/lpp/ars/V10R5M0/samples/ARSIVPR4.ind and /usr/lpp/ars/V10R5M0/samples/ARSIVPR4.out to the /tmp directory.

   If /tmp directory is not used, edit ARSIVPR4.ind and change the statement GROUP_FILENAME:/tmp/ARSIVPR4.out to match the directory that is chosen.
6. Submit the ARSIVPJ4 job.

   The ARSIVPJ4 job completes with the following messages:

```
2014-08-26 10:01:54.116159: ARS1144I OnDemand Load Id =
>34796-0-0-1FAA-20140826170154-20140826170154-34797<
2014-08-26 10:01:55.261672: ARS1146I Loaded 8 rows into the database
2014-08-26 10:01:55.266556: ARS1405I The data that was loaded was not a part of a distribution in
ODF.
2014-08-26 10:01:55.266595: ARS1175I Document compression type used - OD77.  Bytes Stored = >18204< Rows
= >8<
2014-08-26 10:01:55.266614: ARS4310I Loading completed
2014-08-26 10:01:55.269276: ARS4317I Processing successful for file >/ars/ARCHIVE/tmp//tempname
(DD:OBJINPT-APK.V4R5M0.SAPKSAM3(APKBANK))<
```

   Some variation of the messages is acceptable.
7. Start the Content Manager OnDemand client, log on and select ARSIVPR4 from the list of folders.
8. Use the default search options to select **Search**.
9. Select and view one or more documents from the hit list.

   Press F3 to exit from each view.
10. After reviewing several documents, select **Close folder** to return to the folder list.
11. Select **Cancel** to exit the list, then select **File** > **Exit OnDemand** to close the client.

IVP task: Cleaning up

About this task

The Content Manager OnDemand Installation Verification Process (IVP) is now complete.

Procedure

To remove captured reports, definitions and modified hfs files, do the following steps:

1. Delete IVP definitions and captured reports:
   a) Start the Content Manager OnDemand administrative client.
   b) Right-click the server from the list, and select **Logon**. Enter a valid user ID and password. Press Enter.
   c) In the left pane, select **Folders** to see a list of folders.
   d) With the Ctrl key pressed, select (and highlight) ARSIVPR1, ARSIVPR2, ARSIVPR3 and ARSIVPR4 from the list.
   e) Select **File** > **Delete**.
   f) In the **Delete Folders** window, select **OK**.
   g) In the **Confirm Delete Folders** window, select **Yes to All**.
   h) In the left pane, select **Application Groups** to see a list of application groups.
   i) With the Ctrl key pressed, select and highlight ARSIVPR1, ARSIVPR2, ARSIVPR3 and ARSIVPR4 from the list.
   j) Select **File** > **Delete**.

k) In the **Password Validation** window, enter the ADMIN password and select **OK**.

l) In the **Delete Application Groups** window, select **OK**.

m) In the **Confirm Delete Application Groups** window, select **Yes to All**.

2. Delete the following IVP-modified HFS files: `/tmp/ARSIVPR4.ind`, `/tmp/ARSIVPR4.out`

Part 9. Further customization

Chapter 44. Defining multiple instances

Instances

A Content Manager OnDemand instance is a logical server environment made up of a database, a library server, and one or more object servers. An instance is defined in the `ars.ini` file by naming the instance, identifying the name of the database used by the instance, and identifying the library server on which the database will be maintained. When you configure an object server, you identify its library server in the `ars.cfg` file on the object server. An instance has its own cache storage file systems. The cache storage file systems are defined in the `ars.cache` file on each object server. All of the servers that belong to an instance run in one and only one code page.

You can run multiple instances on the same system, with each instance configured differently:

- To have separate test and production environments
- To have databases using different code pages

Important: When you work with more than one instance, you must identify the instance name when you run Content Manager OnDemand programs (such as ARSDB, ARSLOAD, and ARSSOCKD) and database commands (such as connecting to the database).

Each instance has different security from other instances on the same machine. You must define users and groups to each instance and set application group and folder permissions for users of each instance. Each instance has its own system log.

Each additional instance requires additional system resources, such as virtual storage and disk space, and more administration. For example, you must make sure that the `ars.ini` file is consistent and correct on all servers that are part of a logical Content Manager OnDemand system.

Configuring multiple instances

About this task

You can run multiple instances on a single system. Each instance is independent from the other instances and there no data is shared between instances. Because the instances are independent of each other, you can create several instances on a single computer or logical partition (LPAR) and run the instances concurrently. The ability to run multiple instances provides you with flexibility when you set up Content Manager OnDemand environments that separate production from development or set up instances that use different code pages.

If you plan to run more than one instance on a single system, do the following:

- You must create a copy of the `ars.cfg` file for each instance.
- IBM recommends that you maintain separate cache storage file systems for each instance.

The `ars.ini` file is the only file that is common to all instances of Content Manager OnDemand on the system. The `ars.ini` file must contain one section for each instance. Each section should identify an `ars.cfg` file (instance configuration file) and an `ars.cache` file (cache storage configuration file) that is unique to the instance.

The information in this section describes how to configure Content Manager OnDemand instances:

- Adding an instance
- Modifying the server configuration
- Defining cache storage file systems

Adding an instance

About this task

The `ars.ini` file contains information about Content Manager OnDemand instances. When you install the Content Manager OnDemand software, the `ars.ini` file contains information about the default instance named ARCHIVE. The easiest way to add a new instance to the file is to copy the set of lines that comprise the default instance definition and make changes to the copy.

When you install the Content Manager OnDemand software on the server, the `ars.ini` file gets copied to the Content Manager OnDemand configuration file directory. To define a new instance, log on as the super user or as any user set up with appropriate permissions. Make a backup copy of the `ars.ini` file. Then edit the `ars.ini` file with a standard text editor.

The information in the `ars.ini` file is organized in sections with a header line that identifies each section. The header line is identified by the brackets that delimit the beginning and end of the line.

The first section in the file contains information about the default instance. The following shows the default instance:

```
[@SRV@_ARCHIVE]
HOST=
PROTOCOL=2
PORT=0
SRVR_INSTANCE=archive
SRVR_INSTANCE_OWNER=ARSSERVR
SRVR_OD_CFG=/usr/lpp/ars/V10R5M0/config/ars.cfg
SRVR_SM_CFG=/usr/lpp/ars/V10R5M0/config/ars.cache
```

- The header line contains a string that identifies the name of the instance. By default, the first or only instance is named ARCHIVE.
- The **HOST** parameter identifies the host name alias, IP address, or fully qualified host name of the library server. For information about how Content Manager OnDemand uses the **HOST** parameter, see “Working with instances” on page 145.
- The **PROTOCOL** parameter identifies the communications protocol used to communicate between clients and the server (the number 2 identifies TCP/IP).
- The **PORT** parameter identifies the TCP/IP port number that Content Manager OnDemand monitors for client requests (the number 0 means that the server monitors port number 1445). If you use a port number other than 1445 on the library server, enter that number instead of 0.
- The **SRVR_INSTANCE** parameter identifies the name of the database that is used by the instance. IBM recommends that you use the same name for the database and the instance.
- The **SRVR_INSTANCE_OWNER** parameter identifies the name of the instance owner. This value should always be set to the super user or any other user that has the appropriate permissions.
- The **SRVR_OD_CFG** parameter identifies the configuration file that is used by the instance. Each instance must use a separate configuration file.
- The **SRVR_SM_CFG** parameter identifies the file that specifies the cache storage file systems that are used by the instance. Each instance must use a separate file to specify its cache storage file systems.

When you add an instance to the `ars.ini` file, remember that each instance must use a unique name. For example, to add an instance for testing new applications, you might add an instance named TEST. *When you work with more than one instance, you must identify the instance name when you run Content Manager OnDemand programs (such as ARSDB, ARSLOAD, and ARSSOCKD) and database commands (such as connecting to the database).* The following shows an example of a second instance of Content Manager OnDemand in the `ars.ini` file:

```
[@SRV@_TEST]
HOST=rhone
PROTOCOL=2
PORT=1444
SRVR_INSTANCE=TEST
SRVR_INSTANCE_OWNER=ARSSERVR
```

```
SRVR_OD_CFG=/usr/lpp/ars/V10R5M0/config/ars.test.cfg
SRVR_SM_CFG=/usr/lpp/ars/V10R5M0/config/ars.test.cache
```

The name of the instance is TEST. The HOST statement identifies the host name alias of the library server. (When you add another instance to the ARS.INI file, you must set the HOST statement of the default instance to the host name alias of the library server.) The instance uses TCP/IP to communicate over port number 1444. (Each instance that runs on the same system must use a different port number.) The name of the database that is used by the instance is TEST. The name of the server configuration file that is used by the instance is `ars.test.cfg`. The name of the cache storage configuration file that is used by the instance is `ars.test.cache`.

Important: For a distributed library/object server system, you must make sure that the `ars.ini` file is consistent and correct on all servers that are part of the Content Manager OnDemand instance.

Modifying the ars.cfg server configuration file

About this task

This section provides an overview of the changes you may need to make to the `ars.cfg` server configuration file. For details about modifying the server configuration file, see Chapter 13, “Verify the ARS.CFG file,” on page 39.

The server configuration file contains information about the instance, such as identifying the object servers that belong to the instance, the language settings for the instance, and information used by database, storage, and print manager programs.

When you start an instance, Content Manager OnDemand processes the server configuration file for information used to initialize the instance. The **SRVR_OD_CFG** parameter in the `ars.ini` file identifies the server configuration file that Content Manager OnDemand uses to initialize the instance. (Each instance defined in the `ars.ini` file has its own **SRVR_OD_CFG** parameter, which *must* identify a unique server configuration file.)

By default, the default instance uses a server configuration file named `ars.cfg`. If you plan to run more than one instance, make a unique copy of the `ars.cfg` file and the `ars.cache` file for each additional instance. When you install Content Manager OnDemand software on the server, the `ars.cfg` file is copied to the Content Manager OnDemand configuration file directory. To define a new instance, log on as the super user or as any user set up with appropriate permissions. Copy the `ars.cfg` file. For example, to configure an instance named TEST, copy the `ars.cfg` file to `ars.test.cfg`. Then edit the new server configuration file with a standard text editor.

Settings that you should verify

About this task

When you add an instance to the `ars.ini` file, verify the following information in the `ars.cfg` server configuration file that is used by the instance:

- License information
- Language settings
- Library and object servers
- Database servers
- DB2 parameters

Settings that you must change

About this task

When you add an instance to the `ars.ini` file, IBM recommends that you modify database name in the server configuration file used by the instance to match the name of the instance.

Defining cache storage file systems

About this task

This section provides an overview of what you need to do to define cache storage file systems for an instance. For details about defining cache storage file systems, see Chapter 14, "Modify the ARS.CACHE file," on page 53.

Each instance must use different cache storage file systems. Before using cache storage file systems, you need to configure physical and logical storage, set the correct file system permissions, modify the `ars.ini` file to identify the cache storage configuration file that is used by the instance, and create a cache storage configuration file for the instance. The cache storage configuration file lists the cache storage file systems on which Content Manager OnDemand will maintain the documents that are stored into the instance.

When you start an instance, Content Manager OnDemand reads the `ars.ini` file for information used to initialize the instance. The **SRVR_SM_CFG** parameter in the `ars.ini` file identifies the cache storage configuration file that is used by the instance. (Each instance defined in the `ars.ini` file has its own **SRVR_SM_CFG** parameter, which must identify a unique cache storage configuration file.)

By default, the default instance identifies a cache storage configuration file named `ars.cache`. If you plan to run more than one instance, make a unique copy of the `ars.cache` file for each additional instance. When you install the Content Manager OnDemand software on the server, the `ars.cache` file is copied to the Content Manager OnDemand configuration file directory. To create a new cache storage configuration file, log on as the super user or as any user set up with appropriate permissions. Make a copy of the `ars.cache` file. For example, to create a cache storage configuration file for the instance named TEST, copy the `ars.cache` file to `ars.test.cache`. Then edit the cache storage configuration file with a standard text editor. Add one line to the file for each cache storage file system that you want the instance to use.

Modify the `ars.ini` file to identify the cache storage configuration file that you created for the instance. For example, if you created a cache storage configuration file named `ars.test.cache` for the instance named TEST, locate the `[@SRV@_TEST]` section in the `ars.ini` file and change the **SRVR_SM_CFG** parameter to identify the cache storage configuration file: `SRVR_SM_CFG=/usr/lpp/ars/V10R5M0/config/ars.test.cache`

Creating instances

About this task

After you configure the `ars.ini` file, the server configuration file, and the cache storage configuration file, you can create and initialize the instance:

- Run the ARSDB2 job to create a database for the instance
- Run the ARTSPAC job to create the table space that is used for the Content Manager OnDemand system tables
- Run the ARSDB program to create the Content Manager OnDemand system tables
- Run the ARSSYSCR program to initialize the system log and system migration tables

You create the Content Manager OnDemand system tables by running the ARSDB program on the library server. The ARSDB program initializes the base database tables required by Content Manager OnDemand. To run the ARSDB program, log on as the super user or as any user set up with appropriate permissions. Then run the ARSDB program from the command prompt. For example, the following shows how to create the system tables for the instance named TEST: `arsdb -I TEST -cv`, where `-I TEST` identifies the instance. The ARSDB program creates and initializes the system tables, generating the application groups, applications, and folders required to support the instance. When complete, the ARSDB program displays the `Creation…successful` message.

You can initialize the other system tables by running the ARSSYSCR program on the library server. The ARSSYSCR program initializes the database tables required to support the system log, system migration, and other Content Manager OnDemand system tables. To run the ARSSYSCR program, log on as the super user or as any user set up with appropriate permissions. Then run the ARSSYSCR program from the command prompt. For example, the following shows how to initialize the system log for the instance named TEST: `arssyscr -I TEST -l`, where `-I TEST` identifies the instance. In the example, the ARSSYSCR program creates the application groups, applications, and folders required by the system logging facility.

Manually starting and stopping instances

About this task

You must start an instance before clients can connect to databases on the instance.

To start an instance, issue an MVS START command. For example, from SDSF, enter `/S ARSSOCKT`, where ARSSOCKT is the name of the JCL procedure that was created for the new instance.

To stop an instance, issue an MVS STOP command. For example, from SDSF, issue `/P ARSSOCKT`.

Connecting to instances using Content Manager OnDemand clients

About this task

To connect to a particular instance, the client must log on to the correct library server. Use the **Update Servers** dialog box to add library servers to the client. When you add a library server, you identify the name of the library server and the port number and communications protocol used to communicate with the server. The port number that you specify in the **Update Servers** dialog box must be the same as the value that you specified for the instance in the `ars.ini` file.

Working with instances

About this task

Important: When you run a command or program against an instance, you *must* specify the name of the instance, not the host name or IP address of the library server on which the instance is running. The system locates the specified instance name in the ARS.INI file to determine the IP address, host name alias, or fully-qualified host name of the library server.

Content Manager OnDemand provides programs that you can use to work with an instance. For example, Content Manager OnDemand provides programs to create databases, start instances, load data, and maintain index data. One of the parameters that you can specify to the Content Manager OnDemand programs is the name of the instance to process. By default, if you do not specify otherwise, the name of the instance is ARCHIVE. If you name your default instance something other than ARCHIVE or you work with more than one instance, then always specify the name of the instance when you run a Content Manager OnDemand program. That way, you can be certain that you are working with the correct instance. Depending on the program that you need to run, there are different ways that you can specify the name of the instance.

The ARSDB, ARSLOAD, ARSMAINT, and ARSSYSCR programs use the **–I** *instance* parameter. Content Manager OnDemand retrieves the host name of the library server that is associated with the specified instance from the `ars.ini` file. For example:

- `arsdb -I TEST -cv`
- `arsload -I TEST -d /arsacif/test/acif1 -c /arsacif/test/acif2`
- `arsmaint -I TEST -cmsv`

- `arssyscr -I TEST -l`

Where TEST is the name of the instance. In the examples, the **HOST** parameter in the `[@SRV@_TEST]` section of the `ars.ini` file identifies the library server.

The ARSADMIN program uses the following syntax: `arsadmin load -h TEST -g Credit -i credit.ind`, where TEST is the name of the instance. In the examples, the **HOST** parameter in the `[@SRV@_TEST]` section of the `ars.ini` file identifies the library server.

The ARSSOCKD program uses the following syntax:

```
//ARSSOCKT EXEC PGM=ARSSOCKD,REGION=0M,TIME=NOLIMIT,
// PARM='TEST ARSSOCKD'
```

Where TEST is the name of the instance. In the examples, the **HOST** parameter in the `[@SRV@_TEST]` section of the `ars.ini` file identifies the library server.

The ARSDOC program provides the **-h** parameter to specify the name of the library server when you are working with the default instance. The following example show how to run the programs by specifying the name of the library server and working with the default instance: `arsdoc get -h rhone -F parmfile`, where `rhone` is the name of the library server. However, if you are running more than one instance on the same system, then you must use the **-h** parameter to specify the name of the instance instead of the name of the library server. After locating the instance name in the `ars.ini` file, Content Manager OnDemand retrieves the value of the **HOST** parameter to identify the library server. The following example shows how to run the programs by specifying the name of the instance: `arsdoc get -h TEST -F parmfile`, where TEST is the name of the instance. In the examples, Content Manager OnDemand uses the **HOST** parameter in the `[@SRV@_TEST]` section of the `ars.ini` file to identify the library server.

Planning for capacity

About this task

To plan the capacity requirements for storing documents in your environment, you must consider the following factors:

- Temporary spaces for loading reports and printing documents
- Cache (short-term) storage for documents
- Archive (long-term) storage for documents
- Application group data tables
- Backup and recovery procedures

For details, examples, and worksheets to help you estimate the amount of storage needed to support your environment, see the *IBM Content Manager OnDemand for z/OS: Introduction and Planning Guide*.

Physically separating program directories, staging areas, temporary spaces, cache storage, and archive storage on to different disk drives and controllers will improve performance and the time that it takes to recover from problems.

Chapter 45. Server print

Overview

The Content Manager OnDemand server print function is the highest performance print option that is available to end-users of the system. The server print function is designed to allow many documents to be selected for reprint from the document list, without retrieving the documents to the user's PC. The Content Manager OnDemand server print function supports printing by using the *Direct Server* printing method, to submit a print job to a JES output class or by using the *Infoprint Server* printing method, to submit a print job to an IBM Infoprint Server for z/OS print queue.

When a user invokes the Content Manager OnDemand server print function, the client sends a print request to the Content Manager OnDemand server. The server sorts the documents by storage volume before retrieving them. After the documents are retrieved, a print job is submitted to a JES output class or to an Infoprint Server print queue, depending on the printing method that was specified for the Content Manager OnDemand server printer. When a user prints an AFP document, the resource group that was archived at the time that the document was loaded into the system is automatically retrieved and placed in the print data stream, to make sure that the document prints with the correct resources.

A Content Manager OnDemand server printer is an interface between the user and a server print device. Content Manager OnDemand supports printing to the JES spool or to a print queue supported by the Infoprint Server **lp** command. A server print device can be physically connected to the library server or it can be attached to some other system on the network. Printing to network printers requires Infoprint Server. The printing method is chosen when a server printer is added to Content Manager OnDemand, by specifying the string `JES:` (for the direct server printing method) or the name of an Infoprint Server print queue in the **Server Queue Name** field.

To configure the system to support the server print function, you must do the following tasks:

- Define a server printer to Content Manager OnDemand with the administrative client. To use the direct printing method, the server printer must identify a JES destination by specifying the value `JES:` in the **Server Queue Name** field. Any other value will be interpreted as an Infoprint Server queue name, for the Infoprint server printing method.
- Define the printer options to Content Manager OnDemand with the administrative client. The valid printer options are those that can be specified with the **OUTDES** command. The printer options are defined in the Content Manager OnDemand application on the **Print Options** page. For a list of the valid options for the **OUTDES** command, see *z/OS MVS Programming: Authorized Assembler Services Guide*.

Configuring direct server printing

About this task

Note: To print AFP documents with the direct server printing method requires that IBM Print Services Facility Version 4.1 or later be installed and operational.

The direct server printing method allows users to send Content Manager OnDemand documents to a JES SYSOUT class. This document assumes that you have already established the JES SYSOUT class that you want to use to support printing from Content Manager OnDemand and that you have configured the operating system and the print devices. This section describes the configuration that is required on the Content Manager OnDemand library server to use the direct server printing method.

Procedure

To configure the Content Manager OnDemand library server to use the direct server printing method, do the following steps:

1. Verify the HFS data set that will be used to temporarily store server print data.

   The default location is ARS.TMP.HFS with a mount point of `/ars/tmp`. For more information, see Part 2, “Installing the server software,” on page 17.
2. Verify the location that will be used to temporarily store server print data.

   The location is specified on the **ARS_PRINT_PATH** parameter in the `ARS.CFG` file. For more information, see Chapter 13, “Verify the ARS.CFG file,” on page 39.
3. Add a server printer to Content Manager OnDemand by using the Content Manager OnDemand administrative client.
4. For each Content Manager OnDemand application that will be using the direct server printing method, specify printer options for the application by using the Content Manager OnDemand administrative client.

   If an application does not contain any printer options, the default print option is JES SYSOUT CLASS=A.

Adding a server printer

About this task

A Content Manager OnDemand server printer that contains a value of `JES:` in the **Server Queue Name** field will use the direct server printing method. The value that you specify in the **Name** field can be selected from the **Default Server Printer** field in the Content Manager OnDemand application and is available to authorized users from the Windows client.

Procedure

To add a server printer to Content Manager OnDemand by using the administrative client, do the following steps:

1. Start the Content Manager OnDemand administrative client.
2. Log on to the server.
3. Point to **Printers** and click the right mouse button. Then select **New Printer** from the menu to open the **Add a Printer** dialog box.
4. On the **General** page, complete the **Name** field.

   For more information, see the online help.
5. Optional: Specify a description in the **Description** field.
6. Type the value **JES:** in the **Server Queue Name** field.

   This value is case sensitive and must be specified in uppercase.
7. Verify that the **Print Type** field contains the value Printer.
8. On the **Permissions** page, specify the users and groups that should be allowed to use the server printer. To allow all users and groups that are defined on the library server to use the server printer, select the **Access by all Users / Groups** option.
9. Click **OK** to add the server printer to Content Manager OnDemand.

Specifying printer options

About this task

The Content Manager OnDemand application contains the printer options that Content Manager OnDemand sends to JES along with a copy of the document to be printed. The printer options are the same as those that can be specified with the **OUTDES** command.

Procedure

To specify printer options for a Content Manager OnDemand application by using the administrative client, do the following steps:

1. Start the Content Manager OnDemand administrative client.
2. Log on to the server.
3. Point to **Applications**.
   - If you are adding an application, click the right mouse button. Then select **New Application** from the menu to open the **Add an Application** dialog box.
   - If you are modifying an application, point to the application in the list and click the right mouse button. Then select **Update** from the menu to open the **Update an Application** dialog box.
4. Move to the **Print Options** page.
5. Optional: Specify a printer in the **Default Server Printer** field.

   The name of the default server printer will appear in the **server printer name** field when a user prints a document from the application by using the server print command from the Windows client. For example, you can select the server printer that you added in "Adding a server printer" on page 148. For more information, see the online help.
6. Under **Server Options**, complete the **Print Parameters** field by specifying options that can be specified with the **OUTDES** command.

   For a list of the valid options for the **OUTDES** command, see *z/OS MVS Programming: Authorized Assembler Services Guide*.

   Note: If an application does not contain any Print Parameters, then the default print option is JES SYSOUT CLASS=A.
7. When you are finished adding or modifying the application, click **OK** to save the information that you specified.

Results

In addition to the **Printer** options in the application, you can specify additional parameters when you make a print request. You can save these options for future print requests. These parameters include **`class`**, **`dest`**, **node**, **`writer`**, **`pagedef`**, **`formdef`**, **`forms`**, and **`routing`**.

The **node** and **`dest`** are mapped to the **`DEST OUTDES`** parameter. If both **`dest`** and **node** are provided, the **`dest`** parameter is used for the **`DEST OUTDES`** parameter.

If a user specified parameter is provided that matches a parameter that is found in the application print parameters, the user specified parameter takes precedence.

Routing is ignored because it does not directly map to an **OUTDES** parameter.

Configuring Infoprint server printing

About this task

Note: To print documents with the Infoprint server printing method requires that IBM z/OS V1R13.0 Infoprint Server or later be installed and operational. For information on installing, configuring, administering, and using Infoprint Server and to obtain the Infoprint Server manuals, see IBM Infoprint Server and print transforms for z/OS.

By default, the server creates a program to send the file to the JES SPOOL. An alternative is provided that allows a batch job to be submitted that invokes that program. To submit a batch job, do the following steps:

1. Create the `arsprtjcl` file as outlined below. You can use different JCL for direct server printing by creating a different JCL file and using the **ARSMVS_JESPRTJCL** parameter in the `ars.cfg` file to specify a full path name to the alternate JCL.
2. Modify the `ars.cfg` file to specify `ARSMVS_JESPRT_SUBMIT=1`.

You can configure Content Manager OnDemand to use Infoprint server to print documents. In addition to installing the Infoprint server, you must configure Content Manager OnDemand to invoke the Infoprint server **lp** command.

Procedure

Follow these steps:

1. Copy the `/usr/lpp/ars/V10R5M0/samples/arsprtjcl1` file to `/usr/lpp/ars/V10R5M0/config/arsprtjcl`. Ensure that you change the file name from `arsprtjcl1` to `arsprtjcl`.
2. Customize the `/usr/lpp/ars/V10R5M0/config/arsprtjcl` file to meet your installation's requirements. The `//STDIN DD PATHDISP=(DELETE,KEEP)` must be the last JCL statement in the file, and must have a trailing comma. Content Manager OnDemand appends a `// PATH=` parameter after the last line in the file.
3. Copy the `/usr/lpp/ars/V10R5M0/samples/arsprt` file to `/usr/lpp/ars/V10R5M0/bin/arsprt`. You can also customize the script for installation specific processing. If the first line starts with `//` and 9 blanks, the server generates a job name based on the user who makes the print request. If the user is less than 8 characters long, then the numbers from 0 to 9 are appended to the user. If the user is 8 characters or more, just the first 8 characters are used. If the first line does not start with `//` and 9 blanks, the line is not modified.
4. Grant execute permissions to the `/usr/lpp/ars/V10R5M0/bin/arsprt` file. For example: `chmod 755 /usr/lpp/ars/V10R5M0/bin/arsprt`

Results

When Content Manager OnDemand receives a server print request that does not have a queue name of JES:, it submits the JCL in `/usr/lpp/ars/V10R5M0/config/arsprtjcl`. Content Manager OnDemand appends the following lines to that JCL:

```
//       PATH='file-containing-printcmd.shcmd'
```

Where *file-containing-printcmd* is generated based on the setting of **ARS_PRINT_PATH** and the print request. Multiple lines might be appended if the constructed file name does not fit on a single JCL statement.

The *file-containing-printcmd*.shcmd path contains the actual shell command to be invoked. The first line contains:

```
#!/bin/sh
```

The second line contains the following information on one line:

```
/usr/lpp/ars/V10R5M0/bin/arsprt
 "printer_queue_name"
 "copies"
 "userid"
 "applgroup"
 "appl"
 "appl_print_parameters"
 "file_to_print"
 "type"
 "recfm"
 "lrecl"
 "class"
 "dest"
 "node"
 "writer"
 "pagedef"
 "formdef"
 "forms"
```

```
  "routing"
  "banner"
```

where

/usr/lpp/ars/V10R5M0/bin/arsprt
Script command to run.

printer_queue_name
The printer queue name that is associated with the printer.

copies
Number of copies that you requested.

userid
The user who is making the print request.

applgroup
Application group for the document that is printed.

appl
Application for the document that is printed.

appl_print_parameters
Print parameters from the application.

file_to_print
File containing the document to print.

type
Document type from the application:

```
A - AFP
B - BMP
E - Email
G - GIF
J - JFIF (JPEG File Image Format)
K - Global DJDE
L - Line
M - Metacode
N - None
O - OD Defined
P - PCX
R - PDF
Q - PNG
S - SCS
T - TIFF
U - User Defined
X - SCS Extended
? - Unknown
```

recfm
RECFM from the Application for line and Global DJDE.

```
F - fixed
V - Variable
S - Stream
```

lrecl
LRECL for fixed data. For Stream data, this is the delimiter characters.

class
Class that is specified by the user.

dest
The destination printer that is specified by the user.

node
Node that is specified by the user.

writer
Writer that is specified by the user.

pagedef
Pagedef that is specified by the user.

formdef
Formdef that is specified by the user.

forms
Forms that is specified by the user.

routing
Routine that is specified by the user.

banner
A banner was requested by the user: Y for Yes, N for No.

For example:

```
/usr/lpp/ars/V10R5M0/bin/arsprt "TESTQNAM" "6" "ARSUSER" "RTSLOD05"
"RTSLOD05" "" "/tmp/ARS.67109357.18861C000000133C.2421.0"
"L" "" "" "" "" "" "" "" "" "" "" "N"
```

These lines are on one line in the `.shcmd` file.

Parameters for which no value is supplied are indicated by `""`.

Defining an Infoprint server to Content Manager OnDemand

About this task

A Content Manager OnDemand server printer that does not contain a value of JES: in the **Server Printer Name** field will use the Infoprint server printing method. This method will submit the JCL in the `/usr/lpp/ars/V10R5M0/config/arsprtjcl` file to invoke the `/usr/lpp/ars/V10R5M0/bin/arsprt` script file, which invokes the Infoprint **lp** command.

The value that you specify in the **Server Printer Name** field can be selected from the **Default Server Printer** field in the Content Manager OnDemand application and is available to authorized users from the Windows client.

Procedure

To add an Infoprint server printer to Content Manager OnDemand by using the administrative client, do the following steps:

1. Start the Content Manager OnDemand administrative client.
2. Log on to the server.
3. Point to **Printers** and click the right mouse button. Then select **New Printer** from the menu to open the **Add a Printer** dialog box.
4. On the **General Page**, complete the **Name** field.

   See the online help if you have any questions.
5. Optional: Specify a description in the **Description** field.
6. Type the name of a valid Infoprint server queue name in the **Server Queue Name** field. This value is case sensitive and must match the name the server queue name defined in Infoprint.
7. Verify that the **Print Type** field contains the value Printer.
8. On the **Permissions** page, specify the users and groups that should be allowed to use the server printer.

   To allow all users and groups that are defined on the library server to use the Infoprint server, select the **Access by all Users/Groups** option.
9. Click **OK** to add the server printer.

Specifying printer options

About this task

The Content Manager OnDemand application contains the printer options that Content Manager OnDemand specifies to Infoprint along with a copy of the document to be printed. The printer options are the same as those that can be specified with the Infoprint **lp** command.

Procedure

To specify printer options for an Content Manager OnDemand application by using the administrative client, do the following steps:

1. Start the Content Manager OnDemand administrative client.
2. Log on to the server.
3. Point to **Applications**.
   - If you are adding an application, click the right mouse button. Then select **New Application** from the menu to open the **Add an Application** dialog box.
   - If you are modifying an application, point the application in the list and click the right mouse button. Then select **Update** from the menu to open the **Update an Application** dialog box.
4. Move to the **Print Options** page.
5. Optional: Specify a default server printer in the **Default Server Printer** field.

   The name of the default server printer will appear in the server printer name field when a user prints a document from the application by using the server print command from the Windows client. For more information, see the online help.
6. Under **Server Options**, complete the **Print Parameters** field by specifying options that can be specified with the Infoprint **lp** command.

   For a list of the valid options for the **lp** command, see *z/OS Infoprint Server User's Guide*.

   Note: The Infoprint **lp** command is not the same command as the **lp** command that is delivered with UNIX System Services. The UNIX System Services **lp** command will not work with Content Manager OnDemand.
7. When you are finished adding or modifying the application, click **OK** to save the information that you specified.

Results

In addition to the Printer options in the application, you can specify additional parameters when you make a print request. You can save these options for future print requests. These parameters include **class**, **dest**, **node**, **writer**, **pagedef**, **formdef**, **forms**, and **routing**.

Chapter 46. Server tasks

This section provides information about the Content Manager OnDemand programs that you may want to configure to start automatically or run on a regular schedule.

Starting the server program

About this task

Note: The Content Manager OnDemand server program (ARSSOCKD) must be running, otherwise the data loading program (ARSLOAD) and the maintenance programs (such as ARSMAINT) will fail.

The Content Manager OnDemand server program is started by issuing the MVS **START** command to run the appropriate procedure. A sample of this procedure is provided in the SARSINST library and must be customized to run in your installation. For more information, see Chapter 17, “Modify the ARSSOCKD procedure,” on page 61.

Automating the data loading process

About this task

Note: If you do not plan to configure your system to automatically load data, then you can skip this section.

This section provides information to help you configure the Content Manager OnDemand data loading program (ARSLOAD) to run whenever the server is active. There are two parts to automating this process:

- Specify the user ID and password of a Content Manager OnDemand user that has administrative rights for the application groups that you plan to load data into.
- Configure the ARSLOAD procedure to automatically start whenever the server is restarted and to run whenever the server is active.

Specifying the Content Manager OnDemand user ID and password

About this task

To load a report into the system with the ARSLOAD program, you must specify a user ID with administrator authority for the application group into which you want to load the data. You must also specify a password for the user ID. There are several ways that you can specify the user ID and password:

- Assign a user ID to the ARSLOAD started task through the RACF STARTED class and use the ARS.PTGN exit to provide a PassTicket. For information about the RACF STARTED class, see *z/OS Security Server RACF System Programmer's Guide*. For information about creating PassTickets, see Chapter 51, “Unified login (ARS.PTGN) exit,” on page 201.

Configuring the ARSLOAD procedure

About this task

The ARSLOAD program is the main Content Manager OnDemand data loading and indexing program. You can configure the ARSLOAD program to monitor output parameters (**CLASS**, **DEST**, **WRITER**, **FORM**) for report data. If the data needs to be indexed, then the ARSLOAD program calls the indexing program that is specified in the Content Manager OnDemand application. The ARSLOAD program then works with the

database manager to load the index data into the database and works with the storage manager to load the report data and resources on to storage volumes.

The ARSLOAD procedure is initiated by issuing the MVS **START** command. The procedure to run the ARSLOAD program is provided in the SARSINST library and must be customized to run in your installation. For more information, see Chapter 18, “Modify the ARSLOAD procedure,” on page 63.

Note:

1. The Content Manager OnDemand server program (ARSSOCKD) must be running, otherwise the ARSLOAD program will fail.
2. The data loading process should run when no other programs are updating the database or need exclusive access to the database. For example, do not run the ARSLOAD program at the same time that you run database statistics or maintain the database. (This means do not run the ARSLOAD program while you are running the ARSDB program with the **-m** parameter.) There might be other processes that could conflict with the data loading process. For example, you cannot load data while an offline backup of the database is in process.
3. The ARSLOAD program uses the FORMS part of the spool file name to determine the application group to load. You can use the **-G** parameter to specify the application group to load. If the application group contains more than one application, then you must identify the application to load. Otherwise, the load will fail. You can use the **-A** parameter to specify the application. For more information about the ARSLOAD program and the **-G** and **-A** parameters, see the *IBM Content Manager OnDemand for z/OS: Administration Guide* .
4. To load a report with the ARSLOAD program, you must specify the user ID and password of a Content Manager OnDemand user that has administrator authority for the application group into which the data will be loaded. For more information, see “Specifying the Content Manager OnDemand user ID and password” on page 155.
5. After indexing the data, the ARSLOAD program deletes the input files, unless you specify otherwise with the **-n** parameter. Any output or error messages that are generated by the ARSLOAD program are written to the system log. Using the end-user client, you can open the System Log folder and retrieve any messages that were generated by the ARSLOAD program.

Maintaining database tables

About this task

You must use utilities that are provided with DB2 to reorganize and optimize the database tables. You must use DB2 or operating system facilities to schedule these tasks. For information about DB2 table maintenance, see the appropriate DB2 reference material . Also see the database administrator for your organization.

Note: Unless you disable RUNSTATS, RUNSTATS will be invoked with the DSNUTILS stored procedure. You need to have a WLM address space established. For more information on the WLM requirements when using DSNUTILS, see the appropriate DB2 reference manual.

Chapter 47. Migrated indexes

Setup steps for retrieving documents for migrated indexes

To install the index migration routines needed for retrieving documents for migrated indexes, you must copy members to the user libraries, customize the DB2 package and plan DDL, and prepare the migration environment.

For more details on how to create user libraries, see “Creating the Content Manager OnDemand user libraries” on page 235 .

Task: Copy members to user libraries

About this task

Use the user libraries that you created during the initial migration process.

Copy members as follows:

- From SARSINST to *hlq*.USERPARM
  - ARSSSIDP, renaming member to ARSSSID
- From SARSINST to *hlq*.USERDB2
  - ARSGIAF2
  - ARSKDOCG
  - ARSKRUN
  - ARSLDOCG
  - ARSLRUN
  - ARSSSID
  - ARSTSIN
  - ARSCODS2
  - ARSCODS3
- From SARSINST to *hlq*.USERJCL
  - ARSCTBL1
- From SARSINST to *hlq*.USERPROC
  - ARSBPKG
  - ARSBPLAN
  - ARSTIAD

Task: Customize DB2® package and plan DDL

About this task

You must now convert the DBRM (database request module) library members to application packages and plans and then grant access to the plans.

Note: Each DB2 environment can have multiple subsystem identifiers. Make sure that the subsystem identifiers in the members match the ones in use in your DB2 environment.

Modify the members as follows:

- In USERDB2:

  ARSKDOCG
  Modify this member to reflect the DB2 package name to use for the index migration program and document retrieval program.

  ARSKRUN, ARSLRUN
  Modify these members to reflect the DB2 RUNLIB and package and plan owner.

  ARSLDOCG
  Modify this member to reflect the DB2 plan name and package name to use for the index migration program and document retrieval program.

  ARSCODS2, ARSCODS3
  Modify these members to reflect the correct SQLID for the referenced tables.
- In USERJCL:

  ARSCTBL1
  Modify this member to specify the proper USERPROC library on the //JOPBPROC statement.
- In USERPROC:

  ARSBPKG, ARSBPLAN, ARSTIAD
  Modify the high-level qualifiers used in these members as instructed in the member prolog:

  ?ARS?
  User libraries

  ?DB2?
  DB2 library qualifier

  ?SMP?
  Content Manager OnDemand Version 9 target library qualifier

Task: Prepare the document retrieval environment

About this task

Run the ARSCTBL1 job after completing all customization.

- The steps for ARSCODS2 and ARSCODS3 should have been run in the past during the migration process and do not need to be run again if all of the synonyms still exist.
- The steps for ARSLDOCG and ARSGIAF2 should have been run in the past during the migration process to bind and grant the plan for the document retrieve routine, and do not need to be run again if the plan has already been created and granted.
- The step for ARSKDOCG binds the package for the document retrieve routine and needs to be rerun any time a change is made to the ARSZDOCG document retrieve routine.

Next steps

To view the documents stored by Content Manager OnDemand Version 2.1 or IAFC Version 1.3 for the indexes you have migrated, be sure the `ars.cfg` file contains the following two parameters:

```
ARS_MIGRATE_SSID=DB2 Subsystem-Id
ARS_MIGRATE_PLAN=DB2 Plan Name for program ARSZDOCG
```

After performing a migration of index values, you might be unable to retrieve the documents from the Content Manager OnDemand Windows client when accessing the Version 10.5 server. Typically in this situation, when you select a document for viewing with your Windows client, you receive the following message: `The server failed while retrieving a document.`

In this instance, check the ARSSOCKD started task for messages similar to the following:

```
ARS0024E ODUSR OBJECT >1FAAA< IN APPLICATION GROUP >xxx< NOT FOUND IN NODE
>IOD.COLLECT.SGROUP00.M1DT03D<  SRVR->hostname.your.company.com  ipaddr<-
ARS0020E ODUSR SM ERROR: ARSZDOCG: 00000017(00000010) OAM ERROR: FUNCTION=
```

```
ACCESS   RETURN CODE= 000000C8 REASON-CODE= 00C10208, RC=23, REASON=0,
FILE=ARSSMSMS.C, LINE=794  SRVR->hostname.your.company.com ipaddr<-
```

The call attachment facility (CAF) received requests from the same TCB to CONNECT to the DB2 subsystem and to another subsystem.

This abend reason code is issued by the following CSECT: DSNACA00.

You can continue processing by correcting the request. Specify the correct subsystem (SSID) with the **ARS_MIGRATE_SSID** parameter in the `ars.cfg` file. For example, if your DB2 SSID is DSNA, add the following to `/usr/lpp/ars/V10R5M0/config/ars.cfg`: `ARS_MIGRATE_SSID=DSNA`.

Also, if you chose a plan name other than ARSZDOCG in member ARSLDOCG of library SARSDB2 when you performed the plan bind, you should specify the **ARS_MIGRATE_PLAN** parameter in `/usr/lpp/ars/V10R5M0/config/ars.cfg` to properly identify the plan that Content Manager OnDemand should use for the ARSZDOCG program.

Large object support for migrated indexes

Reports that are stored directly in IBM Content Manager OnDemand for z/OS can be defined to use large object support. Large object allows for improved response time when viewing large documents because these documents are stored as multiple small segments. Each segment is returned to the client one at a time for viewing, instead of retrieving the entire document before it can be viewed. To support the large object technique, the load process builds an internal structure called the large object header that is stored as part of the document and provides information as to where the individual segments are located.

For indexes that were migrated from IBM DB2 Content Manager OnDemand for z/OS, line print reports can also use the large object support. However, there is no large object header in the Version 2.1 documents. To provide large object support for these documents, the large object header is built at view time.

For the initial implementation of this technique, the size of the large object header structure was the same as the size of the document. Typically, this approach would not cause a problem at view time. However, when a very large document is requested for viewing, having the large object header structure size match the document size could result in a Content Manager OnDemand memory issue. The new procedure was developed to allow customers to set a fixed size of 10,000 bytes for the large object header structure for selected Application Groups. The only migrated indexes affected by this new sizing are those which use the fix programs ARSZLOFX and ARSMIGF3. Existing migrated indexes which do not use these fix programs will continue to have the large object header structure size match the document size.

The new process might improve performance for documents that are accessed frequently. As Content Manager OnDemand Version 2.1 documents are retrieved, their large object headers are dynamically built and stored in the ARSODHDR table. Subsequent retrievals of a document will use the header that is stored in the ARSODHDR table. A column, HDR_TIMESTAMP, is provided for usage monitoring and clean up of unused rows. This column is updated with the current date and time each time a document is accessed.

Installing the New Large Object Header Table

About this task

The ARSODHDR table is used by the ARSZDOCG retrieval program. This table retains the large object header and is used for all subsequent retrieval requests. It has the following format:

HDR_DOC_NAME
HDR_OFFSET
HDR_TIMESTAMP
HDR_LAST_OFFSET
HDR_LAST_LENGTH
HDR_LO_HDR

The timestamp field is updated every time the row is accessed and you can use it to manage the table. Single segment documents will not have their headers saved in this table, rather the large object header will be rebuilt during each view request. Inactive entries, those with no current retrievals in progress, might be removed from the table at any time.

Procedure

To install the ARSODHDR table, do the following steps:

1. Copy members to user libraries.

   Use the user libraries that you created for the index migration routines when you perform this step.

   - From SARSINST to *hlq*.USERDB2, copy the following members

     ```
     ARSAOD3
     ARSCODH
     ARSCODS3
     ```

   - From SARSINST to *hlq*.USERJCL, copy ARSATBL1. Back up the existing member before replacing with this new version.
2. Define the ARSODHDR.

   Run ARSATBL1, step ARSCODH (or you can run this member under SPUFI). Modify member ARSCODH, in the USERDB2 library, to your standards (table creator / high qualifier).
3. Add the HDR_LEN column to the ARSOD table.

   Make the appropriate modifications (table creator / high qualifier) to the ARSAOD3 member in the USERDB2 library. Run ARSATBL1, step ARSAOD3. This can be run under SPUFI.
4. Establish the DB2 synonym for ARSODHDR.

   Run the ARSATBL1 job, step ARSCODS3. Make the appropriate modifications to the ARSCODS3 member of the USERDB2 library (ARSODHDR table creator name / high qualifier).
5. Run the bind for ARSZDOCG.

   Run the bind of the package for the new DBRM for program ARSZDOCG by running step ARSKDOCG of the ARSATBL1 job.

Installing and running the large object fix programs

About this task

Making sizing changes to the large objects involves running two programs. The first (ARSZLOFX) updates the DOC_LEN field in the application group data tables for the migrated indexes for a specified application group, and optionally a range of dates. The second program (ARSMIGF3) updates the ARSODIND table corresponding to the rows modified by ARSZLOFX.

You do the following tasks to install the items in this procedure:

- copy members to the user libraries
- customize DB2 package and plan DDL
- prepare the DB2 environment

Procedure

To install ARSZLOFX, do the following steps:

1. Copy members to user libraries.

   Use the user libraries that you created for the index migration routines when you perform this step.

   - From SARSINST to *hlq*.USERPARM, copy ARSZLOFP, renaming the member to ARSZLOFX.
   - From SARSINST to *hlq*.USERDB2, copy the following members:

     ```
     ARSKL0FX
     ```

ARSLL0FX
ARSGL0FX

- From SARSINST to *hlq*.USERJCL, copy the following members:

  ARSZLOF, renaming the member to ARSZLOFX
  ARSATBL1. Back up the existing member before replacing with this new version

2. Customize the DB2 package and plan DDL.

   You must now convert the DBRM (database request module) library members to application packages and plans and grant access to the plans.

   Note: Each DB2 environment can have multiple subsystem identifiers. Make sure that the subsystem identifiers in the members match the ones in use in your DB2 environment.

   a) Make modifications to the members in USERDB2.

      Modify ARSKLOFX to reflect the DB2 package name to use for the ARSZLOFX program.
      Modify ARSLLOFX to reflect the DB2 plan name and package name to use for the ARSZLOFX program.
      Modify ARSGLOFX to reflect the DB2 plan name for the ARSZLOFX program.

   b) Modify member ARSATBL1 in USERJCL to specify the proper USERPROC library on the //JOPBPROC statement.

      See the note in step “4” on page 163.

3. Prepare the DB2 environment.

   This step is only for environments where the Content Manager OnDemandlibrary server is on one MVS image and the Content Manager OnDemand object server is on a different MVS image. If your environment has a combined Content Manager OnDemand library and object server, do not do this step.

   The ARSZLOFX program runs on the library server and accesses several tables that reside on that server. It also accesses the ARSOD table that resides on the object server. The following steps describe how to access the ARSOD table on the object server from the library server. The following table describes the references used in those steps.

Table 6. References for the ARSOD table procedure

Reference	Description
hlq1	Owner ID for the ARSAG, ARSNODE and ARSSEG tables on the library server.
hlq2	Owner ID for the ARSOD table on the object server.
link1	The LINKNAME for the link from the DB2 subsystem on the MVS image of the library server to the DB2 subsystem on the MVS image of the object server.
link2	The LINKNAME for the link from the DB2 subsystem on the MVS image of the object server to the DB2 subsystem on the MVS image of the library server.

   a) Define the location of the object server to the DB2 subsystem of the library server.

      For the bind of the package to be successful, the ARSOD table on the object server needs to be defined to the library server through a DB2 alias . This package bind occurs on the MVS image of the library server.

      Coordinate with your DB2 DBA to update the SYSIBM.IPNAMES and SYSIBM.LOCATIONS tables on the MVS image of the library server to create a LINKNAME (*link1*) pointing to the MVS image where the object server resides.

   b) Define a DB2 alias on the library server to the ARSOD table on the object server.

The bind of the package on the library server for the ARSZLOFX program resolves the references in the program to the ARSOD table on the object server. This step defines the alias for the name resolution.

When you bind your package and plan on the library server, the ARSKRUN member of the USERDB2 library contains an **OWNER** parameter. The value of this parameter is the owner-id of the Content Manager OnDemand tables on the library server. Use this value as the *hlq1* value in the following SQL statement: `CREATE ALIAS hlq1.VARSOD for link1.hlq2.ARSOD` The *hlq2* value is the owner of the ARSOD table on the object server. *link1* is the value you used when creating the SYSIBM.LOCATIONS entry in step "3.a" on page 161.

c) Define the location of the library server to the DB2 subsystem of the object server.

Several of the tables referenced in the ARSZLOFX program reside on the library server. A second package bind of this program must be performed on the object server. For this bind to work, these tables must be defined with an alias on the object server. A LINKNAME must be created on the object server that points to the library server before the alias can be created.

Coordinate with your DB2 DBA to update the SYSIBM.IPNAMES and SYSIBM.LOCATIONS tables on the MVS image of the object server to create a LINKNAME (*link2*) pointing to the MVS image where the library server resides.

d) Define the DB2 alias on the object server to the tables used on the library server.

The bind of the package on the object server for the ARSZLOFX program resolves the references in the program to the ARSAG, ARSNODE and ARSSEG tables on the library server. This step defines the alias for the name resolution.

When you bind your package on the object server, set *hlq1* to the owner of the ARSOD table on the object server in the CREATE ALIAS. Set *hlq2* to the owner of the Content Manager OnDemand tables on the library server. Set *link2* to the value you used when creating the SYSIBM.LOCATIONS entry in step "3.c" on page 162.

```
CREATE ALIAS hlq1.ARSAG FOR link2.hlq2.ARSAG;
CREATE ALIAS hlq1.ARSNODE FOR link2.hlq2.ARSNODE;
CREATE ALIAS hlq1.ARSSEG FOR link2.hlq2.ARSSEG;
```

e) Bind the package on the object server.

A bind of the package is required on the object server and is included in the PKLIST of the Plan Bind on the library server.

This bind uses *link1* defined in step "3.a" on page 161 and *hlq2* as the owner of the ARSOD table on the object server. This bind runs on the image where the object server resides.

```
BIND PACKAGE(link1.ARSZLOFR)
MEMBER(ARSZLOFX)
ACTION(REP)
EXPLAIN(NO)
VALIDATE(BIND)
ISOLATION(CS)
RELEASE(COMMIT)
OWNER(hlq2)
```

f) Modify the bind plan parameters on the library server.

The ARSLLOFX member of the USERDB2 library on the library server is shipped containing the following:

```
BIND PLAN(ARSZLOFX)
PKLIST(ARSZLOFX.*)
```

The PKLIST must be modified to include the package from step "3.e" on page 162:

```
BIND PLAN(ARSZLOFX)
     PKLIST(ARSZLOFX.*,
            link1.ARSZLOFR.*)
```

The plan references two packages. The first is the one bound on the library server (in step “4” on page 163) and the second is the one bound on the object server (in step “3.e” on page 162). This allows the execution of the ARSZLOFX program on the library server to access the ARSOD table which resides on the object server.

4. Prepare the DB2 package and plan.

   Run the ARSATBL1 job on the library server after completing all customization. Comment out all but the following steps in the ARSATBL1 JCL member:

   ```
   ARSLLOFX
   ARSKLOFX
   ARSGLOFX
   ```

   Note: The binding of the package and plan for the ARSZLOFX program, using the ARSKLOFX and ARSLLOFX USERDB2 members, must be done with a DB2 owner value equal to that for which the base Version 7.1 or Version 8.4 tables were created. That is the value specified in the **SRVR_INSTANCE_OWNER** parameter of the ARS.INI file.

   The ARSKRUN and ARSLRUN members of the USERDB2 library contain the **OWNER** parameter used during the package and plan binds. If the OWNER value needed for the ARSZLOFX program is different from what is currently there, modify the value before running the package and plan binds.

Run the large object fix programs

About this task

The ARSZLOFX program requires input, through the INAGLIST DD statement, of the application group names you want the program to work with. An optional date range can be entered to limit how much work is done at one time. For each application group name input record, the program identifies all of the application group data tables (where the indexes are stored) and the ARSOD table entries (limited by the optional date range) and updates the length value used for the large object header. A report is generated identifying the updates that were made.

The ARSZLOFX program generates an output file to the ODINDOUT DD statement. This file is used as input to the second step, the ARSMIGF3 program. The ARSMIGF3 program updates the Index Objects in the ARSODIND table.

You must prepare the input for the ARSZLOFX program before running this job.

Procedure

1. Prepare the inputs for ARSZLOFX.

   The *hlq*.USERPARM(ARSSSID) library member must contain:

   ```
   record 1
        columns 1-8: the constant DB2SSID=
        columns 9-12: the DB2 subsystem ID
   ```

   The *hlq*.USERPARM(ARSZLOFX) library member must contain:

   ```
   record 1
        columns 1-8: the constant DB2PLAN=
        columns 9-16: the DB2 plan name for program ARSZLOFX

   record 2
        columns 1-5: the constant TRACE (optional parameter which will generate
                     debugging messages to SYSOUT)
   ```

   The INAGLIST DD statement is required. It has a record length of 82. It contains:

   ```
   record 1 through n
        columns 1-60: one Application Group Name per record
        columns 62-71: start date as YYYY-MM-DD
        columns 73-82: end date as YYYY-MM-DD
   ```

If a start date is specified, an end date must also be specified.

2. Run the ARSZLOFX and ARSMIGF3 programs.

   Changes required to the USERJCL member ARSZLOFX are described in comments at the beginning of this member.

3. Submit the ARSZLOFX member of the USERJCL library.

   This program must be run on the Library Server, and must be run with the same userid as the server. The report from ARSZLOFX is written to the STATSOUT DD. Output from ARSMIGF3 is written to SYSPRINT Trace and error messages are written to the SYSOUT DD.

Chapter 48. Storage manager expiration (deleting OAM and VSAM objects)

Process description

Content Manager OnDemand usually expects objects to be deleted based on the value of the Life of Data and Indexes field. This value is specified on the **Storage Management** page of an application group by using the Content Manager OnDemand administrative client. Content Manager OnDemand will keep index entries for the period of time specified in that field. It is expected that the storage manager will be configured to keep the objects around for that same period of time. If the storage manager (VSAM or OAM) is not configured in that way, it is possible that the storage manager will delete objects before Content Manager OnDemand deletes the indexes that reference them. This causes a document list to be presented to the user that contains entries for which documents cannot be retrieved.

If an installation does not desire to configure their SMS management policies to expire objects in accordance with the values that they have specified in the Life of Data and Indexes fields, it is possible to have the deletions of the object by the storage manager cause the deletion of the Content Manager OnDemand indexes.

This requires the use of the ARSEXPIR program and the collection of SMF data that reflects the object deletions. If you are using VSAM, you must collect type 65 SMF records.

SMF collects management data in the SYS1.MAN*x* data sets. These data sets must be periodically dumped to other data sets using the IFASMFDP program. Those other data sets are then used as input to ARSEXPIR.

If you are using OAM, you must collect type 85 and install ARSSMFWR as the CBRHADUX OAM auto-delete exit.

Note:

1. Storage nodes must be unique across all storage sets. The program uses the OAM Collection name or the high-level qualifier of the VSAM LDS to determine what storage node the object being deleted belongs to. If the names are not unique, the correct storage node cannot be determined. ARSEXPIR will not complete successfully if it detects this condition.
2. All objects recorded as being deleted by the SMF records will have their index entries deleted regardless of the settings in the Life of Data and Indexes section on the **Storage Management** page of the application group. If you are going to use Storage Management expiration, then all application groups should have the Expiration Type set to Storage Manager.

The data must be collected and ARSEXPIR periodically run with the data to cause the removal of the indexes that point to those objects.

Note: A load of a document can produce multiple objects. If one of those objects is deleted, all the indexes created by that load will be deleted.

For information about collecting SMF data, see *z/OS MVS System Management Facilities (SMF)*. The types that need to be collected for ARSEXPIR are the type 65, 85, and the user type written by ARSSMFWR. Other records may be collected, however they are ignored by ARSEXPIR.

For more information about the ARSEXPIR program, see *IBM Content Manager OnDemand for z/OS: Administration Guide*.

ARSSMFWR

OAM does not usually write SMF records to record individual object deletion when performing its maintenance cycle. To collect these events, the OAM auto-delete exit (CBRHADUX) must be modified to write a user SMF record. A sample is provided as ARSSMFWR (shipped in SARSINST). The record that needs to be written is shipped as ARSSMF in SARSINST. If you are already using the CBRHADUX exit, you must modify it to write the record. The record type written by the exit must also be specified by the **USERSMF** parameter to ARSEXPIR.

Note: The exit should only write an SMF record if it permits OAM to delete the record (return code ADUDELOK). Writing the record and preventing OAM from deleting the record will cause Content Manager OnDemand to erroneously remove the indexes.

After assembling, the sample must be link-edited as CBRHADUX into SYS1.LINKLIB. After link-editing, LLA must be refreshed. After refreshing, OAM must be recycled. For more information about the CBRHADUX exit, see *z/OS DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*.

Expiration processing without SMF

As an alternative to using SMF for expiration processing, expiration of OAM objects can use a Db2 table. When OAM deletes an object, rows will be added to the table. When expire processing is run, it will use the rows in that table to determine what Content Manager OnDemand objects to expire instead of, or in addition to, using the SMF records.

If OAM uses a different Db2 subsystem than the Db2 subsystem that Content Manager OnDemand uses, you must create the ARSOAM_DELETE table in the Db2 subsystem that is used by OAM. Use the following sample to create the ARSOAM_DELETE table as the **SRVR_INSTANCE_OWNER** (in `ars.ini`) for the instance for which deletion events need to be captured.

The following is the ARSTBEXP sample:

```
ARSTBEXP JOB (account),userid,MSGCLASS=A
//*************************************************************
//*
//*  FUNCTION: CREATE THE Db2 TABLES AND TABLESPACES REQUIRED
//*            TO RUN ARSEXOAM.
//*
//*  FMID:     H272950
//*
//*  NOTES:    ON MULTIPLE CPU SYSTEMS, THIS JOB MUST BE RUN
//*            ON THE SYSTEM WHERE Db2 EXECUTES.
//*
//*************************************************************
//*
//* TO CREATE THE ONDEMAND ARSOAMDL TABLESPACE
//*
//* 1. PROVIDE A SUITABLE JOBCARD FOR YOUR ENVIRONMENT.
//*
//* 2. REPLACE THE FOLLOWING SYMBOLIC FIELDS:
//*    db2ssid  - Db2 SUBSYSTEM NAME
//*    LIB      - Db2 RUNTIME LOAD LIBRARY
//*    PLAN     - DEFAULT Db2 PLAN NAME IS DSNTIA10
//*    ARSUSER  - DEFAULT SQLID IS ARSUSER
//*
//* 3. (OPTIONAL) CHANGE THE FOLLOWING CREATE PARAMETERS:
//*    DATABASE   - DEFAULT DATABASE IS ARSDBASE
//*    STOGROUP   - DEFAULT STORAGE GROUP IS ARSSGRP
//*    TABLESPACE - CHANGE DEFAULT TABLESPACE NAMES
//*    PRIQTY     - PRIMARY ALLOCATIONS
//*    SECQTY     - SECONDARY ALLOCATIONS
//*    BUFFERPOOL - 4K BUFFERPOOL NAME
//*    STEPLIB    - SPECIFIES THE SDSNLOAD DATA SET
//*
//* 4. SUBMIT THE JOB.
//*
//*************************************************************
//* STEP1: CREATE THE ARSEXOAM TABLESPACE, TABLE, AND INDEXES
```

```
//************************************************************
//STEP1    EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=DB2.V10R5M0.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(db2ssid)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA10) -
      LIB('DB2.RUNTIME.LOADLIB')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
 SET CURRENT SQLID='ARSUSER';
 CREATE TABLESPACE ARSOAMDL
      IN ARSDBASE
      USING STOGROUP ARSSGRP
      PRIQTY 75
      SECQTY 75
      SEGSIZE 64
      BUFFERPOOL BP0;

 CREATE TABLE ARSOAM_DELETE
      (
        ODCLNAME  CHAR(44)    NOT NULL,
        ODNAME    CHAR(44)    NOT NULL,
        AGID_NAME CHAR(8)     NOT NULL WITH DEFAULT,
        NID       SMALLINT    NOT NULL WITH DEFAULT,
        DOC_NAME  CHAR(11)    NOT NULL WITH DEFAULT
      )
      IN ARSDBASE.ARSOAMDL;

  CREATE INDEX ARSOAM_DELETE_0
      ON ARSOAM_DELETE
      (ODCLNAME, ODNAME)
      USING STOGROUP ARSSGRP
      PRIQTY 60
      SECQTY 60
      BUFFERPOOL BP0;

  CREATE INDEX ARSOAM_DELETE_1
      ON ARSOAM_DELETE
      (AGID_NAME, NID, DOC_NAME)
      USING STOGROUP ARSSGRP
      PRIQTY 15
      SECQTY 15
      BUFFERPOOL BP0;
/*
```

After creating the table, you must create a trigger on the OAM GROUP*xx*.OSM_OBJ_DIR table that contains the Content Manager OnDemand objects. A sample ARSEXTRG member is provided in SARSINST for you to modify and use. If multiple servers are running for multiple database owners, you have to create multiple triggers for each.

The following is ARSEXTRG sample:

```
//ARSEXTRG JOB (account),userid,MSGCLASS=A
//************************************************************
//*
//*  FUNCTION: CREATE A TRIGGER ON THE OAM DIRECTORY TABLE TO
//*            INSERT THE COLLECTION AND OBJECT INTO THE OD
//*            ARSOAM_DELETE TABLE
//*
//*  FMID:     H272950
//*
//*  NOTES:    ON MULTIPLE CPU SYSTEMS, THIS JOB MUST BE RUN
//*            ON THE SYSTEM WHERE THE OAM Db2 EXECUTES.
//*
//************************************************************
//*
//* TO CREATE THE ARSDEL TRIGGER:
//*
//* 1. PROVIDE A SUITABLE JOBCARD FOR YOUR ENVIRONMENT.
//*
//* 2. REPLACE THE FOLLOWING SYMBOLIC FIELDS:
//*    db2ssid  - DB2 SUBSYSTEM NAME
//*    LIB      - DB2 RUNTIME LOAD LIBRARY
//*    PLAN     - DEFAULT DB2 PLAN NAME IS DSNTIA10
//*    ARSUSER  - DEFAULT SQLID IS ARSUSER
//*
//* 3. (OPTIONAL) CHANGE THE FOLLOWING CREATE PARAMETERS:
```

```
//*    STEPLIB    - SPECIFIES THE SDSNLOAD DATA SET
//*    ARSDEL     - SPECIFIES THE TRIGGER NAME
//*    GROUP00    - NAME OF THE OAM DIRECTORY TABLE THAT
//*                 CONTAINS THE OD OAM OBJECTS
//*
//* 4. SUBMIT THE JOB.
//*
//*************************************************************
//* STEP1: CREATE THE ARSDEL TRIGGER
//*************************************************************
//STEP1    EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=DB2.V10R5M0.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(db2ssid)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA10) -
      LIB('DB2.RUNTIME.LOADLIB') PARMS('SQLTERM(?)')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 SET CURRENT SQLID='ARSUSER'?

 CREATE TRIGGER ARSDEL AFTER DELETE ON GROUP00.OSM_OBJ_DIR
   REFERENCING OLD AS OLD_OBJ FOR EACH ROW MODE DB2SQL
   BEGIN ATOMIC
     INSERT INTO ARSUSER.ARSOAM_DELETE
        SELECT C.ODCLNAME, OLD_OBJ.ODNAME, ' ', 0, ' '
          FROM OAMADMIN.CBR_COLLECTION_TBL C
          WHERE C.ODCLID = OLD_OBJ.ODCLID;
   END?
/*
```

In addition to the ARSEXPIR program , a new program (ARSEXOAM) will delete the indexes that are associated with objects deleted by OAM by processing the ARSOAM_DELETE table. The ARSEXOAM program will only delete indexes for objects stored in OAM. If you are using VSAM linear data sets to store objects, you will still have to capture SMF records for VSAM linear data and run ARSEXPIR. You can choose one of the following options:

1. Capturing the SMF records for both VSAM linear data and OAM deletions and using ARSEXPIR for both
2. Using ARSEXPIR just for VSAM linear data and using the Db2 trigger and ARSEXOAM program for OAM

If you are using only OAM, you may choose to use only the ARSEXOAM program

The following is a sample job:

```
//ARSEXOAM EXEC PGM=ARSEXOAM
//STEPLIB  DD   DISP=SHR,DSN=ARS.SARSLOAD
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//DSNAOINI DD   PATH='/usr/lpp/ars/V10R5M0/config/cli.ini'
//ARSPARM  DD   *
ODINSTANCE=instance
ODUSER=userid
ODUSERPW=passwd
COMMITCNT=nnn
UNLOADMAX=nnn
```

Where:

STEPLIB
Specifies the SARSLOAD data set.

SYSPRINT
Specifies the message output from the ARSEXOAM utility

SYSOUT
Specifies the location for message output directed to stderr.

DSNAOINI
Specifies the location of the ODBC configuration file. This can be an MVS data set.

ARSPARM

Specifies the location for the input control statements. It accepts the following control statements, 1 per line:

ODINSTANCE= *instance*

Required. Specifies the instance name of theContent Manager OnDemand library server whose objects will be deleted. The ARSEXOAM program must be able to locate the SRVR_INSTANCE_OWNER for that instance name in the `/usr/lpp/ars/V10R5M0/config/ars.ini` of the MVS on which the ARSEXOAM runs

ODUSER=*userid*

Optional. Specifies a Content Manager OnDemand user ID that exists on the server.

ODUSERPW=*passwd*

Optional. Specifies the password associated with the user ID.

COMMITCNT=*nnn*

Optional. Specifies how many fetches are done between commits. Default is 1000. If 0, no commits are done while fetching

UNLOADMAX=*nnn*

Optional. Specifies how many objects to pass to arsadmin for each invocation. The default is to pass all of them in a single invocation of arsadmin. A commit is also done before invoking arsadmin.

A sample job ARSEXOAM is shipped in the SARSINST data set:

```
//ARSEXOAM JOB (account),userid,MSGCLASS=A
//*************************************************************
//STEP1    EXEC PGM=ARSEXOAM,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=ARS.V10R5M0.SARSLOAD
//         DD DISP=SHR,DSN=DB2.V10R5M0.SDSNEXIT
//         DD DISP=SHR,DSN=DB2.V10R5M0.SDSNLOAD
//DSNAOINI DD PATH='/usr/lpp/ars/V10R5M0/config/cli.ini'
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//ARSPARM  DD *
  odinstance=ARCHIVE
  oduser=arsuser
  oduserpw="password"
```

When you use the ARSEXOAM program to expire documents from OAM, it is possible to improve the unload performance by using the **ARS_EXPIRE_REQLIMIT** parameter. This parameter controls the number of load IDs that are sent to the server in a single expiration request at a time. The default value is 1, meaning that a separate request for each load ID is processed. Load IDs for the same application group can be grouped together up to the value specified for the **ARS_EXPIRE_REQLIMIT** parameter. However, the grouped load IDs must be for the same application group.

Part 10. Exits

Exits are points within the Content Manager OnDemand code where control can be transferred to programs that you develop to provide customized functionality. Content Manager OnDemand provides six user exits, as illustrated in the following figure.

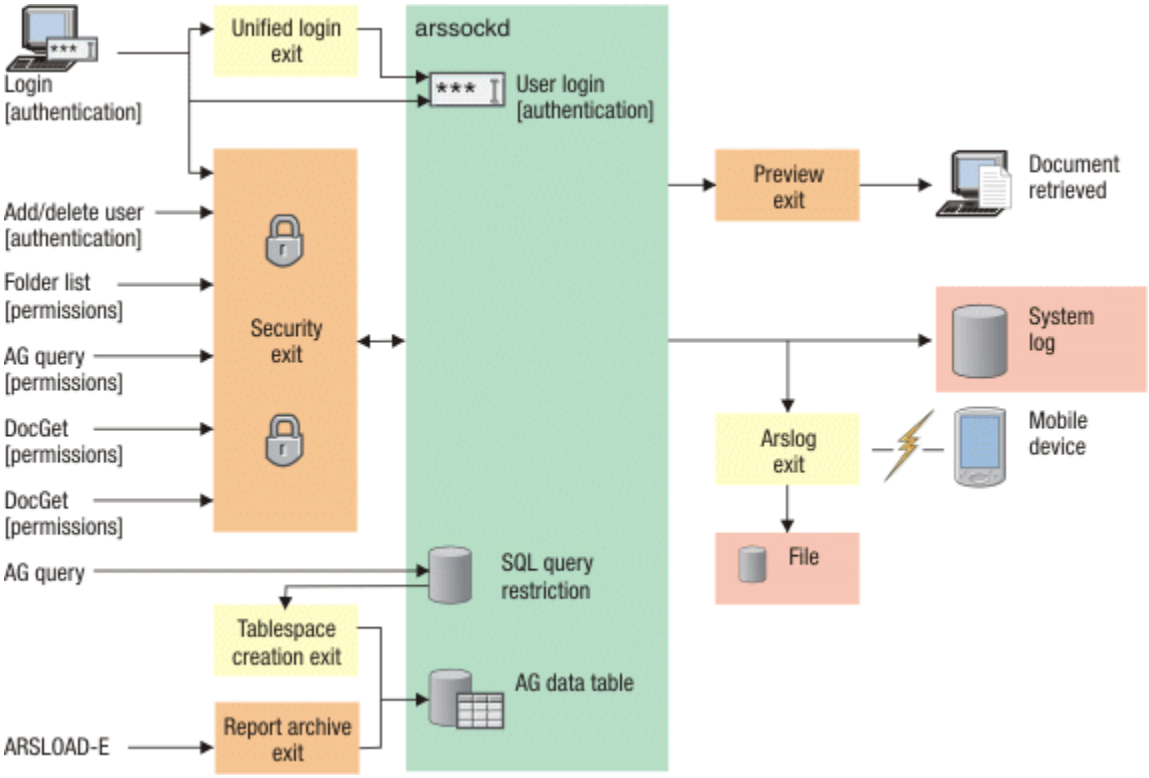

Figure 9. Overview of Content Manager OnDemand exits

Unified login exit
Called at logon time to simplify the logon process.

User security exit
Called to perform user authentication at logon time and during folder, application group, and document access operations.

Tablespace creation exit
Called when a new application group data table is created.

Report archive exit
Called when an object is stored in the Content Manager OnDemand archive.

Client preview exit
Called when a user makes a request to view a document.

System log exit
Called when an entry is recorded to the system log.

Chapter 49. System log exit

Overview

You can use the administrative client to configure Content Manager OnDemand to record information, warning, and error messages in the system logging facility. Content Manager OnDemand can record messages about system activity, such as when users log on and log off the system and application group activity, such as when clients query and retrieve data. In addition, you can configure Content Manager OnDemand to send these messages to the ARSLOG installation exit. You can use the ARSLOG installation exit to issue WTOs, compile statistics, and generate accounting information. The generated information can be written to a log file, sent as an email or forwarded to another device, such as a mobile device. The following figure shows an overview of the system log exit. In the following figure, the ARSLOG exit routine is named ARSULOG.

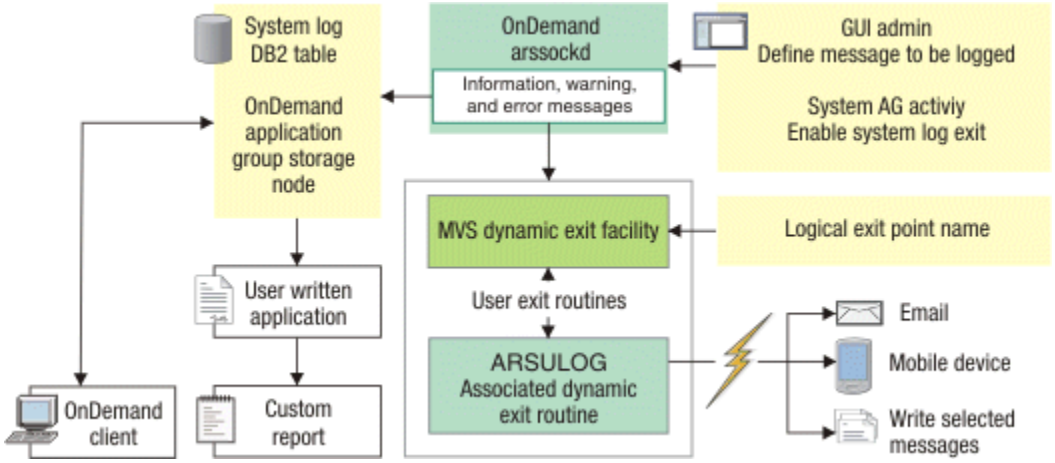

Figure 10. Overview of the system log exit

Configuring the system log user exit program requires the following steps:

1. Use the administrative client to perform the following tasks:
   - Enable Content Manager OnDemand to record system messages and determine the types of system messages recorded in the system log by selecting the appropriate options in the **System Parameters** dialog box (by using the administrative client).
   - Enable Content Manager OnDemand to record application group messages by selecting the appropriate option in the **System Parameters** dialog box (by using the administrative client).
   - Determine the types of application group messages Content Manager OnDemand records in the system log by setting **Message Logging** options when you create or update an application group (by using the administrative client).
   - Enable Content Manager OnDemand to send messages to the system log user exit by selecting the appropriate options in the **System Parameters** dialog box (by using the administrative client).
2. Associate an exit routine with the ARSLOG exit. This can be done in one of the following ways:
   - The EXIT statement of the PROG*xx* parmlib member. See *z/OS MVS Initialization and Tuning Reference* for more information about the PROG*xx* parmlib member.
   - The **SETPROG EXIT** operator command. For example: `SETPROG EXIT, ADD, EXITNAME=ARSLOG, MOD=ARSULOG, DSNAME=ODEXITS.LOAD`. For information about the **SETPROG EXIT** command, see the *z/OS MVS System Commands*.

After you have completed these steps, Content Manager OnDemand automatically records messages in the system log and sends the messages to the ARSLOG exit routine. The exit is implemented as an MVS dynamic exit.

Messages are sent to the ARSLOG exit routine as they are generated (that is, in real time). The messages are recorded in the Content Manager OnDemand system log table, with one row per generated message. You can also write an application that retrieves the records from the Content Manager OnDemand system log and generate custom reports that are specific to your needs.

The online help for the administrative client provides information about enabling Content Manager OnDemand to generate messages and send them to the ARSLOG exit routine. The online help also provides information about how to select which application group messages Content Manager OnDemand generates and sends to the ARSLOG exit routine.

ARSLOG exit routine

This section contains interface information for the ARSLOG exit routine.

Exit routine environment

The ARSLOG exit routine is entered in the following environment:

Authorization
Problem state and PSW key 8

Dispatchable Unit Mode
Task

Cross memory mode
PASN=HASN=SASN

AMODE
31-bit

ASC mode
Primary

Interrupt status
Enabled for IO and external interrupts

Locks
Primary

Programming considerations

The ARSLOG exit must be reentrant.

Registers at entry

On entry to the ARSLOG exit routine, the registers contain the information listed in Table 7 on page 174.

Table 7. Registers at entry to the ARSLOG exit routine.

Register	Contents
GPR 0	Does not contain any information for use by the ARSLOG exit

Table 7. Registers at entry to the ARSLOG exit routine. (continued)	
Register	**Contents**
GPR 1	Address of an eight- or nine-word structure. **Word 1** Address of the null-delimited instance name **Word 2** Address of the null-delimited timestamp string **Word 3** Address of the null-delimited log record ID string **Word 4** Address of the null-delimited Content Manager OnDemand user ID string **Word 5** Address of the null-delimited user accounting string **Word 6** Address of the null-delimited severity. **Word 7** Address of the null-delimited message number string **Word 8** Address of the null-delimited message text. If this is the last address, the high-order bit will be turned on. **Word 9** Address of the null terminated string of the HFS file containing the document data. The document data is the data that is displayed when you select documents in the Content Manager OnDemand System Log that contain viewable data. If there is no document data, this will be a null string. If this is the last address, it will have the high-order bit turned on. This file will be deleted after the exit is invoked.
GPRs 2-12	Do not contain any information for use by the ARSLOG exit
GPR 13	Address of an 18-word save area
GPR 14	Return address
GPR 15	Entry point address

Return specifications

The ARSLOG user exit does not need to set any return codes.

System Log database table

The System Log database table contains one row for each message that Content Manager OnDemand generates. Table 8 on page 176 lists the fields, data types, and descriptions of the fields of the records in the System Log database table.

Table 8. System Log database table

Column Name	Data Type	Size	Description
time_stamp	Date/Time (TZ)	4	The time stamp of the log record in the Content Manager OnDemand internal date format. For more information, see the ARSDATE program reference.
userid	VARCHAR	128	The user ID of the user that generated the log record.
severity	CHAR	1	The severity of the log record. **1** Alert **2** Error **3** Warning **4** Information **5** Debugging
msg_num	SMALLINT	2	The message number of the log record.
msg_text	VARCHAR	2000	The message text of the log record.
appl_id	CHAR	1	Determines whether Content Manager OnDemand overhead information is valid. **A** Not applicable. The overhead information does not apply to the log record. However, the overhead information may be useful for other purposes. For example, a log record created when a document is retrieved contains overhead information about the document. **N** No. The overhead information does not contain useful information. **Y** The overhead information contains information about the document belonging to this particular log record.
log_id	INTEGER	4	The identifier for the Content Manager OnDemand client connection.
account	VARCHAR	60	The user's accounting information.
doc_name	VARCHAR	11	The name of the object.
doc_off	INTEGER	4	The offset of the document within the compressed object.

Table 8. System Log database table (continued)

Column Name	Data Type	Size	Description
doc_len	INTEGER	4	The length of the document within the compressed object. A 0 means all of the data.
comp_off	INTEGER	4	The offset of the document within the compressed object.
comp_len	INTEGER	4	The length of the document within the compressed object. A 0 means all of the data.
annot	CHAR	1	Determines whether annotations exist for the document. Applies only if the annotation flag is set (YES) for the application group.
comp_type	CHAR	1	The method used to compress document data.
resource	INTEGER	4	The resource identifier for the document.
pri_nid	SMALLINT	2	The primary storage node identifier.
sec_nid	SMALLINT	2	The secondary storage node identifier.

Chapter 50. User security exit

The Content Manager OnDemand security system interface exits

The Content Manager OnDemand security system interface exits allows an installation to augment the user authentication and resource authorization security processing that is provided with the Content Manager OnDemand system. The following activities or events are controlled by the exits:

User authentication (checking user security)

- Logon
- Change Password
- Add User ID
- Delete User ID

Resource Authorization (checking user permissions)

- Access to a Content Manager OnDemand folder
- Access to a Content Manager OnDemand application group
- Restrict access to specific documents
- Control the SQL search criteria used for searching folders.

The security interface consists of two high-level language exit routines (ARSUSEC and ARSUPERM) and a single assembler exit interface routine (ARSUSECX). The exits that you implement depend on your security requirements.

At the lowest level, an installation-written exit routine (such as ARSUSECZ) or set of exit routines can interact through SAF with a security system such as RACF (or any other SAF compliant security system) to determine if the given activity is to be allowed or disallowed.

Controlling access to documents and SQL search criteria is restricted to the high-level language interface and is not available to the assembler exit interface routine (ARSUSECX).

You must set up and enable the exits that are needed to implement the required level or type of security. When using any of the security exits, the user ID must be defined for both TSO and Content Manager OnDemand.

The following diagram presents an overview of the security system interface exits. This diagram can be used as a reference throughout the rest of the text.

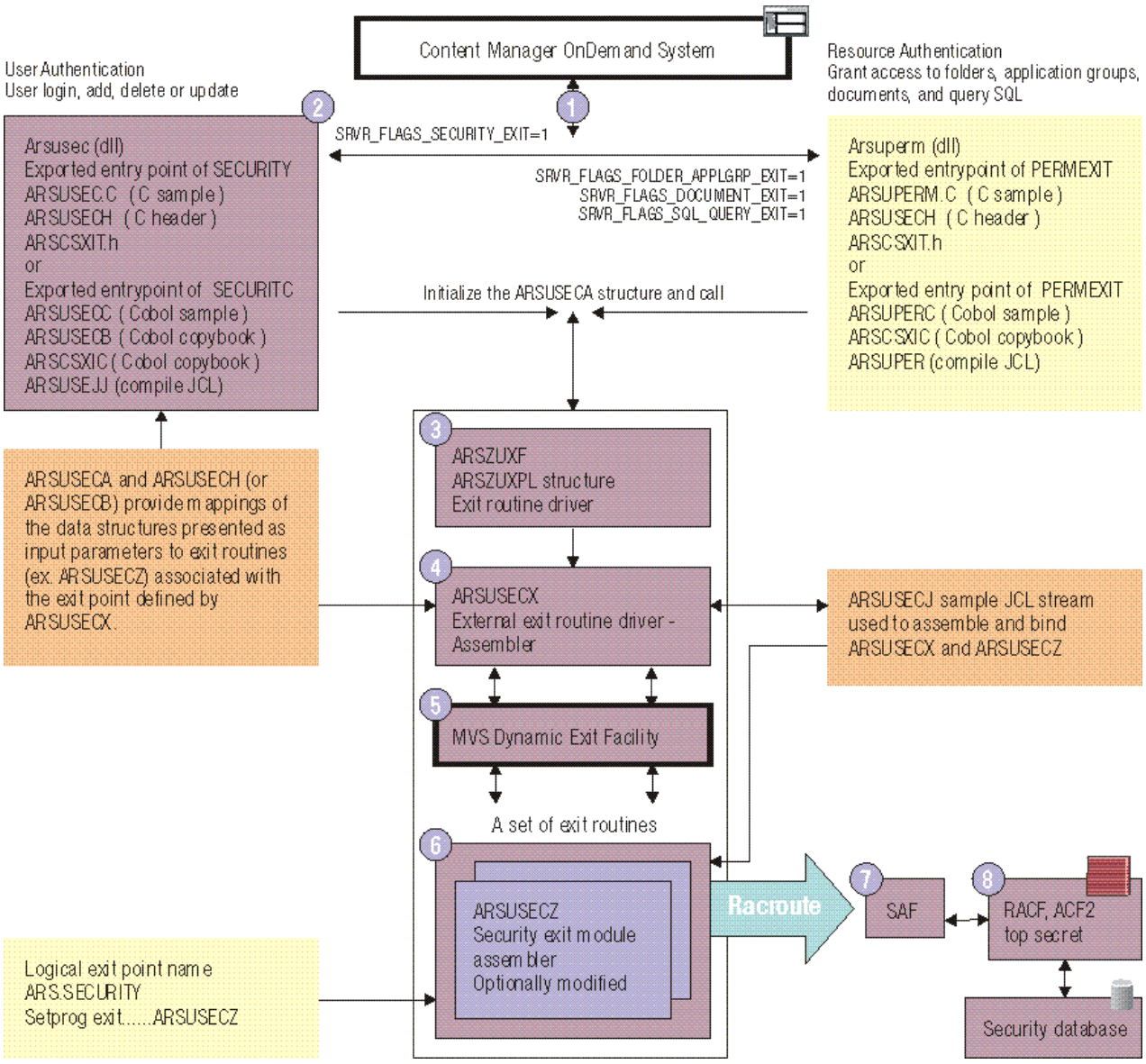

Figure 11. Security system interface exits

Setting one or more security parameters in the `ars.cfg` file (1) causes the Content Manager OnDemand server to call the corresponding security exit. For example, if the parameter SRVR_FLAGS_SECURITY_EXIT=1 was set, then the arsusec DLL (2) is loaded. The arsusec DLL then calls the ARSZUXF (3) exit driver, which calls the ARSUSECX (4) external exit routine driver.

ARSUSECX calls the MVS Dynamic Exit Facility (5), which calls one or more exit routines that are associated with the ARS.SECURITY exit point. In this example, ARSUSECX calls ARSUSECZ (6). ARSUSECZ issues a RACROUTE call to SAF (7), which forwards the request to the systems security system (in this example, RACF) (8). RACF allows or denies the request and the result is returned to the Content Manager OnDemand server through the same route.

The Content Manager OnDemand server expects one of several specific return codes from the arsusec or arsuperm DLL to determine how to process the security request. In most cases when you use SAF, the arsusec or arsuperm DLL follows the previously described process and calls SAF. However, calling SAF is not a requirement of the Content Manager OnDemand server and the arsusec or arsuperm DLL interface. The DLLs can immediately set an appropriate return code and return the response to the Content Manager OnDemand server without calling SAF through the MVS Dynamic Exit Facility.

Restriction: Controlling access to documents and SQL search criteria through the arsuperm DLL is restricted to the high-level language of the arsuperm DLL. You cannot call the SAF through the MVS Dynamic Exit Facility.

In order to avoid having to define all potential users to the Content Manager OnDemand system, the concept of a Content Manager OnDemand proxy user has been created. A Content Manager OnDemand proxy user is the ability to be authenticated based on your login user ID and password, however the external security system (for example RACF) can default your actual Content Manager OnDemand userid to something else. For instance, inside your Content Manager OnDemand security system you might have only 10 real sets of permissions for users. For example, ODUSER1, ODUSER2, ODUSER3 to ODUSER10. Although you might have 10,000 end users to the Content Manager OnDemand system, you do not want to define all 10,000 users to Content Manager OnDemand.

When Content Manager OnDemand users login using their RACF user ID and password, you can use the user proxy to assign them one of the defined users in Content Manager OnDemand (for example ODUSER1). The supplied sample ARSUSECX and ARSUSECZ commands used together implement this capability by leveraging the capability of the RACF CSDATA user segment. The ARSPROXY field is first defined in the CSDATA segment. For those RACF users that are to be assigned a Content Manager OnDemand proxy user, the ARSPROXY field is then assigned a value that is to be the proxy user. After the user has been authenticated using the RACF user and password, the ARSPROXY field is extracted, and if found, the value is returned as the proxy user. For example, the value ODUSER1 could be assigned to the ARSPROXY field for a given user or users, and ODUSER1 would then be used for Content Manager OnDemand permissions. If a user does not have an ARSPROXY field in the CSDATA segment, the RACF user ID is to be used as the Content Manager OnDemand user ID.

For more information on RACF custom fields, see the "Defining and using custom fields" section of the *z/OS Security Server RACF Administrators Guide*. Other external security products might have different procedures, or use different methods for handling custom fields. The **SARSINST(ARSUSECZ)** parameter might need to be customized to provide the proxy capability with other security products. For RACF, the ARSPROXY field should be defined as follows:

```
         RDEFINE CFIELD USER.CSDATA.ARSPROXY UACC(NONE) CFDEF(TYPE(CHAR)
MAXLENGTH(8) FIRST(ANY) OTHER(ANY) HELP('ONDEMAND PROXY, UP TO 8 CHARACTERS')
MIXED(NO) LISTHEAD('ONDEMAND PROXY ='))
```

For example, to assign a Content Manager OnDemand proxy ODUSER1 to the RACF USER RACF001:

```
ALTUSER RACF001 CSDATA(ARSPROXY(ODUSER1))
```

For example, to enable Content Manager OnDemand user proxy you also need to specify the following in the `ars.cfg` file:

```
ARSMVS_ARSUSEC_USERPROXY=1
```

In order to use the Content Manager OnDemand proxy facility, you must reassemble the ARSUSECX and ARSUSECZ command members of SARSINST, and associate the new ARSUSECZ with the ARS.SECURITY dynamic exit.

Remember: The new ARSUSECX and ARSUSECZ commands can be invoked by lower levels of Content Manager OnDemand, but the lower levels do not have the user proxy capability.

Interface exit components

The security system interface exit consists of the following components. The component groupings are for the two high-level language exit points and the low-level language (assembler) exit point. All of the components are provided in source form.

Components of the high-level language interfaces

There are two high level language interfaces. Both, either, or neither of them can be activated based on the settings in the `ars.ini` file. The following sample files are provided for each interface:

User Authentication
The executable file is the `arsusec` DLL, which is used for checking user security.

- ARSUSEC (c sample)
- ARSUSECH (c header file)

or

- ARSUSECC (sample COBOL program equivalent to the ARSUSEC C sample)
- ARSUSECB (COBOL copybook equivalent to the C ARSUSECH header file)
- ARSUSEJJ (sample JCL to compile and link the ARSUSECC sample)

Resource Authorization
The executable file is the `arsuperm` DLL, which is used for checking user permissions.

- ARSUPERM (c sample)
- ARSUSECH (c header file)

or

- ARSUPERC (sample COBOL program equivalent to the ARSUPERM C sample)
- ARSUPERJ (sample JCL to compile and link the ARSUPERC sample)

The sample modules ARSUSEC and ARSUPERM provide the high level interface from the Content Manager OnDemand system to the external exit routine driver, ARSUSECX.

Components of the low-level language interface

There is a single low level assembler language interface. It is activated based on the settings in the `ars.ini` file.

- Assembler Interface 1.ARSUSECX (Assembler external exit routine driver)
- ARSUSECZ (Assembler associated exit routine)
- ARSUSECA (Assembler header file)
- ARSUSECJ (JCL stream used to assemble and bind ARSUSECX and ARSUSECZ)

Module ARSUSECX interfaces with the MVS Dynamic Exit Facility to route control to a set of exit routines. Module ARSUSECZ is an example of such an exit routine.

ARSUSECA (assembler), ARSUSECH (C), and ARSUSECB (COBOL) provide mappings of the data structures presented to exit routines associated with the exit point defined by ARSUSECX. (That is, the data structure presented as an input parameter to ARSUSECZ.)

High level language interfaces

The high level language security interface is implemented by two DLLs:

- arsusec
- arsuperm

The arsusec DLL is called for user login, add, delete or update. The arsuperm DLL is called to grant access to folders, application groups, documents, and query SQL.

The samples that are shipped (ARSUSEC, ARSUPERM, ARSUSECC, and ARSUPERC) initialize the ARSUSECA structure and call the ARSUSECX ARS.SECURITY exit driver. The high level language interfaces provide control of all the security functions. The assembler language interface accessed through the ARS.SECURITY exit does not allow for control of document and SQL query security.

The arsusec and arsuperm DLL modules are written in the C or COBOL programming language. They are provided in both source and executable forms, with the source being provided mainly as an aid in understanding the entire security system exit interface. The C version of arsusec and arsuperm are

shipped in object form with the product and is located in the `exits` directory. If you modify ARSUPERM or ARSUSEC, they must be compiled with the IBM C/C++ compiler. For details on compiling and binding a C DLL, see http://www.ibm.com/systems/z/os/zos/bkserv/.

Note: Both the sample ARSUSEC and ARSUPERM code will try to call arsusecz regardless of whether there is a dynamic exit associated with ARS.SECURITY. If you do not have an exit associated, the request will be denied. If you want to change this behavior, you need to modify the provided sample code.

For information about the COBOL compiler, refer to the Enterprise COBOL for z/OS documentation at http://www.ibm.com/software/awdtools/cobol/zos/library/

Arsusec DLL

The arsusec DLL is called when a user attempts to log on to Content Manager OnDemand, or an attempt is made to add, change, or delete a password, and the security exit is enabled through a flag setting in the `ars.ini` file. The arsusec DLL must reside in the `exits` directory and must have the APF extended attribute turned on. The `exits` directory can be specified by setting the **ARS_USER_EXITS_DIR** parameter in the `ars.cfg` file. By default, the path to the exits directory is `/usr/lpp/ars/V10R5M0/bin/exits`.

The arsusec DLL must have an exported entry point of SECURITY or SECURITC. A sample to compile and link the ARSUSECC is provided in ARSUSEJJ in the SARSINST data set.

The following statement must exist in the ARS.INI file in order for the arsusec DLL to be invoked:
`SRVR_FLAGS_SECURITY_EXIT=1`

The arsusec DLL runs in a threaded environment. The exit must be thread-safe.

When compiling the ARSUSEC C sample, the following compiler options must be specified:

```
DEFINE(OS390)
ROSTRING
DLL
LONGNAME
RENT
XPLINK
```

When compiling the ARSUSECC COBOL sample, the following compiler options must be specified:

```
DLL,LIB
OBJ
RENT
TRUNC(OPT)
THREAD
PGMN(LM )
```

Format

Figure 12 on page 184 shows the C language `arsusec`.

```
#pragma export(SECURITY)
#include "arscsxit.h"
ArcCSXitSecurityRC SECURITY( char *act_userid,
                             char *cur_userid,
                             char *cur_passwd,
                             char *new_userid,
                             char *new_passwd,
                             ArcCSXitSecurityAction action,
                             char *msg,
                             char *clnt_id
                             char *instance,
                             char *passthru_text,
                             ArcU32 passthru_size,
                             ArcByte *passthru_buf
                            )
```

Figure 12. C language arsusec

The `ODServer.setPassThruToken()` method provides values for the following parameters:

- **passthru_text**
- **passthru_size**
- **passthru_buf**

Figure 13 on page 184 shows the COBOL language `arsusec`.

```
identification division.
 program-id.   "SECURITC" recursive.

linkage section.

copy arscsxic.

procedure division using
          by reference ArcCSXitSecurity-act-userid,
          by reference ArcCSXitSecurity-cur-userid,
          by reference ArcCSXitSecurity-cur-passwd,
          by reference ArcCSXitSecurity-new-userid,
          by reference ArcCSXitSecurity-new-passwd,
          by value     ArcCSXitSecurityAction,
          by reference ArcCSXitSecurity-msg,
          by reference ArcCSXitSecurity-clnt-id,
          by reference ArcCSXitSecurity-instance,
          by value     ArcCSXitSecurity-passthru-text,
          by value     ArcCSXitSecurity-passthru-size,
          by value     ArcCSXitSecurity-passthru-buf.
```

Figure 13. COBOL language arsusec

General description

On entry, review the values of *action* or *ArcCSXitSecurityAction* to determine the type of action being performed:

Login

ARCCSXIT_SECURITY_USER_LOGIN (C)
arccsxit-security-user-login (COBOL)

Add

ARCCSXIT_SECURITY_USER_ADD (C)
arccsxit-security-user-add (COBOL)

Delete

ARCCSXIT_SECURITY_USER_DELETE (C)
arccsxit-security-user-delete (COBOL)

Update

ARCCSXIT_SECURITY_USER_UPDATE (C)

Valid input parameters for the login action

The following list describes the valid input parameters for the login action. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters.

cur_userid
ArcCSXitSecurity-cur-userid
Null delimited user ID attempting to log on.

cur_passwd
ArcCSXitSecurity-cur-passwd
Null delimited password for the user attempting to log on.

clnt_id
ArcCSXitSecurity-clnt-id
Null delimited host name and IP address of the requester, if available.

new_userid
ArcCSXitSecurity-new-userid
Output proxy user ID. Contains the null delimited Content Manager OnDemand user ID to be used for this user. The maximum is 129 characters including the trailing nul. If an empty string is returned, the **cur_userid** is used.

passthru_text
ArcCSXitSecurity-passthru-text
Null delimited string that is the EBCDIC logon passthru text.

passthru_size
ArcCSXitSecurity-passthru-size
Length of the area pointed to by **passthru_buf**.

passthru_buf
ArcCSXitSecurity-passthru-buf
Address of an area that is the binary logon passthru buffer.

Valid input parameters for the add action

The following list describes the valid input parameters for the login action. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters.

act_userid
ArcCSXitSecurity-act-userid
Null delimited user ID performing the add.

new_userid
ArcCSXitSecurity-new-userid
Null delimited user ID being added.

new_passwd
ArcCSXitSecurity-new-passwd
Null delimited password for the user being added.

clnt_id
ArcCSXitSecurity-clnt-id
Null delimited host name and IP address of the requester, if available.

Valid input parameters for the delete action

The following list describes the valid input parameters for the delete action. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters.

act_userid
ArcCSXitSecurity-act-userid
Null delimited user ID performing the delete.

cur_userid
ArcCSXitSecurity-cur-userid
Null delimited user ID being deleted.

clnt_id
ArcCSXitSecurity-clnt-id
Null delimited host name and IP address of the requester, if available.

Valid input parameters for the update action
The following list describes the valid input parameters for the update action. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters.

act_userid
ArcCSXitSecurity-act-userid
Null delimited user ID performing the update.

cur_userid
ArcCSXitSecurity-cur-userid
Null delimited user ID being changed.

cur_passwd
ArcCSXitSecurity-cur-passwd
Null delimited current password for the user being changed.

new_userid
ArcCSXitSecurity-new-userid
Null delimited new user ID.

new_passwd
ArcCSXitSecurity-new-passwd
Null delimited new password

Note: The address of this is NULL if the password is not being changed.

clnt_id
ArcCSXitSecurity-clnt-id
Null delimited host name and ip address of the requester, if available.

Returned values

The following list describes the return codes that can be specified.

0
Allow the action

1
Deny the action

2
(Login only) Allow the log on but require the user to change the password.

3
Exit failed

4
(Login only) The exit executed successfully. However Content Manager OnDemandshould still verify the password to allow or deny login.

The following symbols for the return code are defined for C in ARSCSXIT:

0
ARCCSXIT_SECURITY_RC_OKAY

1
ARCCSXIT_SECURITY_RC_PERMS

2
ARCCSXIT_SECURITY_RC_PASSWD_CHNG

3
ARCCSXIT_SECURITY_RC_FAILED

4
ARCCSXIT_SECURITY_RC_OKAY_BUT_VALIDATE_IN_OD

For any non-zero return code, the exit can place a null delimited string in **msg** or **ArcCSXitSecurity-msg**. This string will be displayed on the client in place of the client's normal failure message. This string (including the terminating X'00') should not exceed **ARCCSXIT_MAX_SRVR_MESSAGE_SIZE** (C) or the length of **ArcCSXitSecurity-msg** (COBOL) bytes.

Arsuperm DLL

The arsuperm DLL is called when a user attempts to access a folder, application group, or document, or perform an SQL query, and the security exit is enabled through a flag setting in the `ars.ini` file. The arsuperm DLL must be stored in the `/usr/lpp/ars/V10R5M0/bin/exits` directory and must have the APF extended attribute turned on. This DLL must have an exported entry point of PERMEXIT.

The arsuperm DLL is called if one of the following statements exists in the `ARS.INI` file:

- `SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1` (for folder and application group permission checking)
- `SRVR_FLAGS_DOCUMENT_EXIT=1` (for document permission checking)
- `SRVR_FLAGS_SQL_QUERY_EXIT=1` (for SQL query processing)

Note: Enabling document permission checking or query processing might decrease Content Manager OnDemand performance when performing a document query.

The arsuperm DLL runs in a threaded environment. The exit must be thread-safe.

When compiling the C ARSUPERM sample, the following compiler options must be specified:

DEFINE(OS390)
ROSTRING
DLL
LONGNAME
RENT
XPLINK

When compiling the ARSUPERC (COBOL) sample, the following compiler options must be specified:

DLL
LIB
OBJ
RENT
TRUNC(OPT)
THREAD
PGMN(LM)

Format

Figure 14 on page 187 shows the C language arsuperm.

```
#pragma export(PERMEXIT)
#include "arscsxit.h"

int PERMEXIT( char userid, ArcCSXitPermExit perm, int access )
```

Figure 14. C language arsuperm

Figure 15 on page 188 shows the COBOL language arsuperc.

```
identification division.
program-id.    "PERMEXIT" recursive.

linkage section.

copy arscsxic.

procedure division using
                by reference ArcCSXitPermExit-userid,
                by reference ArcCSXitPermExit,
                by reference ArcCSXitPermExit-access.
```

Figure 15. COBOL language arsuperc

A sample to compile and link the COBOL source ARSUPERC is provided in ARSUPERJ in the SARSINST data set.

General Description

Folder permission parameters

userid or **ArcCSXitPermExit-userid** contains the null delimited user ID for whom access permission is checked. The ArcCSXitPermExit structure contains information that provides details about the object being checked. The perm->action or ArcCSXitPermExit-action field should be interrogated to determine the type of object whose preadmissions are being checked:

Folder

1(C)
ArcCSXitPermExit-act-fol (COBOL)

Application Group

2 (C)
ArcCSXitPermExit-act-ag (COBOL)

Document

3(C)
ArcCSXitPermExit-act-doc (COBOL)

SQL query

4 (C)
ArcCSXitPermExit-act-sql (COBOL)

The following list describes the folder permission parameters. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters.

perm->u.folder_perm.folder_name
ArcCSXitPermExit-fol-namep
Contains a pointer to the null delimited folder name.

Application group permission parameters

The following list describes the application group permission parameters. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters.

perm->u.appl_grp_perm.appl_grp.name
ArcCSXitPermExit-ag-namep
Pointer to the null delimited application group name.

perm->u.appl_grp_perm.appl_grp.agid
ArcCSXitPermExit-ag-agid
The application group identifier.

perm->u.appl_grp_perm.appl_grp.agid_name
ArcCSXitPermExit-ag-agid-namep
Pointer to the null delimited internal application group identifier.

Document permission parameters

The following list describes the document permission parameters. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters.

perm->u.doc_perm.appl_grp.name
ArcCSXitPermExit-doc-ag-namep
Pointer to the null delimited application group name.

perm->u.doc_perm.appl_grp.agid
ArcCSXitPermExit-doc-agid
The application group identifier.

perm->u.doc_perm.appl_grp.agid_name
ArcCSXitPermExit-doc-agid-namp
Pointer to the null delimited internal application group identifier.

perm->u.doc_perm.doc.doc_flds.flds_num
ArcCSXitPermExit-doc-flds-num
Number of entries in the index field descriptor table.

perm->u.doc_perm.doc.doc_flds.flds
ArcCSXitPermExit-doc-flds
Address of the index field descriptor table for this document.

perm->u.doc_perm.doc.doc_hndl.name
ArcCSXitPermExit-doc-name
Internal document name for this document (for example, 23FAAC).

perm->u.doc_perm.doc.doc_hndl.doc_off
ArcCSXitPermExit-doc-doc-off
Offset of the document within the decompressed object.

perm->u.doc_perm.doc.doc_hndl.doc_len
ArcCSXitPermExit-doc-doc-len
Length of the document within the decompressed object.

perm->u.doc_perm.doc.doc_hndl.comp_off
ArcCSXitPermExit-doc-comp-off
Offset of the compressed object within the object.

perm->u.doc_perm.doc.doc_hndl.comp_len
ArcCSXitPermExit-doc-comp-len
Length of the compressed object within the object.

SQL query permission parameters

The following list describes the SQUL query permission parameters. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters.

perm->u.sql_query_perm.appl_grp.name
ArcCSXitPermExit-sql-name
Pointer to the null delimited application group name.

perm->u.sql_query_perm.appl_grp.agid
ArcCSXitPermExit-sql-agid
The application group identifier.

perm->u.sql_query_perm.appl_grp.agid_name
ArcCSXitPermExit-sql-agid-name
Pointer to the null delimited internal application group identifier.

perm->u.sql_query_perm.in_sql
ArcCSXitPermExit-sql-pin-sql
Pointer to the SQL query string that is being used for the search.

perm->u.sql_query_perm.in_sql_r
ArcCSXitPermExit-sql-pin-sql-r
Pointer to the SQL query restriction string that is being used for the search.

For document permissions, a pointer to the index field descriptor table is provided. The index field descriptor table contains one entry for each type of index value associated with the document. It contains the database column name, an indication of the type of value, and the value for the index entry. This field is mapped by ArcCSXitField of the ARSCSXIT (C) or ARSCSXIC (COBOL) members of SARSINST.

Returned values

For the folder permissions check, `*access` or `ArcCSXitPermExit-access` can be set to any of the following return codes:

0
Deny access to the folder

1
Allow *PUBLIC access

2
Grant access to the primary folder

3
Grant access to the secondary folder

Important: If the return code is a value other than zero and there is a X'08' value with a Boolean OR operator, the user will have full report browse permission on the folder.

For the application group permission check, `*access` or `ArcCSXitPermExit-access` can be set to any of the following return codes:

0
Deny access to the application group

1
Allow *PUBLIC access

2
Grant access to the application group

For the document permission check, `*access` or `ArcCSXitPermExit-access` can be set to any of the following return codes:

0
Deny access to the document

1
Grant access to the document

For the SQL query permission, if you want to change the SQL query string, set `perm->u.sql_query_perm.out_sql` or `ArcCSXitPermExit-sql-pout-sql` to the address of a different SQL string to be used for the query. This string must be allocated using the C `malloc()` or `calloc()` functions.

To change the query restriction string, set `perm->u.sql_query_perm.out_sql-r` or `ArcCSXitPermExit-sql-pout-sqlr` to the address of a different SQL query restriction string to be used for the query. This string must be allocated using the C `malloc()` or `calloc()` functions.

The following sample code shows how to call the `malloc()` function from COBOL program:

```
     local-storage section

01 malloc-bytes pic s9(9) binary.
01 my-ptr                   pointer.

linkage section

procedure division.

    move 100 to malloc-bytes
    call "malloc" using by value malloc-bytes
                  returning my-ptr.
```

Debugging DLL Load failure

About this task

If an error occurs when Content Manager OnDemand attempts to load a DLL and get the entry point for the function in the DLL, an ARS0160E message is issued. The RC= portion of the message provides the system **errno** value resulting from the `dlopen()` or `dlsym()` call that failed.

Additional information for diagnosing DLL failures can be obtained by using the **_EDC_DLL_DIAG** environment variable. The **_EDC_DLL_DIAG** variable can be set in either the `ars.cfg` file, or through the Language Environment ENVAR runtime option. Additional information about the **_EDC_DLL_DIAG** variable can be found in the *z/OS C/C++ Programming Guide*.

Assembler language interfaces

The ARSUSECX and ARSUSECZ exit routines are written in S/390® assembly language and require the use of the IBM High Level Assembler for proper assembly. The ARSUSECJ sample JCL stream is used to assemble and bind the ARSUSECX and ARSUSECZ exit routines.

Use of the MVS™ dynamic exit facility

The ARSUSECX exit routine causes the MVS Dynamic Exit Facility to call any programs that are associated with the ARS.SECURITY logical exit point name. The ARSUSECX module interfaces with the MVS Dynamic Exit Facility to:

- Define the logical exit point name: ARS.SECURITY
- Route control to a set of associated exit routines and process the results of their execution.

Module ARSUSECZ is implemented as an associated dynamic exit routine.

An exit routine is eligible for execution once it has become associated with the logical exit point. The MVS Dynamic Exit Facility provides several methods for performing this association. For example, the following operator command can be used to associate ARSUSECZ with the logical exit point name ARS.SECURITY: `SETPROG EXIT,ADD,EXITNAME=ARS.SECURITY,MODNAME=ARSUSECZ`. (The example assumes that ARSUSECZ can be found in the LPA or a LNKLST data set.)

Note: Whether you enable the high level or low level exits, at least one module (such as ARSUSECZ) needs to be associated with the logical exit point name ARS.SECURITY. Multiple modules can also be associated with the single logical exit point ARS.SECURITY. When multiple exit routines are associated with the ARS.SECURITY logical exit point, the routines in the set are called one at time, conditionally. If a given routine denies access, then the subsequent exit routines are not called.

See the following IBM publications for more information about the MVS Dynamic Exit Facility:

- *z/OS MVS Programming: Authorized Assembler Services Guide*
- *MVS Programming: Authorized Assembler Services Reference*, Volumes 1 – 4
- *z/OS MVS Initialization and Tuning Reference*
- *z/OS MVS System Commands*

Enabling the security system interface exit

About this task

The enablement or disablement of the security system interface exit is controlled by settings in the `ARS.INI` file, located in the `/usr/lpp/ars/V10R5M0/config` directory.

To enable the exit for the processing of the following events, you must include the `SRVR_FLAGS_SECURITY_EXIT=1` statement in the `ARS.INI` file:

- Logon
- Password changing
- The addition or deletion of a user ID by using the Content Manager OnDemand administrative functions

To enable the exit for the processing of the following events, you must include the `SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1` statement in the `ARS.INI` file:

- Access to a Content Manager OnDemand folder
- Access to a Content Manager OnDemand application group

To disable a group of exit functions, either omit the corresponding statement or set the statement operand to 0.

Note: To restrict document access or control the SQL search criteria that is used for searching folders, you must use the High Level language Interface (arsuperm).

Content Manager OnDemand SAF resource classes

The following Resource Classes must be defined to the SAF conforming security system to protect Content Manager OnDemand resources:

Resource Class Name
ARS1FLDR

Protects
Content Manager OnDemand folders

Minimum Authority for Access
Read

Resource Name
The folder name string

Maximum Resource Name Length
60 characters

Mixed Case Resource Names
Yes

Resource Class Name
ARS1APGP

Protects
Content Manager OnDemand application groups

Minimum Authority for Access
Read

Resource Name
The application group name string

Maximum Resource Name Length
60 characters

Mixed Case Resource Names
Yes

For the naming rules associated with Content Manager OnDemand folders and application groups, see the *IBM Content Manager OnDemand for z/OS: Introduction and Planning Guide*.

Defining Content Manager OnDemand resource classes and entities to RACF

About this task

RACF requires entries in its Resource Class Descriptor Table corresponding to the Content Manager OnDemand resources to be protected. Additionally, entries need to exist in the RACF Router Table for each entry placed in the Resource Class Descriptor Table.

For details regarding the definition and protection of general resources via RACF, see the following publications:

- *IBM SecureWay Security Server for z/OS and OS/390 RACF: Command Language Reference*
- *IBM SecureWay Security Server for z/OS and OS/390 RACF: Macros and Interfaces*
- *IBM SecureWay Security Server for z/OS and OS/390 RACF: Security Administrator's Guide*
- *IBM SecureWay Security Server for z/OS and OS/390 RACF: System Programmer's Guide*

General resource classes are defined to RACF through the assembly and link-editing of the RACF Class Descriptor Table. Table entries are created through the invocation of the ICHERCDE macro.

The following illustrates the ICHERCDE macro invocations required to define the Content Manager OnDemand resource classes to RACF.

```
ARS1FLDR ICHERCDE CLASS=ARS1FLDR,  OnDemand V7 Folder            +
               CASE=ASIS,          See note (1) below            +
               ID=135,             See note (2) below            +
               POSIT=37,           See note (2) below            +
               MAXLNTH=60,                                       +
               FIRST=ANY,                                        +
               OTHER=ANY,                                        +
               OPER=YES,                                         +
               DFTUACC=NONE,                                     +
               RACLIST=ALLOWED,                                  +
               GENLIST=ALLOWED
*
ARS1APGP ICHERCDE CLASS=ARS1APGP,  OnDemand V7 Application Group +
               CASE=ASIS,          See note (1) below            +
               ID=136,             See note (2) below            +
               POSIT=37,           See note (2) below            +
               MAXLNTH=60,                                       +
               FIRST=ANY,                                        +
               OTHER=ANY,                                        +
               OPER=YES,                                         +
               DFTUACC=NONE,                                     +
               RACLIST=ALLOWED,                                  +
               GENLIST=ALLOWED
*
         ICHERCDE ,
```

Note: The `ID` and `POSIT` values as shown are only for illustration purposes. The actual values selected might be different and are a function of other general resource classes which might already be defined.

The RACF Router Table must also be assembled and link-edited. Table entries are created through the invocation of the ICHRFRTB macro.

The following illustrates the ICHRFRTB macro invocations required to define the RACF Router Table entries corresponding to the Content Manager OnDemand Resource Classes.

```
ICHRFR01 CSECT ,
ARS1FLDR ICHRFRTB CLASS=ARS1FLDR,  OnDemand V7 Folder            +
```

```
               ACTION=RACF
*
ARS1APGP ICHRFRTB CLASS=ARS1APGP,  OnDemand V7 Application Group +
               ACTION=RACF
*
ENDTAB   ICHRFRTB TYPE=END
*
         END   ,
```

In Content Manager OnDemand, it is possible to create folder and application group name strings which contain embedded blanks. However, RACF resource names cannot contain blank characters. As a consequence, module ARSUSECZ translates all embedded blanks in these name strings to the underscore character (X'6D') when they are presented to the security system.

To protect Content Manager OnDemand resources whose names contain embedded blanks, you must define RACF profiles that will match an underscore (either explicitly or through generics) in place of the blank characters.

Note: The security exit routine is not called for a report if the ACT SECURITY EXIT field contains the word Public.

Overview of the operation of ARSUSECX

ARSUSECX iteratively invokes the MVS Dynamic Exit Facility to route control to each of the exit routines associated with the ARS.SECURITY logical exit point. If there are no exit routines associated with the exit point, ARSUSECX indicates that the event for which it was invoked is not to be allowed to continue.

For folder permission, if the exit returns ARSUSECA_RCNORM or ARSUSECA_RCPUBLC, the exit might additionally set ARSUSECA_PERMFRB to a non-zero value to indicate that user has full report browse permission.

The execution environment at entry to an exit routine is as follows:

Dispatchable Unit Mode
Task

Cross Memory Mode
PASN=HASN=SASN

Addressing Mode
31-bit

ASC Mode
Primary

Interrupt Status
Fully enabled

Locks
No locks are held

Authorization
Problem State and APF Authorized

Control parameters
All parameters reside in the Primary SpaceAll addresses are pure 31-bit values

General Registers

R1
Points to a fullword which, in turn, points to the interface parameter list structure as mapped by the ARSUSECA DSECT.

R13
Points to a standard 18-word general register save area.

R14
Contains the address to which the exit routine is to branch upon completion, that is, the Return Address.

R15
Contains the entry point address of the exit routine.

Upon completion, an exit routine must return to the address contained in R14 at entry, with the general registers loaded as follows:

R0-R14
The same as at entry to the exit routine.

R15
Contains a return code.

Field ARSUSECA_ACTION within the ARSUSECA structure indicates the event or activity for which the exit routine is being invoked. The following list describes the values that might be contained in the ARSUSECA_ACTION field

ARSUSECA_ACLOGIN
User logon processing

ARSUSECA_ACUSRADD
Administration function: Adding a new user ID

ARSUSECA_ACUSRDEL
Administration function: Deleting a user ID

ARSUSECA_ACUSRUPD
Administration function: Changing a user ID and associated password

ARSUSECA_ACAFOLD
Access to a Content Manager OnDemand folder

ARSUSECA_ACAAPGP
Access to a Content Manager OnDemand application group

These values are defined within the ARSUSECA DSECT as well as within the ARSUSECH C structure mapping.

The return code values from an exit routine are shown in the following list:

ARSUSECA_RCNORM
Normal completion. The event or activity is to be allowed to continue.

ARSUSECA_RCPERMS
Access denied, that is, the event or activity is not to be allowed to proceed.

ARSUSECA_RCPWCHG
The user ID and password are valid, but the password must be changed.

ARSUSECA_RCFAIL
The security exit has failed. Access is denied.

ARSUSECA_RCPUBLC
Grant *PUBLIC access. This is valid for ARSUSECA_ACAFOLD and ARSUSECA_ACAAPGP only, and is valid only if ARSUSECA_VERSION is 4 or higher.

ARSUSECA_RCREVAL
The user ID and password are valid, but must be revalidated by Content Manager OnDemand.

The symbols are defined within the ARSUSECA DSECT as well as within the ARSUSECH C structure mapping.

If an exit routine ABENDs or returns an invalid (that is, undefined) return code, ARSUSECX regards the situation as though the exit routine had returned the ARSUSECA_RCPERMS (Permission Denied) return code.

If an exit routine returns a valid return code other than ARSUSECA_RCNORM, no additional associated exit routines are invoked.

Overview of the operation of ARSUSECZ

ARSUSECZ processes the following events and activities:

Logon
Logon events are processed in the conventional way. That is, the user ID and password are presented to the SAF conforming security system for verification, and, if valid, access is granted. Otherwise, access is denied.

Adding a new user ID
If access is granted, and an ARSPROXY field can be extracted from the RACF CSDATA segment, the associated value is returned as the new user.

It is expected that the user ID has been defined to the SAF conforming security system before being defined to the Content Manager OnDemand system. If this is the case, the addition event is allowed to proceed. Otherwise, the addition event is disallowed.

Deleting a user ID
Deletion events are always allowed to proceed.

Changing a user ID and password
Although the Content Manager OnDemand system allows renaming of an existing user ID, most SAF-conforming security systems do not. Therefore, ARSUSECZ does not allow renaming of existing user IDs.

ARSUSECZ does allow changing the password of an existing user ID. If ARSUSECZ receives a password that is longer than 8 characters, ARSUSECZ includes the **PHRASE** parameter when it invokes the RACROUTE REQUEST=VERIFY macro to indicate that it is providing a password phrase. For information on enabling the use of password phrases, see "Enabling support for password phrases" on page 196.

Access to a Content Manager OnDemand folder or application group
A RACROUTE AUTH request is issued to determine if the requesting user ID is to be granted read access to the folder or application group.

Access is granted for the following situations:

- The SAF conforming security system has granted access.
- The SAF conforming security system has not made a decision. This can occur, for example, when the resource class is not defined to the security system or when no profile exists for the named entity.

Otherwise, access is denied.

Review the supplied source code for details on the operation of ARSUSECX and ARSUSECZ.

Enabling support for password phrases

About this task

To enable support for password phrases, you must make sure that the server does not convert passwords to uppercase before passing them to ARSUSECZ.

Procedure

To prevent the server from converting passwords to uppercase, do the following steps:

1. Check that your SAF-conforming security system supports case sensitive passwords.

   For RACF, the RCVTPLC flag enables support for case sensitive passwords. Set this flag to ON. If you do not set the RCVTPLC flag to ON, when ARSUSECZ receives a password that is eight characters or less, ARSUSECZ converts the password to uppercase.

   If your SAF-conforming security system does not support case sensitive passwords, when you do step "3" on page 197, modify ARSUSECZ so that it does not convert passwords to uppercase.
2. Check that the instance supports case sensitive passwords by doing the following steps:
   a) Start the Content Manager OnDemand Administrator client and log in to the Content Manager OnDemand instance.

b) Right-click the server name and select **System Properties**.

c) In the**Login Processing** box, if the**Password Case Sensitive** check box is checked, go to the next step. If it is not checked, check it and notify users with existing user IDs that they must now enter their password in uppercase and that the next time they modify their password, it will be case sensitive.

3. Rebuild the ARSUSECZ user exit by following the instructions in the supplied source code. Remember that if your SAF-conforming security system does not support case sensitive passwords, add logic to ARSUSECZ so that it does not convert passwords to uppercase.

After enabling ARSUSECZ, permissions provided by *PUBLIC group no longer honored

Symptoms
You have used the product groups and permissions to allow various groups to see folders and application groups. You then enable the ARSUSECZ exit to check the permission in RACF. At this point the permissions that are provided by the *PUBLIC group are no longer honored.

Causes
The *PUBLIC group that is defined in Content Manager OnDemand is not checked through the SAF interface. The sample exits that are shipped with the product do not use that group; thus, any permissions that are granted through that group are lost unless the samples are changed.

Resolving the problem
Change the sample exit to use the *PUBLIC group for permission checks, and then the permissions should more closely match what are in the definitions. You can change the return codes from the ARSUSECZ exit to let Content Manager OnDemand check permissions in the *PUBLIC group. In the "folder access" and "application group access" routines, change the **ARSUSECA_RCPERMS** parameter to return ARSUSECA_RCPUBLC where permissions are defined.

Programming Interface Specifications for module ARSZUXF and ARSZUXFX

Module ARSZUXF loads the ARSUSECX module and handles the construction, caching, and deletion of security system objects (ACEE and related structures). Module ARSZUXF offers the following services:

- Executable module fetch service for modules that resides in conventional MVS data sets as well as in the HFS. Language Environment function descriptors are constructed for fetched modules. After a module is fetched, subsequent requests for a given module are satisfied by using the existing instance of that module.
- The construction, caching, and deletion of security system objects (ACEE and related structures).

For C programs compiled with the XPLINK compiler option, use the module ARSZUXFX. The module provides the `arszuxfx_acee()` function, which can help you construct, cache, and delete security system objects (ACEE and related structures). If you want to fetch modules, use the Language Environment CEEPLOD2 service.

The ARSZUXPL structure is the principal input to ARSZUXF. This structure and any areas that are referenced by pointers that are contained within this structure might reside anywhere within the 31-bit addressable Private Area of the Primary Space. The ARSZUXPL structure and, in general, all areas that are referenced by pointers that are contained within the ARSZUXPL structure, must be Read/Write accessible in the PSW key of the caller.

The ARSZUXPL structure is mapped in several programming languages:

Assembler
Macro ARSZUXP produces the ARSZUXPL mapping DSECT.

C
The ARSZUXPL structure definition is contained in the ARSUSECH header file. This header file also contains the ARSZUXF function prototype definition.

COBOL
The ARSZUXPL structure definition is contained in the ARSUSECB copy book.

To use the services provided by ARSZUXF, the caller must establish the appropriate values within the ARSZUXPL structure. Field usage (that is, an input item, an output item, or an item which might be both input and output) is identified by comments within the various structure mappings.

Examples of ARSZUXF usage are contained in source parts that are supplied with Content Manager OnDemand:

Table 9. Source parts that supplied containing ARSZUXF examples

Part name	Programming languages
ARSUSEC	C
ARSUSECC	COBOL
ARSUPERM	C
ARSUPERC	COBOL

Module fetch service

The name of the module to be fetched is provided in field ARSZUXPL_MODNAME. If the fetch is successful, the function pointer is returned in field ARSZUXPL_MODFCNP.

The search order for modules is the conventional MVS search order that is followed by a search of the HFS. The HFS path search order is described in *z/OS Language Environment Vendor Interfaces*.

All invocations of the fetched module, regardless of the programming language that is used, should be performed by dereferencing the function pointer that is contained in field ARSZUXPL_MODFCNP. For example, in assembler, the following instruction sequence should be used to call a fetched module:

```
L      R15,ARSZUXPL_MODFCNP
BASR   R14,R15
```

Managing the security system object caching service

About this task

Use a set of calls of ARSZUXF to manage and access the security system object data.

To access the security system object data, set the ARSZUXPL_FCN field to the ARSZUXPL_FCNLACEE value and provide in the ARSZUXPL_USERID field the user ID for which security system object data is obtained.

If field ARSZUXPL_SAFLOG is non-zero, then the SAF services that are invoked by ARSZUXF are performed requesting security system logging and messaging. Otherwise, security system logging and messaging are disabled.

A cached security system object for a given user ID is used, and the corresponding ACEE pointer is returned to the caller when the cached object is not older than *n* seconds. Otherwise, a new security system object is created. The age limit (in seconds) might be supplied in field ARSZUXPL_AGELIM. If you specify a zero value, a default value of 45 seconds is used. The effective value, whether explicitly specified or defaulted, applies only when the security system object is initially created and cached. A caller specified age limit value has no effect when a cached security system object is used to satisfy the request.

For a successful ARSZUXPL_FCNLACEE request, ARSZUXF returns a 64-bit token value in field ARSZUXPL_ACEETKN. This value represents the registered interest in the security system object data.

If the caller no longer has an interest in the security system object data, make sure that you perform a corresponding ARSZUXF call. Place the token value that is returned from a prior ARSZUXPL_FCNLACEE

request into field ARSZUXPL_ACEETKN and set the ARSZUXPL_FCN field to the ARSZUXPL_FCNRACEE value.

Important:

- A cached security system object is not destroyed as long as there is a registered interest in the object.
- A caller should use the security system object data, for example, the returned ACEE pointer, for a relatively brief period of time. A reasonable upper bound on this time period is 3 minutes.

Execution environment requirements

You must invoke ARSZUXF within the context of a valid Language Environment enclave.

General execution environment requirements include the following:

Cross Memory Mode
PASN=SASN=HASH

Locks
No locks might be held

Addressing Mode
31-bit

ASC Mode
Primary

PSW Key
Non-system

PSW State
Problem State

APF Authorization
APF Authorization is required

Make sure that you call ARSZUXF by using OS linkage; XPLINK linkage is not supported.

Register requirements

If you call from an assembler language module, the following register usage requirements must be satisfied:

Input GPRs

R1
Must contain the address of a fullword pointer. The pointer contains the address of the ARSZUXPL structure.

R12
Must contain the address of the active Language Environment Common Anchor Area (CEECAA).

R13
Must contain the address of a valid Language Environment Dynamic Storage Area (CEEDSA).

R14
Must contain the address to which ARSZUXF branches upon completion, that is, the return address.

Input ARs

AR0-AR15
No requirement.

Output GPRs

R0-R14:
Unchanged.

R15:
Request processing status codes. These status codes are defined by equates that are contained within the ARSZUXPL DSECT. The return code is contained in the low order 16 bits of this register. The reason code is contained in the left adjacent 16 bits of this register, that is, the high order 16 bits of the 32-bit GPR.

Output ARs

AR0-AR14:
Unchanged.

AR15:
Unpredictable.

ABEND codes

ARSZUXF itself does not intentionally terminate a request via ABEND. However, if you attempt to call functions through function pointers that are returned by ARSZUXF, the following ABENDs might be produced.

ABEND U0039, Reason Code x'0601'

ABEND U0039, Reason Code x'0601'

Explanation:
A recursion into the Intercepted Function trap procedure occurred.

System action:

User response:
Contact IBM support. Make sure that you have a machine readable binary dump, that is, a SYSMDUMP or SVC Dump of the environment at the time of the failure.

ABEND U0039, Reason Code x'0602'

ABEND U0039, Reason Code x'0602'

Explanation:
An attempt to invoke a module fetched by a terminated thread was intercepted.

User response:
This is most likely a user programming error. You cannot dereference a function pointer that was constructed by a thread that terminated.

ARS.INI file security settings summary

SRVR_FLAGS_FORCE_SECURITY=1
Causes the Content Manager OnDemand security user exit to be called to check folder or application group permissions for user client functions even if the normal Content Manager OnDemand permissions grants permission to the user. For example, Content Manager OnDemand normally calls the permission exit for a system administrator. Setting SRVR_FLAGS_FORCE_SECURITY=1 causes the exit to be called if the system administrator attempts to open a folder for viewing or uses the ARSDOC get function. This setting has no effect when you use the administrative client or the **ARSXML** command.

SRVR_FLAGS_SECURITY_EXIT=1
Specifies that the Content Manager OnDemand security user exit arsusec DLL will be called. Augments security-related processing for Logon, Change Password, Add User ID, or Delete User ID.

SRVR_FLAGS_DOCUMENT_EXIT=1
Enables document permission checking by calling the security user exit arsuperm DLL to check document permissions. Enabling this setting can greatly decrease Content Manager OnDemand performance when performing a document query.

SRVR_FLAGS_FOLDER_APPLGRP_EXIT=1
Specifies that the Content Manager OnDemand security user exit arsuperm DLL will be called to check folder and application group permissions.

SRVR_FLAGS_SQL_QUERY_EXIT=1
Specifies whether the Content Manager OnDemand security user exit arsuperm DLL will be called for SQL query processing.

Chapter 51. Unified login (ARS.PTGN) exit

Overview

The Content Manager OnDemand unified login exit (ARS.PTGN) enables a user to run the Content Manager OnDemand command line utilities (such as ARSLOAD) without specifying a user ID and password. This facility to log on without specifying a password uses the ability to specify a PassTicket as a password when using a RACROUTE REQUEST=VERIFY call. The following figure shows an overview of the unified login exit.

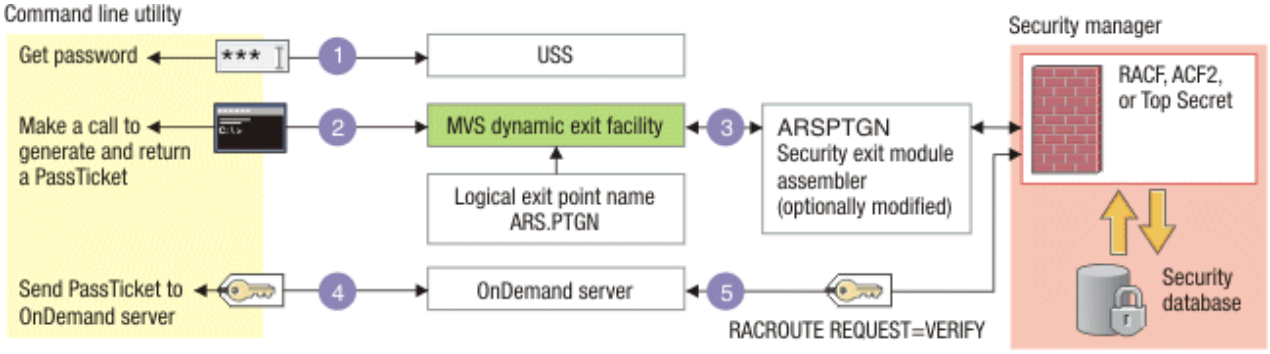

Figure 16. Overview of the unified login exit

The unified login exit is implemented as follows:

1. When a user runs a command line utility, if the login exit is enabled and the user does not specify a Content Manager OnDemand user ID and password, the current user ID that is returned by UNIX System Services (USS) is used to logon to Content Manager OnDemand.
2. Because the function to generate a PassTicket is not a part of the SAF interface, Content Manager OnDemand implements the call to generate the PassTicket as an MVS dynamic exit. The dynamic exit facility calls the ARS.PTGN login exit routine. Content Manager OnDemand provides a sample exit, ARSPTGN.
3. The ARSPTGN sample exit generates a PassTicket in the RACF environment and returns the PassTicket to the command line utility. Installations that use another external security product need to evaluate the supplied exit and possibly modify the exit for their environment.
4. The utility sends the PassTicket to the server when the utility attempts to logon to Content Manager OnDemand.
5. The PassTicket is used by the Content Manager OnDemand server to do a SAF RACROUTE REQUEST=VERIFY call from RACF for the user ID. The result that is returned from the SAF call is used to determine whether the user is allowed access to the system. If the SAF call succeeds, the user is logged on to the Content Manager OnDemand server and the required function is performed. If the SAF call fails, the user is prompted for a user ID and password.

Tip: To enable PassTickets in a security manager such as RACF, you must do the following steps:

- Activate the PKTDATA class.
- Define a secured sign-on application key for each application.
- Issue the **SETROPTS RACLIST(PTKTDATA)** command.

Configuring the use of ARS.PTGN

About this task

For this exit to be called, an exit routine must be associated with the ARS.PTGN exit point. To associate the exit routine to the ARS.PTGN exit point, do either of the following tasks:

- Modify the EXIT statement of the PROG*xx* parmlib member. For more information about the PROG*xx* parmlib member, see *z/OS MVS Initialization and Tuning Reference*.
- Run the **SETPROG EXIT** operator command. For information about the **SETPROG EXIT** command, see *z/OS MVS System Commands*.

This exit relies on the use of PassTickets which must be enabled in the external security product. For RACF, this involves:

1. Activating the PTKTDATA class.
2. Defining a secured signon application key for each application in a profile in the PTKTDATA class.
3. Issuing the **SETROPTS RACLIST(PTKTDATA)** command.

Consult the *z/OS SecureWay Security Server RACF System Programmer's Guide* in the section titled "Using the Secured Signon Function" for more details. For other products, consult the appropriate documentation on enabling the PTKTDATA class.

The sample ARSPTGN routine uses an application named ARSSOCKD. This application can be changed to suit installation requirements. To enable the sample user exit, the following command can be issued: `SETPROG EXIT,ADD,EXITNAME=ARS.PTGN,MOD=ARSPTGN,DSNAME=X.SARSLOAD`, where the **DSNAME** parameter specifies the actual data set name of the SARSLOAD data set.

ARS.PTGN exit routine

This section contains information for the ARS.PTGN installation exit.

Exit routine environment

The exit routine is entered in the following environment:

Authorization
Supervisor state and key 0

Dispatchable unit mode
Task

Cross Memory Mode
PASN=HASN=SASN

AMODE
31-bit

ASC Mode
Primary

Interrupt Status
Enabled for IO and External interrupts

Locks
None

Programming considerations

The ARS.PTGN exit must be reentrant. Do not assume that the exit is going to be run under the address space or TCB of the user for whom the PassTicket is required.

Registers at entry

The following list describes the contents of the registers at entry:

GPR 0
Does not contain any information for use by ARS.PTGN exit

GPR 1
Contains the address of the address of a ARSXPTGN

GPR 2 – 12
Does not contain any information for use by ARS.PTGN exit

GPR 13
Address of an 18 word save area

GPR 14
Return Address

GPR 15
Entry point address

On the call to the MVS dynamic exit facility, the PTGNACTN field is initially set to PTGNANO, PTGNEYE is set to "ARSXPTGN", PTGNVERS is set to 1, PTGNUSER contains the length and user ID the PassTicket is to be generated for, and PTGNGRP contains the length and group the PassTicket is to be generated for. The ARSXPTGN macro and the ARSPTGN sample exit is provided in the SARSINST data set.

Return specifications

If the exit wants Content Manager OnDemand to attempt to logon using a PassTicket, it must set the PTGNACTN field to PTGNAOK, and it must set PTGNAPPL to a valid eight-byte application name, and set PTGNTKT to a one-byte length followed by an eight byte PassTicket. These values must be acceptable for the **APPL** and **PASSWRD** parameters of a RACROUTE REQUEST=VERIFY call.

If the exit does not want Content Manager OnDemand to do PassTicket Processing, set PTGNACTN to PTGNANO.

Chapter 52. CICS® client unified login

To allow CICS users to logon without a password, define a profile in the FACILITY class to protect the ARS.CICSUNIF resource and grant users READ access. Only users that are allowed to view documents by using the CICS client should be allowed access to this resource. Specifically, users with Content Manager OnDemand administrative capabilities should be prohibited from using this resource. That is because the authentication used for the CICS client is not as strong as PassTickets. More specifically, the UTOKEN of the CICS client is sent to the server. This UTOKEN is then converted to external format by using a RACROUT REQUEST=TOKENMAP and examined. If the UTOKEN appears correct and is permitted read access to the ARS.CICSUNIF resource in the FACILITY class, the login is allowed.

Chapter 53. Report specifications archive definition exit

The Content Manager OnDemand report specifications archive definition exit allows an installation to modify some of the parameters used by Content Manager OnDemand when document data is being captured (loaded) by the ARSLOAD program, as illustrated in the following figure.

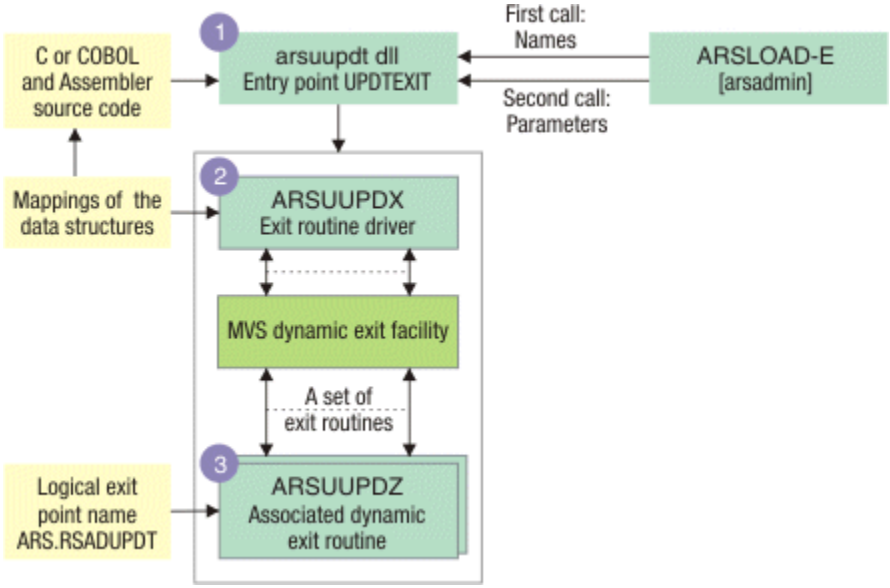

Figure 17. Overview of the report specifications archive definition exit

The following parameters can be modified by using two calls to the ARSUUPDT DLL file:

1. The Names call is used to modify the following items:
   - The application group name.
   - The application name.
   - The name of the object server to be used for data storage.
   - The name of the storage node to be used for data storage.
2. The Parameters call is used to modify the following items:
   - The indexer parameters set.
   - The input file control character type, logical record length and record format.

Interface exit components

The report specifications archive definition exit interface exit consists of the following components, all of which are provided in source form:

Assembler interface modules

- ARSUUPDA
- ARSUUPDX
- ARSUUPDZ

C source modules

- ARSUUPDH

- ARSUUPDT

COBOL source modules

- ARSUUPDB
- ARSUUPDC
- ARSUUPDJ

ARSUUPDT is a DLL module written in the C programming language. It is provided in both source and executable forms, with the source being provided mainly to help in understanding how the ARS.RSADUPDT exit is driven. If you choose to modify ARSUUPDT, it must be compiled with the IBM C/C++ compiler. For information about compiling and binding a C DLL, see http://www.ibm.com/systems/z/os/zos/bkserv/. The ARSUUPDT DLL invokes module ARSUUPDX.

Module ARSUUPDX interfaces with the MVS Dynamic Exit Facility to route control to a set of exit routines. Module ARSUUPDZ is an example of such an exit routine. Modules ARSUUPDX and ARSUUPDZ are written in S/390 assembly language and require the use of the IBM High Level Assembler for proper assembly.

ARSUUPDA (assembler) and ARSUUPDH (C) provide mappings of the data structures that are presented to exit routines associated with the exit point defined by ARSUUPDX (for example, the data structure presented as an input parameter to ARSUUPDZ).

The ARSUUPDT DLL can alternatively be built using the COBOL source modules provided. ARSUUPDB contains a COBOL copybook equivalent to the C ARSUUPDH header file. ARSUUPDC contains a COBOL equivalent to ARSUUPDT. ARSUUPDJ contains sample JCL to compile and bind the ARSUUPDC sample.

High level language interfaces

The report specifications archive definition exit is implemented by a single DLL - arsuupdt. The samples shipped (ARSUUPDT and ARSUUPDC) initialize the ARSUUPDA structure and call ARSUUPDX ARS.RSADUPDT exit driver.

Arsuupdt DLL

When ARSLOAD loads a document, it makes two calls of the DLL.

1. The Names call allows the exit to modify the following specifications:

   application group name
   application name
   object server name
   storage node name
   DbField_date format
   DbField_name

2. The Parameters call allows the exit to modify the following parameters:

   - indexer parameters
   - viewer parameters:

     logical record length
     record format
     control character type

The arsuupdt DLL must be stored in the `/usr/lpp/ars/V10R5M0/bin/exits` directory and must have the APF extended attribute turned on. The arsuupdt DLL must have an exported entry point of UPDTEXIT. A sample to compile and link the ARSUUPDC is provided in ARSUUPDJ in the SARSINST data set.

When compiling the ARSUUPDT C sample the following compiler options must be specified:

```
DEFINE(OS390)
ROSTRING
```

DLL
LONGNAME
RENT
XPLINK

The **-E** option must be specified to ARSLOAD before the exit will be called.

The exit can run in a threaded environment. The exit must be thread-safe.

Format

The following shows the C language arsuupdt.

```
#pragma export(UPDTEXIT)
#include "arscsxit.h"

int UPDTEXIT( ArsCSXitUpdtExit  updt )
```

The following shows the COBOL language arsuupdt.

```
identification division.
 program-id.   "UPDTEXIT" recursive.
linkage section.

copy arscsxic.

procedure division using
            by reference ArsCSXitPrepExit.
```

General description

On entry, review the values stored in the `updt->Function` or `ArsCSXitUpdtExit-Function` fields to determine the type of call to make:

Names call

ARCCSXIT_PROCESS_NAMES (C)
arccsxit-processnames (COBOL)

Parameters call

ARCCSXIT_PROCESS_PARMS (C)
arccsxit-processparms (COBOL)

Input parameters for Names call

The following list shows the input parameters valid for the Names call. Do not attempt to use parameters that are not specified as being valid for a given call.

updt->pFileName
ArsCSXitUpdtExit-pFileName
Address of the null delimited file name. If the JES information pointer is null and the file name begins with the characters DD:, the remainder of the string contains the DD name of the MVS data set name used as input.

updt->ApplGrpName
ArsCSXitUpdtExit-ApplGrpName
Null delimited application group name. This is the application group name that ARSLOAD will attempt to use if no action is performed by the Names call.

updt->ApplName
ArsCSXitUpdtExit-ApplName
Null delimited application name. This is the application name that ARSLOAD will attempt to use if no action is performed by the Names call.

updt->ObjServer
ArsCSXitUpdtExit-ObjServer
Null delimited application name. This is the object server that ARSLOAD will attempt to use if no action is performed by the Names call. If no object server is explicitly specified, ARSLOAD will use the object server specified by the storage node that is designated for loading in the storage set assigned to the application group.

updt->StorageNode
ArsCSXitUpdtExit-Node
Null delimited storage node name. This is the storage node that ARSLOAD will attempt to use if no action is performed by the Names call. If no node is explicitly specified, ARSLOAD will use the primary storage node specified by the storage node that is designated for loading in the storage set assigned to the application group.

updt->ArsCSXitUpdtExit-pJES
ArsCSXitUpdtExit-pJES
Pointer to the JES information: ArsCSXitUpdtExit_JES (C) or ArsCSXitUpdtExit-JES (COBOL). If this pointer is not null, the object being loaded is being read from the JES SPOOL. The JES information contains a DD name that is currently allocated to the SPOOL file and a pointer to the JES SSS2 SSOB extension. If null, the file being processed is either an MVS data set or an HFS file.

Input parameters for Parameters call

The following list shows the input parameters valid for the Parameters call. Do not attempt to use parameters that are not specified as being valid for a given call.

updt->pFileName
ArsCSXitUpdtExit-pFileName
Address of the null delimited file name. If the JES information pointer is null and the file name begins with the characters DD:, the remainder of the string contains the DD name of the MVS data set name used as input.

updt->ApplGrpName
ArsCSXitUpdtExit-ApplGrpName
Null delimited application group name. This is the application group name the document will be stored under.

updt->ApplName
ArsCSXitUpdtExit-ApplName
Null delimited application name. This is the application name the document will be stored under.

updt->ObjServer
ArsCSXitUpdtExit-ObjServer
Null delimited application name. This is the object server that will be stored in. If the object server is X'00', ARSLOAD will use the object server specified by the storage node that is designated for loading in the storage set assigned to the application group.

updt->StorageNode
ArsCSXitUpdtExit-Node
Null delimited storage node name. This is the storage node that ARSLOAD will be stored in. If the node is X'00', ARSLOAD will use the primary storage node that is specified by the storage node that is designated for loading in the storage set that is assigned to the application group.

updt->ArsCSXitUpdtExit-pJES
ArsCSXitUpdtExit-pJES
Pointer to the JES information: **ArsCSXitUpdtExit_JES** (C) or **ArsCSXitUpdtExit-JES** (COBOL). If this pointer is not null, the object being loaded is being read from the JES SPOOL. The JES information contains a DDNAME that is currently allocated to the SPOOL file and a pointer to the JES SSS2 SSOB extension. If null, the file being processed is either an MVS data set or an HFS file.

updt->IndexerParms
ArsCSXitUpdtExit-Indexer
The indexer parameter that will be passed to the indexer. The indexer parameters are a series of records separated by newline characters (X'15'). If altered by the exit, this parameter is passed to the

indexer. If `updt->update_appl` or `ArsCSXitUpdtExit-update-appl` are non-zero, the indexer parameter is stored back into the indexer parameters that are associated with the application.

updt->CCType
ArcCSXitUpdtExit-cc-type
Carriage control type. If altered by the exit, this value will be stored in the view information that is associated with the application. Changing this value after reports have been loaded may cause previously loaded documents to display incorrectly.

updt->LRECL
ArsCSXitUpdtExit-lrecl
For fixed record format the length of each line. This has no meaning for variable format. If altered by the exit, this value will be stored in the view information that is associated with the application. Changing this value after reports have been loaded may cause previously loaded documents to display incorrectly.

updt->RECFM
ArsCSXitUpdtExit-recfm
The record format the document is stored in Content Manager OnDemand. If altered by the exit, this value will be stored in the view information that is associated with the application. Changing this value after reports have been loaded may cause previously loaded documents to display incorrectly.

updt->update_appl
ArsCSXitUpdtExit-update-appl
Indicates that the application is to be updated. If zero, any changes made by the exit will not be reflected in the application definitions. If non-zero, the application will be updated with the new values.

updt->Delim
ArcCSXitUpdtExit-delim
For documents stored as stream, this is the NULL terminated string that contains the string that is used to determine record boundaries. For example, records that use EBCDIC newline characters as record delimiters would specify X'1500'.

Valid values for record format

Valid values for record format are described in Table 10 on page 211.

Table 10. Record formats

Format	C	COBOL
Fixed	ARCCSXIT_DOC_FORMAT_FIXED	ArcCSXitUpdtExit-fixed
Variable	ARCCSXIT_DOC_FORMAT_VARIABLE	ArcCSXitUpdtExit-variable
Stream	ARCCSXIT_DOC_FORMAT_STREAM	ArcCSXitUpdtExit-stream

Valid values for carriage control

Valid values for carriage control are described in Table 11 on page 211.

Table 11. Carriage control

Format	C	COBOL
ANSI	ARCCSXIT_CC_ANSI	ArcCSXitUpdtExit-cc-ansi
Machine	ARCCSXIT_CC_MACHINE	ArcCSXitUpdtExit-cc-machine
None	ARCCSXIT_CC_NONE	ArcCSXitUpdtExit-cc-none

Note:

1. Carriage control, lrecl, recfm, and record delimiter are only valid for documents stored as Line data.

2. Documents stored as variable in Content Manager OnDemand are stored with a two-byte length prefix followed by the data for the record. The length does not include the two-byte prefix.

Returned values

In addition to updating the ArsCSXitPrepExit as appropriate, the DLL should set a return code zero to indicate success, and a non-zero to indicate failure.

Assembler language interfaces

Use of the MVS™ Dynamic Exit Facility

Module ARSUUPDX interfaces with the MVS Dynamic Exit Facility to:

- Define the logical exit point name: ARS.RSADUPDT
- Route control to a set of associated exit routines and process the results of their execution.

Module ARSUUPDZ is implemented as an associated dynamic exit routine.

An exit routine is eligible for execution once it has become associated with the logical exit point. The MVS Dynamic Exit Facility provides several methods for performing this association. For example, the following operator command can be used to associate ARSUUPDZ with the logical exit point name ARS.RSADUPDT. (The example assumes that ARSUUPDZ can be found in the LPA or a LNKLST data set.)

```
SETPROG EXIT,ADD,EXITNAME=ARS.RSADUPDT,MODNAME=ARSUUPDZ
```

See the following IBM publications for more information about the MVS Dynamic Exit Facility:

- *z/OS MVS Programming: Authorized Assembler Services Guide*
- *MVS Programming: Authorized Assembler Services Reference*
- *z/OS MVS Initialization and Tuning Reference*
- *z/OS MVS System Commands*

Enabling the report specifications archive definition exit

About this task

The report specifications archive definition exit is enabled only when the **-E** parameter is passed to the ARSLOAD program.

Note: The **-E** parameter must be specified in upper case.

Overview of the operation of ARSUUPDX

ARSUUPDX iteratively invokes the MVS Dynamic Exit Facility to route control to each of the exit routines associated with the ARS.RSADUPDT logical exit point. When multiple exit routines are associated with the exit point, it is unpredictable in which order these routines are invoked.

If an associated exit routine ABENDs or returns an undefined Return Code, that exit routine is regarded as invalid. It is essentially treated as though it had not been associated with the exit point.

For a given event, each successive exit routine sees the effects of the execution of its predecessors. If an associated exit routine fails, an attempt is made to mitigate the effects of that failure by restoring the state of all exit routine interface parameters to the values which existed prior to the invocation of the failed exit routine.

Capabilities and general characteristics of an associated exit routine

An associated exit routine is invoked twice for each document being processed by the ARSLOAD program. The first invocation is referred to as Names processing and the second is referred to as Parameters processing. The characteristics of each of these invocations is further discussed below.

Upon return, an associated exit routine must indicate whether it has altered any of the modifiable parameters via the setting of the ARSUUPDA_UPDATE field. A value of zero in this field indicates that no permanent parameter modifications have been performed. A non-zero value indicates that at least one parameter has been modified, and that the changes are to be stored for the application.

When the returned value in ARSUUPDA_UPDATE is zero, any parameter modifications except for ARSUUPDA_IXPRMP that might have been performed by the exit routine are discarded. Changes to ARSUUPDA_IXPRMP are passed to the indexer, but do not save the altered value in the application.

In addition to setting the ARSUUPDA_UPDATE field, an associated exit routine must provide a return code in General Register 15. If this return code is non-zero:

- Processing of the current input stream is not performed and any parameter modifications that may have been performed by the exit routine are discarded.
- Additional exit routines associated with the exit point, if any, are not invoked for the current event.

In general, all character string items that are allowed to be modified by an associated exit routine are accessed by using descriptors that are contained in the ARSUUPDA structure. Each descriptor contains the following information:

- The pointer to the string area.
- The true length of the text string.

  Note: This value may be zero if the corresponding item is undefined to the Content Manager OnDemandsystem.
- The length of the string area. This value reflects the maximum length string which may be placed in the area.

When an exit routine wants to change a string item, it must place the new string data in the area pointed to by the descriptor that is contained within the ARSUUPDA structure and must set the corresponding string length field accordingly. The exit routine must not alter the pointer itself and must not alter the area length field. If, for any modifiable item, either of these fields is altered, the behavior of the exit routine is regarded as invalid and all results of the associated exit routine are discarded.

When the ARSLOAD program is attempting to capture a JES SPOOL file:

- A pointer to a Read Only copy of the SAPI SSOB Extension is provided in field ARSUUPDA_SSS2P. The SAPI SSOB Extension is mapped by macro IAZSSS2. For more information regarding the content of this structure, see *z/OS MVS Using the Subsystem Interface*.
- Field ARSUUPDA_INPDDN contains the DDNAME associated with the JES SPOOL file being processed. The exit routine may OPEN for INPUT processing a QSAM DCB specifying this DDNAME and can read the data contained in the SPOOL file.

When the ARSLOAD program is processing other input sources (for example, when reading a conventional data set while executing as a step in a batch job), no interactions with JES are taking place and, therefore, no SAPI SSOB Extension exists. In this situation:

- The ARSUUPDA_SSS2P field is zero.
- The ARSUUPDA_INPDDN field is blanks.
- The name of the actual file being processed is available through the descriptor, composed of fields ARSUUPDA_FNAMEP and ARSUUPDA_FNAMEL. The file name string can be up to 1024 bytes in length and might be a conventional MVS data set name, an HFS file name (with pathing information, for example: `usr/tmp/input.txt`), or a special Language Environment file name (for example: DDN:SYS00001).

Note: When the ARSLOAD program is invoked by ARSYSPIN, it appears to the ARSLOAD program that it is processing a conventional MVS data set, therefore, none of the JES related information is available.

All associated exit routines must possess the REENTRANT attribute and must reside in an APF Authorized library.

The execution environment at entry to an exit routine is as follows:

Dispatchable Unit Mode
Task

Cross Memory Mode
PASN=HASN=SASN

Addressing Mode
31-bit

ASC Mode
Primary

Interrupt Status
Fully enabled

Locks
No locks are held.

Authorization
Problem State -AND-APF Authorized

Control parameters
All parameters reside in the Primary Space.All addresses are pure 31-bit values.

General Registers

R1:
Points to a fullword which, in turn, points to the interface parameter list structure as mapped by the ARSUUPDA DSECT.

R13:
Points to a standard 18-word general register save area.

R14:
Contains the address to which the exit routine is to branch upon completion, that is, the Return Address.

R15:
Contains the entry point address of the exit routine.

Upon completion, an exit routine must return to the address contained in R14 at entry, with the general registers loaded as follows:

R0 - R14:
The same as at entry to the exit routine.

R15:
Contains a return code.

Field ARSUUPDA_ACTION within the ARSUUPDA structure indicates the event or activity for which the exit routine is being invoked. The following list describes the values contained in ARSUUPDA_ACTION:

ARSUUPDA_ACNAMES
Names processing.

ARSUUPDA_ACPARMS
Report definition parameters processing.

The symbols are defined within the ARSUUPDA DSECT as well as within the ARSUUPDH C structure mapping.

The valid return code value from an exit routine is ARSUUPDA_RCNORM, which means normal completion. The symbol is defined within the ARSUUPDA DSECT as well as within the ARSUUPDH C structure mapping.

Names processing

The Names processing invocation allows an associated exit routine to modify the following specifications:

- The application group name
- The application name
- The object server name
- The storage node name
- The DbField_date format
- The DbField_name

Any of these name strings may be null (that is, the corresponding field length value within the ARSUUPDA structure may be zero) if the corresponding definitions do not exist within the Content Manager OnDemand system.

The modification of any other interface item (for example, the indexer parameters string) is discarded.

Upon the successful completion of execution of all associated exit routines, Content Manager OnDemand validates the effective names values. If any validation is unsuccessful, the Parameters invocation for the current event is not performed.

Parameters processing

The Parameters processing invocation for a given event occurs after all associated exit routines have been invoked for Names processing for the same event.

The Parameters processing invocation allows an associated exit routine to modify the following specifications:

- The Indexer Parameters set string.
- The following expected characteristics of the input file being captured:
  - Logical record length
  - Record format
  - Control characters type

The Indexer Parameters set string might be null (that is, the corresponding field length value within the ARSUUPDA structure may be zero) and the values associated with the expected characteristics of the input file may be indeterminate if the corresponding definitions do not exist within the Content Manager OnDemand system.

The modification of any other interface item (for example, the application name) is discarded.

When the Indexer Parameters set string is not null, each element of the set (for example, each ACIF control statement) is terminated by an EBCDIC newline character (X'15').

Upon the successful completion of execution of all associated exit routines, Content Manager OnDemand attempts to continue processing of the input file using the effective parameters.

If the report specification archive definition exit alters the indexer parameters and specifies a zero value for ARSUUPDA_UPDATE (assembler), `updt->update_appl` (C), or `ArsCSXitUpdtExit-update-appl` (COBOL), the indexer parameters altered by the exit are passed to the indexer, however, they are not updated in the application.

Other considerations

Upon the successful completion of the execution of all associated exit routines for a given event, the effective value of each parameter item is stored in the Content Manager OnDemand system tables. Changes to these parameter items can negatively effect subsequent processing of data previously captured by Content Manager OnDemand. Consequently, extreme caution must be observed when modifying any of these items.

Chapter 54. Client preview exit

The Content Manager OnDemand client preview exit allows an installation to process document data before the document is presented to the client. The following figure shows an overview of the client preview exit.

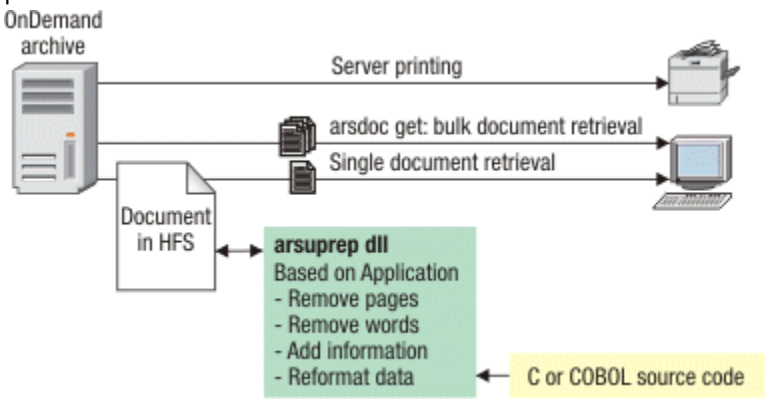

Figure 18. Overview of the client preview exit

The client preview exit can be used to add, remove, or reformat data before the document is presented to the client. For example:

- Remove pages from the document, such as banner pages, title pages, all pages but the summary page, and so on.
- Remove specific words, columns of data, or other information from the document. That is, omit ("white out") sensitive information such as salaries, social security numbers, and birth dates.
- Add information to the document, for example, a summary page, data analysis information, and Confidential or Copy statements.
- Reformat data contained in the document, for example, reorder the columns of data.

Note:

1. The client preview exit is not called for all document retrievals. In particular, it is not called for functions that use the so-called Bulk Retrieval method of retrieving documents or for server printing. For example, running the ARSDOC GET function without specifying the **-n** parameter performs a bulk retrieval, and documents retrieved will not be presented to the client preview exit.
2. The client preview exit is not called for server printing.
3. If a request is made to retrieve a large object document, care should be taken to make certain that the client preview exit does not remove any pages from the document. The large object segment size and page navigation information are based on the number of pages that existed when the document was loaded on the server. Unexpected results may occur if this information is changed.
4. The client preview exit is enabled at the application level. The exit is invoked only for the specified applications.

Interface exit components

In addition to the ARSCSXIT (C header) and ARSCSXIC (COBOL copybook) members of SARSINST, the following are provided in source form in SARSINST:

ARSUPREP
ARSUPREC
ARSUPREJ

ARSUPREP contains a sample exit written in C. ARSUPREC contains a sample exit in COBOL. ARSUPREJ contains sample JCL to compile and bind the COBOL sample.

The client preview exit is implemented by a DLL called arsuprep. There is only one arsuprep DLL per Content Manager OnDemand instance. This DLL must be stored in the `/usr/lpp/ars/V10R5M0/bin/exits` directory. The DLL must have an exported entry point of PREPEXIT and be APF authorized.

When compiling the ARSUPREP C sample the following compiler options must be specified:

DEFINE(OS390)
ROSTRING
DLL
LONGNAME
RENT

The client preview exit is enabled for specific applications. For example, your archive might contain thousands of applications but you only want to enable the client preview exit on a few of these applications. To enable the client preview exit for a specific application, ensure that the **Use Preview Exit** option is selected on the **Miscellaneous Options** page of the application.

This exit runs in a threaded environment. The exit must be thread-safe.

ARSUPREP contains logic to read and write the HFS files that are passed from the Content Manager OnDemand object server to the client. When the Content Manager OnDemand server displays a segment, the server retrieves the segment and places it in an HFS file. When a preview exit is specified for an application, the HFS file is passed to the preview exit, arsuprep. After the preview exit completes its processing, a modified HFS file is created and control is passed back to the Content Manager OnDemand server. In a multiple object server environment, the arsuprep DLL is installed on each object server.

Format

The following shows the C language arsuprep.

```
#pragma export(PREPEXIT)
#include "arscsxit.h"
int PREPEXIT( ArsCSXitPrepExit prep )
```

The following shows the COBOL language arsuprep

```
identification division.
 program-id.   "PREPEXIT" recursive.

linkage section.

copy arscsxic.

procedure division using
                by reference ArsCSXitPrepExit.
```

General Description

The following list shows the input parameters for the exit. Both C (ARSCSXIT) and COBOL (ARSCSXIC) field definitions are provided, with the C definitions appearing first. Do not attempt to access storage beyond the terminating X'00' for string values. The exit should not modify any input parameters except the output file name.

prep->pUserid
ArsCSXitPrepExit-pUserid
Contains the address of the null delimited userid requesting the document.

prep->pInFileName
ArsCSXitPrepExit-pInFileName
Contains the address of the null delimited absolute file name of the HFS file containing the document as retrieved by Content Manager OnDemand.

prep->OutFileName
ArsCSXitPrepExit-OutFileName
To alter the document presented to the user, set this field to the null delimited absolute file name of the HFS file containing the document.

prep->pUserParms
ArsCSXitPrepExit-pUserParms
A pointer to the null delimited user parameters. These parameters are specified on the **Miscellaneous Options** page of the application by using the Content Manager OnDemand administrative client.

prep->pApplGrp
ArsCSXitPrepExit-pApplGrp
A pointer to the ArcCSXitApplGroup for the application group that is associated with the document that the user is requesting.

prep->pAppl
ArsCSXitPrepExit-pAppl
A pointer to the ArcCSXitAppl for the application that is associated with the document that the user is requesting.

prep->pDoc
ArsCSXitPrepExit-pDoc
A pointer to the ArcCSXitDoc for the document that the user is requesting.

The following list shows the fields in the ArcCSXitApplGroup:

name
ArcCSXitApplGroup-namep
Contains the address of the null delimited application group name.

agid
ArcCSXitApplGroup-agid
Contains the full word application group identifier.

agid_name
ArcCSXitApplGroup-agid-namep
Contains the address of the null delimited internal group name.

The following list shows the fields in the ArcCSXitAppl.

name
ArcCSXitAppl-name
Contains the address of the null delimited application name.

aid
ArcCSXitAppl-aid
Contains the full word application identifier.

doc-type
ArcCSXitAppl-type
Contains a one byte document type indicator.

doc-fmt
ArcCSXitAppl-doc-fmt
For line data documents, contains a one byte document format indicator.

u.fixed
ArcCSXitAppl-fixed
For fixed format line data documents, contains a fullword that is the line length for each record.

u.stream
ArcCSXitAppl-stream
For stream format line data documents, contains a null-delimited string that is the end of record indicator. Note that this indicator may be more than one byte.

trc_present
ArcCSXitAppl-trc-present
A one-byte document indicator that a table reference character is present (1) or not (0). Only valid for line data.

line_count
ArcCSXitAppl-line-count
A full word count of the lines per page. Only valid for line data.

code_page
ArcCSXitAppl-code-page
A full word containing the code page that is associated with the data. Only valid for line data.

cc_type
ArcCSXitAppl-cc-type
A one byte indicator of the carriage control. Only

prmode
ArcCSXitAppl-prmode
For an SCS or line data document, if the document contains shift-out and shift-in codes, this field determines the processing mode for the document.

The following list shows the valid document types:

ARCCSXIT_DOC_TYPE_AFP
ArcCSXitAppl-type-afp
AFP

ARCCSXIT_DOC_TYPE_BMP
ArcCSXitAppl-type-bmp
Bitmap

ARCCSXIT_DOC_TYPE_EMAIL
ArcCSXitAppl-type-email
E-mail

ARCCSXIT_DOC_TYPE_GIF
ArcCSXitAppl-type-gif
GIF

ARCCSXIT_DOC_TYPE_JFIF
ArcCSXitAppl-type-jfif
JFIF

ARCCSXIT_DOC_TYPE_LINE
ArcCSXitAppl-type-line
Line data

ARCCSXIT_DOC_TYPE_META
ArcCSXitAppl-type-meta
Metacode

ARCCSXIT_DOC_TYPE_PCX
ArcCSXitAppl-type-pcx
PCX

ARCCSXIT_DOC_TYPE_PDF
ArcCSXitAppl-type-pdf
PDF

ARCCSXIT_DOC_TYPE_PNG
ArcCSXitAppl-type-png
PNG

ARCCSXIT_DOC_TYPE_SCS
ArcCSXitAppl-type-scs
SCS

ARCCSXIT_DOC_TYPE_SCS_ EXT
ArcCSXitAppl-type-scs-ex
SCS extended

ARCCSXIT_DOC_TYPE_TIFF
ArcCSXitAppl-type-tiff
TIFF

ARCCSXIT_DOC_TYPE_ USRDEF
ArcCSXitAppl-type-usrdef
User defined

The following list shows the valid document formats for line data:

ARCCSXIT_DOC_FORMAT_ FIXED
ArcCSXitAppl-doc-fmt-fixed
The document is stored as fixed length records.

ARCCSXIT_DOC_FORMAT_ VARIABLE
ArcCSXitAppl-doc-fmt-variable
The document is stored as a series of records consisting of a two-byte length followed by the actual record. The length does not include the two-byte length.

ARCCSXIT_DOC_FORMAT_ STREAM
ArcCSXitAppl-doc-fmt-stream
The document is stored as a series of records delimited by a stream delimiter character sequence.

The following list shows the valid carriage control:

ARCCSXIT_CC_ANSI
ArcCSXitAppl-cc-ansi
ANSI Carriage control

ARCCSXIT_CC_MACHINE
ArcCSXitAppl-cc-machine
Machine Carriage control

ARCCSXIT_CC_NONE
ArcCSXitAppl-cc-none
No carriage control

The following list shows the valid processing modes:

ARCCSXIT_PRMODE_NONE
ArcCSXitAppl-pr-none
The document does not contain shift-out and shift-in characters.

ARCCSXIT_PRMODE_SOSI1
ArcCSXitAppl-pr-sosi1
Shift-out and shift-in characters are replaced with a blank.

ARCCSXIT_PRMODE_SOSI2
ArcCSXitAppl-pr-sosi2
Shift-out and shift-in characters are removed.

ARCCSXIT_PRMODE_SOSI3
ArcCSXitAppl-pr-sosi3
Shift-out and shift-in characters are replaced with two blanks.

The following list shows the fields in ArcCSXitDoc:

doc_flds.flds_num
ArcCSXitDoc-flds-num
A full word containing the number of index fields that are associated with the document.

doc_flds.flds
ArcCSXitDoc-flds
Contains a pointer to the ArcCSXitField table for the document. This table consists of one entry for each index field, containing the database column name and value for index.

doc_hndl.name
ArcCSXitDoc-name
Null delimited internal document name for this document (for example, 23FAAC)

doc_hndl.doc_off
ArcCSXitDoc-doc-off
Offset of the document within the decompressed object.

doc_hndl.doc_len
ArcCSXitDoc-doc-len
Length of the document within the decompressed object.

doc_hndl.comp_off
ArcCSXitDoc-comp-off
Offset of the compressed object within the object.

doc_hndl.comp_len
ArcCSXitDoc-comp-len
Length of the compressed object within the object.

The following list shows the fields in the ArcCSXitField table:

db-name
ArcCSXitField-db-namep
A pointer to the null delimited database column name.

type
ArcCSXitField-type
A one byte indicator for the type of the data base field type.

qual
ArcCSXitField-qual
Indicates if the field is a date/time value and if so, what type.

u.d
ArcCSXitField-d
Double precision floating point value.

u.b
ArcCSXitField-b
Eight-byte integer value.

u.i
ArcCSXitField-i
Four-byte integer value.

u.n
ArcCSXitField-n
Two-byte integer value.

u.str
ArcCSXitField-strp
Address of the null delimited string.

The following list shows the database field types:

ARCCSXIT_FIELD_TYPE_ BIGINT
ArcCSXitField-type-bigint
Eight byte integer.

ARCCSXIT_FIELD_TYPE_ DECIMAL
ArcCSXitField-type-decimal
Double precision floating point.

ARCCSXIT_FIELD_TYPE_ INTEGER
ArcCSXitField-type-integer
Four byte signed integer.

ARCCSXIT_FIELD_TYPE_ SMALLINT
ArcCSXitField-type-smallint
Two byte signed integer.

ARCCSXIT_FIELD_TYPE_ STRING
ArcCSXitField-type-string
Null delimited string.

The following shows the database field qualifiers:

ARCCSXIT_FIELD_TYPE_ QUAL_BASE
ArcCSXitField-qual-base
Not a date/time value.

ARCCSXIT_FIELD_TYPE_ QUAL_DATETIME
ArcCSXitField-qual-datetime
Represents a date/time value.

ARCCSXIT_FIELD_TYPE_ QUAL_DATE
ArcCSXitField-qual-date
Represents a date value.

ARCCSXIT_FIELD_TYPE_ QUAL_TIME
ArcCSXitField-qual-time
Represents a time value.

ARCCSXIT_FIELD_TYPE_ QUAL_TZ_DATETIME
ArcCSXitField-qual-tz-datetime
Represents a local date/time value.

Returned values

If the exit wants a different file presented to the user, it should set `prep->OutFileName` (C) or `ArsCSXitPrepExit-OutFileName` (COBOL) to the name of the file. This file must be formatted to agree with the specification in ArcCSXitAppl. For example, the file cannot be in variable format if the ArcCSXitAppl indicates fixed.

Content Manager OnDemandwill delete this file after returning its contents to the user.

The exit should set a return code of 0.

Chapter 55. Table space creation exit

The Content Manager OnDemand table space creation exit is called when Content Manager OnDemand creates a table space, table, or index table that will be used to store application index data.

A Content Manager OnDemand system contains two sets of tables:

- System tables that contain system definitions
- Application group data tables that contain the key values and references to the stored data

The Content Manager OnDemand table space creation exit is not called when Content Manager OnDemand creates the system tables.

The following figure shows an overview of the table space creation exit.

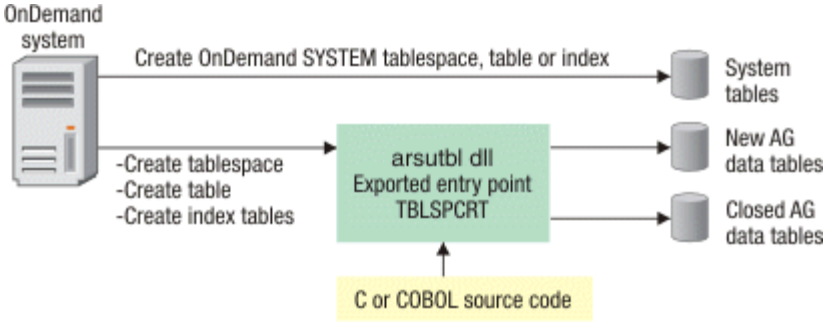

Figure 19. Overview of the table space creation exit

The table space creation exit can alter the SQL that is used to create the table or index. The exit can also cause early closure of an application group data table for earlier than anticipated migrations. However, the exit cannot change table names or table space names for application group data tables.

Interface exit components

The table space creation exit consists of the following components, all of which are provided in source form in SARSINST:

ARSUTBL
Contains a sample table space creation exit in C.

ARSUTBLC
Contains a sample exit in COBOL.

ARSUTBLJ
Contains sample JCL to compile and bind the COBOL sample.

The table space creation exit is implemented by a DLL which in this document is called arsutbl. However the DLL can have any name and can reside in any HFS directory. The sample JCL in ARSUTBLJ places the DLL in `/usr/lpp/ars/V10R5M0/bin/exits/arsutbl`. The DLL must have an exported entry point of TBLSPCRT.

When compiling the ARSUTBL C sample the following compiler options must be specified:

```
DEFINE(OS390)
ROSTRING
DLL
LONGNAME
RENT
```

The following statement must exist in the ARS.CFG file that is associated with the instance so that the arsutbl DLL can be invoked: ARS_DB_TABLESPACE_USEREXIT=*absolute_path_to_exit*. For the sample arsutbl, you would specify the following statement in the ARS.CFG file: ARS_DB_TABLESPACE_USEREXIT=/usr/lpp/ars/V10R5M0/bin/exits/arsutbl

The exit runs in a threaded environment. The exit must be thread-safe. The exit must run APF-authorized. The link edit JCL that is provided in ARSUTBLJ specifies SETOPT option **EXTATTR(APF)**. This option must be specified in the link edit for the table space creation exit, regardless of the programming language or exit name.

Format

The following shows the C language arsutbl.

```
#pragma export(TBLSPCRT)
#include "arscsxit.h"
int TBLSPCRT( ArcCSXitApplGroup appl_grp,
              char tblsp_name,
              char table_name,
              char idx_name,
              char sql,
              int  action,
              int  created
            )
```

The following shows the COBOL language arsutbl.

```
identification division.
program-id. "TBLSPCRT" recursive.

linkage section.

copy arscsxic.

procedure division using
                by reference ArcCSXitApplGroup,
                by reference ArsCSXitTblspCrt-tblsp,
                by reference ArsCSXitTblspCrt-table,
                by reference ArsCSXitTblspCrt-idx-name,
                by reference ArsCSXitTblspCrt-sql,
                byvalue      ArsCSXitTblspCrt-action,
                by reference ArsCSXitTblspCrt-created.
```

General description

On entry, review the values stored in `action` or `ArsCSXitTblspCrt-action` to determine the type of action to perform:

Table space creation
1 (C)

ArsCSXitTblspCrt-act-tspace (COBOL)

Table creation
2 (C)

ArsCSXitTblspCrt-act-table (COBOL)

Index creation
3 (C)

ArsCSXitTblspCrt-act-index (COBOL)

Final call
4 (C)

arccsxit-security-user-update (COBOL)

There are no applicable parameters for the Final call invocation.

Input parameters for the Tablespace Create action

The following list shows the input parameters valid for the Tablespace Create action. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters except the SQL string if supplied for the action.

appl_grp
ArcCSXitApplGroup
Contains information related to the application group for which the table space is being created. This includes the application group name, the application group identifier, and the internal application group name.

tblsp_name
ArsCSXitTblspCrt-tblsp
Null delimited table space name being created.

Input parameters for the Table Create action

The following list shows the input parameters valid for the Table Create action. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters except the SQL string if supplied for the action.

appl_grp
ArcCSXitApplGroup
Contains information related to the application group for which the table space is being created. This includes the application group name, the application group identifier, and the internal application group name.

tblsp_name
ArsCSXitTblspCrt-tblsp
Null delimited table space name being created.

table_name
ArsCSXitTblspCrt-table
Null delimited table name being created.

sql
ArsCSXitTblspCrt-sql
Null delimited SQL that will be used to create the table. The installation can alter this, however the resultant string plus the tailing X'00' must not exceed 16384 bytes.

Input parameters for the Index Create action

The following list shows the input parameters valid for the Table Create action. Index Creation is called once for each index that is being created on the table. Do not attempt to use parameters that are not specified as being valid for a given action. Do not attempt to access storage beyond the terminating X'00'. The exit should not modify any input parameters except the SQL string if supplied for the action.

appl_grp
ArcCSXitApplGroup
Contains information related to the application group for which the table space is being created. This includes the application group name, the application group identifier, and the internal application group name.

tblsp_name
ArsCSXitTblspCrt-tblsp
Null delimited table space name being created.

table_name
ArsCSXitTblspCrt-table
Null delimited table name being created.

idx
ArsCSXitTblspCrt-idx-name
Null delimited index name being created.

sql
ArsCSXitTblspCrt-sql
Null delimited SQL that will be used to create the table. The installation can alter this, however the resultant string plus the tailing X'00' must not exceed 16384 bytes.

Returned values

Set `*created` or `ArsCSXitTblspCrt-created` to one of the following values:

0
Instructs Content Manager OnDemand to create the table space, table, or index.

non-zero
The exit creates the table space, table or index.

The exit should set a return code of 0.

Part 11. Structured APIs

Chapter 56. Structured APIs overview

The following sections give the functional and implementation overview of the APIs that you can use within the Content Manager OnDemand environment for z/OS.

API function overview

Structured APIs allow customer applications in CICS, IMS, TSO or batch environments the ability to invoke the server functions described in Table 12 on page 231.

Table 12. Server functions and their descriptions

Function	Description
LOGON	Establishes a connection to the Content Manager OnDemand library server. After a successful logon, the server returns a list of authorized folders that can be accessed by a specific user.
FOLDER OPEN	Identifies the folder name to be processed by subsequent search and retrieve requests.
HIT LIST	Requests that the Content Manager OnDemand server return a list of items matching the user supplied search criteria.
RETRIEVE	Retrieves a document from a Content Manager OnDemand archive. If a RETRIEVE request with the implicit logoff flag set to Y is processed, a LOGOFF call is made and invokes a RELEASE request after the document is retrieved. The RELEASE request then frees all storage areas except the document structure. When the document structure is no longer needed, you need to make a separate RELEASED request to release the document structure.
BULK RTRIEV	Requests multiple documents to be identified with an appropriate WHERE clause, and retrieves all identified documents.
ANNOTATIONS	Requests that the Content Manager OnDemand server return the annotations for the selected document.
LOGOFF	Allows users to log off a Content Manager OnDemand server. LOGOFF also frees all storage areas that are used by the LOGON, FOLDER OPEN, HIT LIST, and RETRIEVE functions.
RELEASE	Frees all storage areas that are used by the LOGON, FOLDER OPEN, HIT LIST, and RETRIEVE functions.
RELEASEA	Frees the Annotations List Structure created by ANNOTATIONS
RELEASEC	Frees the FolderCriteriaStructure created by FOLDER OPEN
RELEASED	Frees the Document Structure created by RETRIEVE (The Document Structure must be freed by using RELEASED)
RELEASEH	Frees the HitListStructure created by HIT LIST
RELEASEL	Frees the FolderListStructure created by LOGON

Software requirements

The following list contains the requirements that your system needs to use the structured APIs mentioned in “API function overview” on page 231:

- CICS Transaction Server Version 3.1 (or later) if the Structured APIs are implemented in CICS environment.
- IMS Transaction Manager Version 10 (or later) if the Structured APIs are implemented in IMS environment.
- IBM Content Manager OnDemand for z/OS Web Enablement Kit Version 10.5 proof of entitlement.

Implementation environment

The structured APIs are designed to operate in a three-tier environment. You can install these three tiers on a single image or multiple z/OS images (see Figure 20 on page 233).

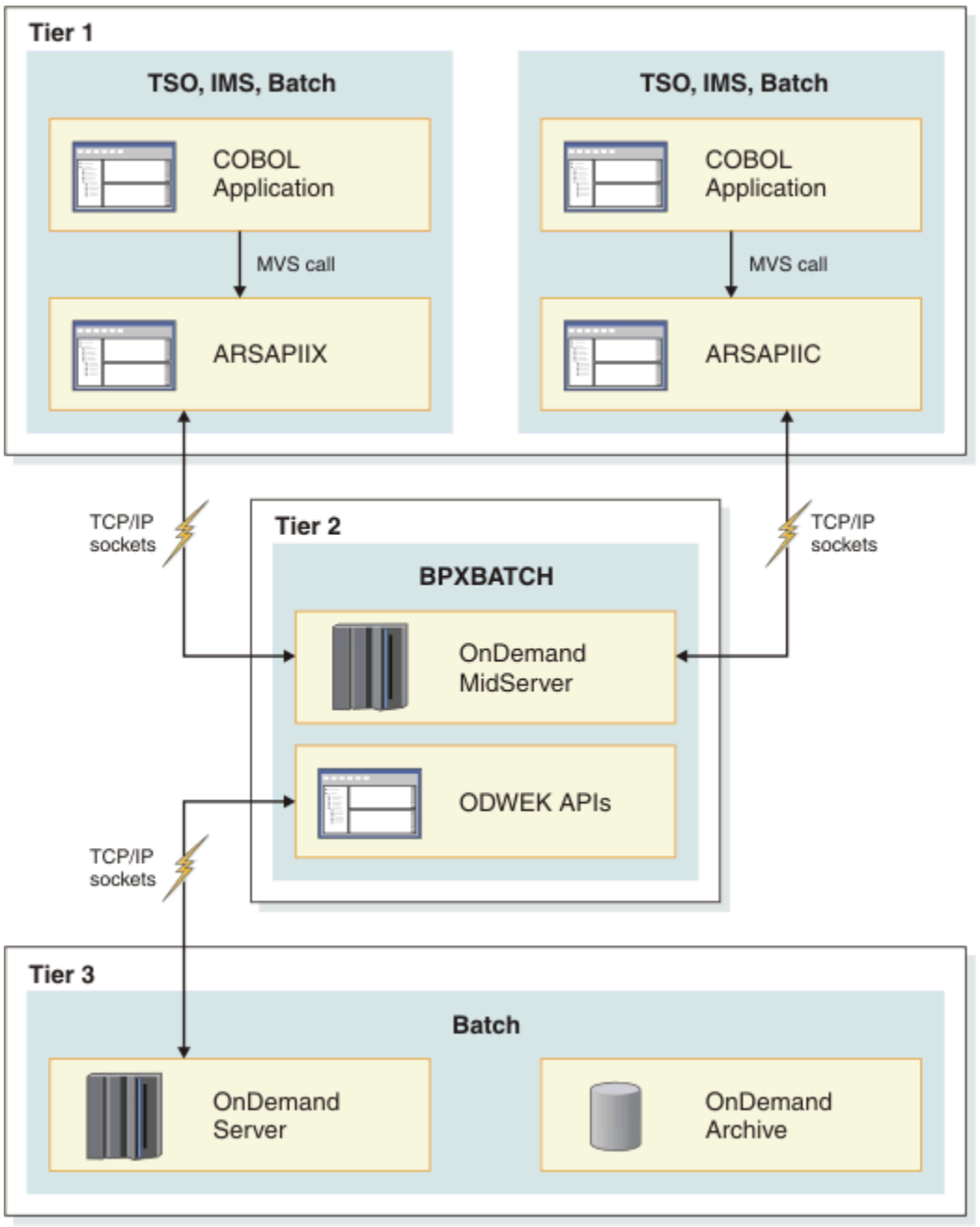

Figure 20. Shows three tiers and the components that are needed for each tier

Tier 1 is where the customer application code runs in conjunction with the IBM-supplied structured API interface module, also known as a stub. Table 13 on page 233 shows which customer application code runs with the stubs supplied by IBM.

Table 13. Customer application program environments and the stubs they use

Environment	Structured API interface module
Batch	ARSAPIIX
TSO	ARSAPIIX
IMS	ARSAPIIX
CICS	ARSAPIIC

Within Tier 1, the customer application program first links to the stub through an MVS dynamic call. Linking to the stub allows the customer application to conduct a series of request/response dialogs with the Content Manager OnDemand server. Each request/response dialog adheres to the following pattern:

1. The customer application code passes a request to the stub.
2. The stub forwards the request to the MidServer in Tier 2 by using TCP/IP sockets as the transmission method.
3. The MidServer presents the data to the standard Content Manager OnDemand Java™ APIs that are also running in Tier 2, and converts the data into Java.
4. The Java APIs forward the request to and receive the response from the Content Manager OnDemand server that runs in Tier 3.
5. The Java APIs return the result and response to the MidServer.
6. The Midserver transfers the data back to the customer application program through the stub.

This architecture provides the following advantages:

- Each of the tiers can run in different LPARs or on different z/OS images, which provides increased flexibility in implementation and load distribution.
- Multiple customer applications can communicate with a single MidServer.
- Multiple Midservers can be set up on different LPARs or on different systems to communicate with the Content Manager OnDemand server.
- The Midserver code can be transferred to a Linux on System z® environment.

Chapter 57. Content Manager OnDemand structured API installation instructions

About this task

The Content Manager OnDemand structured API components are stored in the SMP/E libraries and HFS. All the steps are mandatory unless otherwise indicated in the step description.

After the installation is complete, proceed to Chapter 58, “Using the structured APIs,” on page 249 for detailed information about how to integrate the structured API components with your application program.

Creating the Content Manager OnDemand user libraries

About this task

You must complete the following tasks to create the Content Manager OnDemand user libraries and prepare your system for the structured APIs. The steps will create the five user configuration libraries so that members of the SMP/E libraries can be moved to them for local modifications. Furthermore, by completing the steps, the user load library is created so it can hold local copies of the structured API batch and CICS driver programs and the CICS map.

Creating user configuration libraries

About this task

Important: If you have successfully executed member ARSUALOC in the past, you might need to reallocate dataset USERSAMP to contain the new members distributed with the structured APIs. Review and compare the new recommended size for USERSAMP with the current dataset allocation to see if you need to reallocate USERSAMP.

This task creates the following user configuration libraries:

- USERDB2
- USERJCL
- USERPARM
- USERPROC
- USERSAMP
- USERBRET

Procedure

To create the user configuration libraries, do the following steps using member ARSUALOC from SMP/E SARSINST library:

1. Add a job card.
2. Change occurrences of ?ARS? to an appropriate high-level qualifier.
3. Change occurrence of ?VV? to the VOLSER value to use for placing these libraries.
4. Run the job.

Results

You have now created the six user libraries.

Creating user load library

About this task

This task creates the user configuration library, USERLOAD

Procedure

To create user load libraries, do the following steps using member ARSUALOL from SMP/E SARSINST library:

1. Add a job card.
2. Change occurrences of ?ARS? to an appropriate high-level qualifier.
3. Run the job.

Results

You have now created a user load library and five user configuration libraries.

Populating the user libraries

About this task

You must populate the user configuration libraries before you can customize the members. To populate the user configuration libraries, you must copy the following members:

- From SARSINST to *hlq*.USERJCL
  - ARSAPICJ
  - ARSZDABJ
  - ARSZDCBJ
  - ARSZDCCJ
  - ARSZMP1J
- From SARSINST to *hlq*.USERPARM
  - ARSZUIVP
  - ARSZUIVB
- From SARSINST to *hlq*.USERSAMP
  - ARSZDAPB
  - ARSZDAPC
  - ARSZDCLS
  - ARSZDCWA
  - ARSZDCWS
  - ARSZMP1M
- From SARSINST to *hlq*.USERBRET
  - ARSZDABR

You have now populated the libraries with the appropriate members.

Populating the HFS

About this task

You can run the structured API MidServer in an alternate mount point that is created for Content Manager OnDemand for z/OS Version 10.5. This allows multiple instances of the MidServer to be executed, one for each mount point defined and mounted. The following instructions use "MountPoint" to reference the mount point for the instance of the MidServer installed. The default mount point is `/usr/lpp/ars/V10R5M0`. An alternate mount has no default directory structure.

Make sure that a local mount point is defined for each Content Manager OnDemand mount point to contain user files, including the MidServer configuration files and script. Chapter 11, "Copy server control files," on page 33 describes how such a directory structure and mount point is defined. The installation instructions use the */MountPoint*`/config` directory structure.

You must create an `arswww.ini` file in the */MountPoint*`/config/midserver` directory, and copy two Content Manager OnDemand for z/OS sample files from the */MountPoint*`/samples` directory to the/*MountPoint*`/config/midserver` directory. To create and copy these files to the appropriate directory, complete the following steps:

Procedure

To copy the files to the appropriate directory, do the following steps:

1. Log on as the super user, or, as any user set up with appropriate permissions.
2. Create a directory under */MountPoint*`/config` by issuing the following command while positioned at the */MountPoint*`/config` directory: `mkdir midserver`
3. Copy the MidServer files, `arsMSVR.cfg` and `arsMSVR.sh` by issuing the following command from the UNIX System Services command prompt:`cp /MountPoint/samples/arsMSVR.* /MountPoint/config/midserver`
4. Create directories under */MountPoint*`/config/midserver`.

   Enter the following commands while positioned at the */MountPoint*`/config/midserver` directory:

   ```
   mkdir cache
   mkdir logs
   mkdir temp
   ```
5. Create the midserver `arswww.ini` file by issuing the following command from the USS command prompt: `touch /MountPoint/config/midserver/arswww.ini`

Results

The HFS is now populated with three files, two MidServer configuration files and one ODWEK file.

Modifying the MidServer configuration files

About this task

You must modify the MidServer configuration file, the MidServer shell file, and the ODWEK initialization file to local standards.

Procedure

To modify the MidServer files, do the following:

1. Log on as the super user, or, as any user set up with appropriate permissions.
2. Position to `/MountPoint/config/midserver` by entering the following command: `cd /MountPoint/config/midserver`

3. Change the access mode of the files to allow an update.
   For example:

   ```
   chmod 777 arsMSVR.cfg
   chmod 777 arsMSVR.sh
   chmod 777 arswww.ini
   ```

4. Edit the MidServer configuration file, `arsMSVR.cfg`.

   Make the following changes:

   ```
   MIDSERVERPORT=3225
   MIDSERVERCODEPAGE=1047
   MIDSERVERTRACE=0
   XMLTRACEDIR=/urs/lpp/midserver/MStrace/
   ```

   Set the parameters that are outlined in Table 14 on page 238 to match your local environment.

Table 14. Describes the parameters and values that you can modify for the `arsMSVR.cfg` file

Parameter	Default value	Description
MIDSERVERPORT	3225	Identifies the port number that the MidServer monitors for requests transmitted by the Structured API stub program. **Important:** The value that is specified must correspond with the MidServer port number supplied with the LOGON request.
MIDSERVERCODEPAGE	1047	Identifies the code page of the image where the MidServer is running. This code page value is independent of the code page value that is specified with the LOGON request.
MIDSERVERTRACE	0	Identifies the MidServer trace level. If you are running ARSZDAPB, set the MidServer trace level to match the trace level set for **TRACELEVEL** parameter. The **TRACELEVEL** parameter is set by the ARSZDAPB program. If they do not match, the value specified for **TRACELEVEL** parameter overrides the MidServer trace level. The MidServer trace messages are written to the //STDOUT DD specified in the MidServer start procedure that is customized in the next step, “Setting up the MidServer startup procedure” on page 240. **0** Do not display trace statements specific to the MidServer. **1** Display trace statements specific to the MidServer.

Table 14. Describes the parameters and values that you can modify for the arsMSVR.cfg file (continued)

Parameter	Default value	Description
TRACE	NATIVE	Indicates that you want ODWEK to transform or convert the instance of a report, report segment, or other data before completing a RETRIEVE request. If you do not want ODWEK to transform or convert the data, do not specify this parameter or specify NATIVE as the value. To indicate which transformation or conversion you want done, specify one of the following values: **ASCII** ODWEK converts data to ASCII text. **HTML** ODWEK runs data through the AFP to HTML transform. **PDF** ODWEK runs data through the AFP to PDF transform. **XML** ODWEK runs data through the AFP to XML transform.
XMLTRACEDIR	/urs/lpp/midserver/MStrace/	Identifies the directory in which the MidServer XML trace file is created. This parameter must be specified if tracing is enabled via the MIDSERTVER-TRACE parameter.

5. Edit the MidServer shell file, arsMSVR.sh.

   You might have to modify the following lines of the arsMSVR.sh file to conform to your local Java environment:

```
OD_JAVA=
OD_JAVA_OPTIONS="-Dpid=$$ -Dppid=$PPID -Xmx512m -verbose:gc"
export OD_JAVA_OPTIONS
OS=OS390
LIBPATH=$LIBPATH:/usr/lpp/ars/V10R5M0/www
export LIBPATH
OD_JRE=/usr/lpp/java/J5.0
export OD_JRE
OD_JAVA=${OD_JRE}/bin/java
export OD_JAVA
OD_JAVAC=${OD_JRE}/bin/javac
export OD_JAVAC
OD_JAVA_JAR=
export OD_JAVA_JAR
OD_ODAPI=/usr/lpp/ars/V10R5M0/www/api/ODApi.jar
export OD_ODAPI
OD_ODLOGGER=/usr/lpp/ars/V10R5M0/www/api/ODLogger.jar
export OD_ODLOGGER
OD_JAVA_CLASSPATH=${OD_WWW_INTERFACE}:${OD_JAVA_JAR}:${OD_ODAPI}:${OD_ODLOGGER}:
${MY_PATH}/midserver/arsMSVR.jar
export OD_JAVA_CLASSPATH
```

   You might need to modify /usr/lpp/ars/V10R5M0 on the following line of the arsMSVR.sh file to conform to your Content Manager OnDemand mount point:

```
OD_PATH=/usr/lpp/ars/V10R5M0
```

If you installed Content Manager OnDemand Web Enablement Kit in a non-default location in the hierarchical file structure, for example if you specified:

```
OD_PATH=/products/CMoD/new
```

You must add the following lines in `arsMSVR.sh`:

```
ARS_INSTALL_ODWEK_V101_DIR=/products/CMoD/new
export ARS_INSTALL_ODWEK_V101_DIR
```

6. Edit the ODWEK initialization file, `arswww.ini`.

   Modify or add the following lines of the `arswww.ini` file to function in the MidServer environment:

   To enable logging of the ODWEK API information:

   ```
   trace=4
   TraceDir=/usr/lpp/ars/V10R5M0/config/midserver/logs
   ```

   For all releases of Content Manager OnDemand:

   ```
   CodePage=1047
   Language=ENU
   CacheDir=/usr/lpp/ars/V10R5M0/config/midserver/cache
   TempDir=/usr/lpp/ars/V10R5M0/config/midserver/temp
   MaxHits=5000
   ```

   If the MidServer is running under an alternate mount point, all occurrences of `/usr/lpp/ars/V10R5M0` are changed to the alternate mount point.

7. Change the access mode of the files to read only.
   For example:

   ```
   chmod 444 arsMSVR.cfg
   chmod 555 arsMSVR.sh
   chmod 444 arswww.ini
   ```

Results

Important: In step “6” on page 240, the **MaxHits** parameter is the upper limit of the size of the HitList returned for a "HIT LIST" request, regardless of the value that is specified in CS-HitListMax. If more than 5000 hits are needed, the MaxHits value must be changed and the MidServer must be stopped/started. If the MaxHits parameter is not entered, the default is 50.

Setting up the MidServer startup procedure

About this task

The ARSMSVR procedure is used to start the MidServer as a started task.

A sample ARSMSVR procedure is provided in the SARSINST library. You must customize the procedure for your environment and copy the customized procedure in the PROCLIB concatenation.

The following example is the sample ARSMSVR procedure:

```
//ARSMSVR  PROC JPATH='/usr/lpp/ars/V10R5M0/config/midserver'
//ARSMSVR  EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT,
//   PARM='SH'
//STDIN    DD PATH='&JPATH/arsMSVR.sh',PATHOPTS=(ORDONLY)
//STDENV   DD DUMMY
//STDOUT   DD PATH='/tmp/arsMSVR.out',
//            PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
//STDERR   DD PATH='/tmp/arsMSVR.err',
//            PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*
```

where:

//ARSMSVR PROC JPATH='/usr/lpp/ars/V10R5M0/config/midserver'
Specifies the PROC name and Java path label for the MidServer job.

PROC name
You can name your MidServer startup procedure to any name required for local standards. After the name has been chosen, the system programmer must define a RACF STARTED profile to match the name of the MidServer started task (STC) or a generic profile that covers a range of names. The STDATA portion of the profile must map to a RACF user ID and group which also have valid OMVS segments defined to allow UNIX System Services access.

Java path
Label identifying the Java path to be used with subsequent DD statements to satisfy the &JPATH literal. The Java path must specify the location of the MidServer configuration and shell files as well as the required ODWEK files. For more information, see "Preparing the structured API batch driver" on page 242.

//ARSMSVR EXEC PGM=BPXBATCH...PARM='SH'
The MidServer is initiated from a shell running under BPXBATCH. The PARM='SH' statement indicates that the PROC runs a shell. Because no shell name is specified with this parameter, BPXBATCH runs the shell that is specified with the //STDIN DD.

//STDIN DD PATH='&JPATH/arsMSVR.sh', PATHOPTS=(ORDONLY)
Standard input file. The PATH= statement identifies the shell program that starts the Midserver..

//STDENV DD DUMMY
Standard environment file not specified.

//STDOUT DD PATH='/tmp/arsMSVR.out',... PATHOPTS=...,PATHMODE=...
Standard output file. The PATH= statement indicates where MidServer start up and trace messages are stored.

//STDERR DD PATH='/tmp/arsMSVR.err', PATHOPTS=...,PATHMODE=...
Standard error file. The PATH= statement indicates where the Midserver error messages are stored.

//SYSPRINT DD SYSOUT=*
Messages.

//SYSOUT DD SYSOUT=*
Messages.

Procedure

To implement the MidServer PROC, do the following steps:

1. Copy ARSMSVR from SARSINST to the local PROC library where the MidServer runs. The MidServer can run on any z/OS image in your network.
2. Edit the JCL according to local standards.
3. Start the procedure by issuing the following command from the z/OS log: `/S ARSMSVR`

Results

The following example is a sample of the messages written to //STDOUT when the MidServer is started:

```
/usr/lpp/java/J5.0
/usr/lpp/java/J5.0/bin/java
/usr/lpp/java/J5.0/bin/javac
/usr/lpp/ars/V10R5M0/www/api/ODApi.jar:/usr/lpp/ars/V10R5M0/www/api/ODLogger.jar:/usr/lpp/ars/V10R5M0/midserver/arsMSVR.jar
```

You can stop the MidServer by issuing the following command from the z/OS system log: `/C ARSMSVR`. After the cancel command completes, you must purge the MidServer sub-proc by issuing the following command from the z/OS system log: `/P ARSMSVR2`.

Structured API MidServer merged logging

The MidServer creates and maintains a log file for debugging purposes. The parameters **MIDSERTVERTRACE**and **XMLTRACEDIR** specify whether the log file is created and where it will be created. The log data is output in XML format.

This log file is separate from the trace file that is produced by the ODWEK Java API, but the log file can be merged with the trace file. Merging the two files eliminates the need to search through multiple log files and compare time stamps when analyzing a problem. Each trace statement will include basic information such as the date, time, level, and function in formats that are consistent with those utilized by the ODWEK Java API. The Java level is also included in the trace.

The Java Logger class is used to generate the log messages. The J2SE standard formatter XMLFormatter is then used to format the messages so that they can be parsed and processed during the merge. The following XML is a sample of the produced XML data.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ondemand_trace>
   <header>
       <os_platform>z/OS</os_platform>
       <os_version>01.11.00</os_version>
       <od_command>SAPI</od_command>
       <od_version>LI2740 -  Unified logging</od_version>
       <log_created_timestamp>07/08/10 14:38:34</log_created_timestamp>
       <log_created_utc>07/08/10 20:38:34</log_created_utc>
       <log_created_secs>1278621514</log_created_secs>
       <mins_west_of_utc>360</mins_west_of_utc>
    </header>
    <trace_msg>
       <timestamp>07/08/10 14:38:33</timestamp>
       <utc_timestamp>07/08/10 20:38:33</utc_timestamp>
       <secs>1278621513</secs>
       <sub_secs>800000</sub_secs>
       <pid>2413</pid>
       <ppid>1</ppid>
       <tid>10</tid>
       <id>FINEST</id>
       <level>INFO</level>
       <file>ArsMsrvr</file>
       <func>initFromConfiguration</func>
        <line>777</line>
        <text> Starting the MidServer(06/02/2010) (LI2740 Unified logging **)</text>
      </trace_msg>
      <trace_msg>
        <timestamp>07/08/10 14:38:34</timestamp>
        <utc_timestamp>07/08/10 20:38:34</utc_timestamp>
        <secs>1278621514</secs>
        <sub_secs>221000</sub_secs>
        <pid>2413</pid>
        <ppid>1</ppid>
        <tid>10</tid>
        <id>FINEST</id>
        <level>INFO</level>
        <file>ArsMsrvr</file>
        <func>initFromConfiguration</func>
        <line>788</line>
        <text>Done arsMSVR.cfg</text>
      </trace_msg>
```

Preparing the structured API batch driver

About this task

The structured API configuration setup is verified by running the IBM-supplied batch driver program in the next step, “Running the structured API installation verification procedure (IVP)” on page 243. You must compile and link edit the batch driver before it can be executed.

Procedure

To prepare the structured API batch driver, do the following steps using member ARSZDCBJ from the USERJCL library:

1. Add a job card.
2. Change the //PROCESS OUTPUT statement parameters to conform with local standards.
3. Change the operands on the various SET statements to conform with local data set naming standards.
4. Run the job.

Running the structured API installation verification procedure (IVP)

About this task

The batch driver program is run to verify that the various structured API components are set up properly on your systems. For more information about the batch driver, see Chapter 58, “Using the structured APIs,” on page 249.

Setting up batch driver JCL

About this task

The following batch driver JCL is provided in USERJCL member, ARSZDABJ:

```
//*
//* ------------------------------------------------------
//*
//* APAR PQ81183: INITIAL DISTRIBUTION
//* APAR PQ90857: Added "TCPIP.DATA"
//*
//* SMP/E LIBRARY: SARSINST
//* LOCAL LIBRARY: USERJCL
//* MEMBER: ARSZDABJ
//*
//*
//* THIS JCL MAKES USE OF THE ONDEMAND "USER" DATSETS
//* ALLOCATED VIA JCL MEMBERS "ARSUALOC" AND "ARSUALOL".
//*
//*  THIS JCL EXECUTES THE BATCH DRIVER PROGRAM TO RUN THE IVP TEST
//*
//*  CHANGE THE OPERANDS ON THE FOLLOWING SET STATEMENTS TO REFERENCE
//*  THE DATA SET NAMES APPROPRIATE FOR YOU INSTALLATION.
//*
//PROCESS   OUTPUT  DEFAULT=YES,CLASS=*,JESDS=ALL,OUTDISP=HOLD
//USERPRE   SET     USER='ARS.V10R5M0'
//SARSPRE   SET     SARS='ARS.V10R5M0'
//*
//* THE "TCPIP.DATA" FILE IS NEEDED BY THE TCPIP REQUESTS
//* ISSUED BY THE STRUCTURED APIS INTERFACE MODULE
//* "ARSAPIIX". THE "TCPIP.DATA" FILE IS DISCUSSED IN THE
//* TCPIP CUSTOMIZATION AND ADMIN PUBLICATION OR THE Z/OS
//* COMMUNICATIONS SERVER: IP CONFIGURATION REFERENCE AND
//* THE Z/OS COMMUNICATIONS SERVER: IP CONFIGURATION GUIDE
//*
//* CHANGE "TCPIP.DATA" IN THE SYSTCPD SET STATEMENT TO
//* MATCH THE FILE NAME IN YOUR SYSTEM.
//*
//* IF "TCPIP.DATA" IS A PDS, CHANGE "(TCPDATA)" IN THE
//* TCPDATA SET STATEMENT TO MATCH THE MEMBER NAME FOR THE
//* "TCPDATA". IF "TCPIP.DATA" IS A SEQUENTIAL FILE, CHANGE
//* "(TCPDATA)" TO "".
//*
//SYSTCPD SET SYSTCPD='TCPIP.DATA'
//TCPDATA SET TCPDATA='(TCPDATA)'
//*
//* ------------------------------------------------------------------
//*
//STEP1    EXEC PGM=ARSZDAPB,PARM='/ABTERMENC(ABEND)',REGION=0M
//STEPLIB  DD DISP=SHR,DSN=&USER..USERLOAD
//         DD DISP=SHR,DSN=&SARS..SARSLOAD
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=134,BLKSIZE=134)
```

```
//SYSTCPD DD DISP=SHR,DSN=&SYSTCPD.&TCPDATA
//REQFILE  DD DISP=SHR,DSN=&USER..USERPARM(ARSZUIVP)  IVP
//*BULKRTV  DD DISP=SHR,DSN=&USER..USERBLKR (ARSZBIVP) IVP
//SAPIOUT  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//REPORT   DD SYSOUT=*
```

The REQFILE DD identifies the batch driver input file. The SAPIOUT DD contains a log describing the REQFILE input records. The REPORT DD contains informational and error messages regarding the batch driver execution.

Procedure

To set up the batch driver JCL, do the following steps using member ARSZDABJ from USERJCL library:

1. Add a job card.
2. Change the //PROCESS OUTPUT statement parameters to conform with local standards.
3. Change the operands on the various SET statements to conform to local data set naming standards.

Setting up batch driver input file

About this task

The //REQFILE DD statement in ARSZDABJ identifies the driver input file. The batch driver reads the REQFILE file records, formats the data into structured API calls, and invokes the stub program. IBM supplies the following sample batch driver input file in USERPARM member, ARSZUIVP, to use when running the structured API IVP. The values shown in italics must be customized to local standards to run the job in your environment.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
REQUEST           LOGON
MIDSERVER         255.255.255.255
MIDPORT           00000
LIBSERVER         255.255.255.255
LIBPORT           00000
FOLDERNAME        ARSIVP%
TRACELEVEL        1
LOGONID           userid1
PASSWORD          passw1
REQUEST           FOLDER OPEN
FLISTMAX          0
TRACELEVEL        1
LOGONID           userid1
PASSWORD          passw1
REQUEST           FOLDER OPEN
FOLDERNAME        ARSIVPR1
REQUEST           HIT LIST
HITLISTMAX        50
CRITERIA          CUST NAME
SCOP              =
SCVALUE1          COFFEE INC
SCSORTOPT         A
REQUEST           RETRIEVE
RSLOGOFLAG        Y
RSDOCFLAG         D
HIT#              2
REQUEST           RELEASE               REQUEST           LOGOFF
```

Figure 21. Input file in ARSZUIVP member. Italics indicate values that you should change to match your local environment.

Table 15 on page 245 describes the values that you must customize in member ARSZUIVP. The data that you provide is column sensitive. All keywords must begin in column one of the input record and all values must begin in column twenty of the input record.

Table 15. Describes the values you must modify in member ARSUIVP to match your environment

Keyword	Value	Description
MIDSERVER	255.255.255.255	IP address of the Content Manager OnDemand MidServer. A host name might also be used
MIDPORT	00000	IP port number of the MidServer. The text must be five (5) zeroes
LIBSERVER	255.255.255.255	IP address of the Content Manager OnDemand server. A host name might also be used
LIBPORT	00000	IP Port Number of the MidServer. The text must be five (5) zeroes
LOGONID	userid1	Content Manager OnDemand user ID
PASSWORD	passw1	Password for the Content Manager OnDemand user ID
SCVALUE1	11/24/03	Specify a date that has entries in the System Log folder. This value might need to change depending on your Content Manager OnDemand system.

Running the batch driver

About this task

Submit the ARSZDABJ job. ARSZDABJ should end with a condition code of zero. You receive a REPORT DD as a result of a batch driver execution. The REPORT DD contains informational and error messages regarding the batch driver execution. Figure 22 on page 246 is an example of the messages you will receive following a successful execution.

```
1 ARSZDAPB Report of REQFILE input
0* = Edit error of input detected by ARSZDAPB
  ** = Message issued by program ARSZDAPB
  ***= Message returned by the API
    REQUEST LOGON
    MIDSERVER 255.255.255.255
    MIDPORT 00000
    LIBSERVER 255.255.255.255
    LIBPORT 00000
    FOLDERNAME ARSIVP%
    FLISTMAX 0
    TRACELEVEL 1
    LOGONID userid1
    PASSWORD passw1
0***Folder List for 1971251398529-9 successful- user has access to <409>
folders.
    REQUEST FOLDER OPEN
    FOLDERNAME ARSIVPR1
0 ** REQUEST(FOLDER OPEN) started: Mon Feb 12 2007 15:22:04.733
0 ** REQUEST(FOLDER OPEN)   ended: Mon Feb 12 2007 15:22:05.736,
     Elapsed time = 00:00:01.003
0***Open Folder for 1971251398529-9 successful
    REQUEST HIT LIST
    HITLISTMAX 50
    CRITERIA CUST NAME
    SCOP =
    SCVALUE1 COFFEE INC
    SCSORTOPT A
0 ** REQUEST(HIT LIST) started: Mon Feb 12 2007 15:22:05.738
0 ** REQUEST(HIT LIST)   ended: Mon Feb 12 2007 15:22:06.351,
     Elapsed time = 00:00:00.613
0***Hit List for 1971251398529-9 successful- user has access to <38>
hits.
    REQUEST RETRIEVE
    RSLOGOFLAG N
    RSDOCFLAG D
    HIT# 2
0 ** REQUEST(RETRIEVE) started: Mon Feb 12 2007 15:22:06.352
0 ** REQUEST(RETRIEVE)   ended: Mon Feb 12 2007 15:22:23.676,
     Elapsed time = 00:00:17.324
0***Retrieve for 1971251398529-9 successful
    REQUEST RELEASE
0 ** REQUEST(LOGOFF) started: Mon Feb 12 2007 15:22:23.705
0 ** REQUEST(LOGOFF)   ended: Mon Feb 12 2007 15:22:24.263,
     Elapsed time = 00:00:00.558
    REQUEST LOGOFF
0 ** REQUEST(LOGOFF) started: Mon Feb 12 2007 15:22:23.705
0 ** REQUEST(LOGOFF)   ended: Mon Feb 12 2007 15:22:24.263,
     Elapsed time = 00:00:00.558
0***Logoff 1971251398529-9 successful
0 ** ARSZDAPB Normal end of job
```

Figure 22. An example of a REPORT file

Preparing the structured API CICS® driver

About this task

IBM supplies a sample CICS driver program for those customers who need to verify the structure API configuration setup for an application running in CICS. See Section IV of this document for additional information on the CICS driver.

Preparing the CICS® stub program

About this task

You must BIND (link edit) the CICS stub before it can be executed.

Procedure

To BIND (link edit) the CICS stub program, do the following steps using member ARSAPICJ from the USERJCL library:

1. Add a job card.
2. Change the //PROCESS OUTPUT statement parameters to conform to local standards.
3. Change the operands on the various SET statements to conform with local data set naming standards.
4. Run the job.

Assembling the CICS® map

About this task

You must assemble and link edit the CICS map before it can be executed and before the CICS driver can be compiled.

Procedure

To assemble and link the CICS map, do the following steps using member ARSZMP1J from the USERJCL library:

1. Add a job card.
2. Change the //PROCESS OUTPUT statement parameters to conform to local standards.
3. Change the operands on the various SET statements to conform with local data set naming standards.
4. Run the job.

Compiling and linking the CICS® driver program

About this task

You must compile and link edit the CICS driver program before it can be executed.

Procedure

To compile and link the CICS driver program, do the following steps using member ARSZDCCJ from the USERJCL library:

1. Add a job card.
2. Change the //PROCESS OUTPUT statement parameters to conform to local standards.
3. Change the operands on the various SET statements to conform with local data set naming standards.
4. Run the job.

Defining CICS® resources

About this task

You must define one transaction, one profile, one map, and two programs to CICS.

Remember: You must define a profile to use with the SAPI transaction in order to turn off upper case translation

Procedure

To define the resources appropriately, do the following steps:

1. Use RDO to define the SAPI transaction to CICS.

   For example:

```
DEFINE TRANSACTION(SAPI) GROUP(ODSAPI)
DESCRIPTION(ONDEMAND CICS DRIVER TRANSACTION)
```

```
       PROGRAM(ARSZDAPC) TWASIZE(0) PROFILE(ARSPNOUC) STATUS(ENABLED)
       TASKDATALOC(ANY) TASKDATAKEY(USER) STORAGECLEAR(NO)
       RUNAWAY(SYSTEM) SHUTDOWN(DISABLED) ISOLATE(YES) DYNAMIC(NO)
       PRIORITY(1) TRANCLASS(DFHTCL00) DTIMOUT(NO) INDOUBT(BACKOUT)
       RESTART(NO) SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
       CONFDATA(NO) RESSEC(NO) CMDSEC(NO)
```

2. Use RDO to define the SAPI profile to CICS.

   For example:

```
DEFINE PROFILE(ARSPNOUC) GROUP(ODSAPI)
DESCRIPTION(ONDEMAND PROFILE)
      UCTRAN(NO)
```

3. Use RDO to define the stub and CICS Driver program to CICS.

   For example:

```
DEFINE PROGRAM(ARSAPIIC) GROUP(ODSAPI)
DESCRIPTION(ONDEMAND C STUB) LANGUAGE(C) RELOAD(NO) RESIDENT(NO) USAGE(NORMAL)
  USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(ANY) EXECKEY(USER)
  EXECUTIONSET(FULLAPI)
*
DEFINE PROGRAM(ARSZDAPC) GROUP(ODSAPI)
DESCRIPTION(ONDEMAND CICS DRIVER) LANGUAGE(COBOL) RELOAD(NO) RESIDENT(NO)
  USAGE(NORMAL) USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(ANY)
  EXECKEY(USER) EXECUTIONSET(FULLAPI)
*
```

4. Use RDO to define the map to CICS.

   For example:

```
DEFINE MAPSET(ARSZMP1M) GROUP(ODSAPI)
DESCRIPTION(ONDEMAND DRIVER MAP)
       RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO) STATUS(ENABLED)
```

Including USERLOAD library in DHFRPL concatenation

About this task

You must add data set USERLOAD to CICS DFHRPL concatenation. To add the data set USERLOAD, specify the USERLOAD library in the CICS DFHRPL, or, move programs ARSAPIIC and ARSZDAPC and MAPSET ARSZMP1M to a library in the DFHRPL concatenation.

Add SYSTCPD to CICS® JCL

About this task

The JCL statement SYSTCPD must be added to the CICS JCL to support using a host name to access the MidServer. The SYSTCPD data set requirements are the same as described in "Setting up Batch Driver JCL."

Chapter 58. Using the structured APIs

The following sections describe the exchange of information between a customer application program issuing structured API calls and the Content Manager OnDemand server.

The customer application program must be modified to issue specific Content Manager OnDemand function requests, in the form of structured API calls, to the Content Manager OnDemand server. The Content Manager OnDemand functions supported by the structured API calls are:

LOGON
Logs on to the Content Manager OnDemand server and returns a folder list.

FOLDER OPEN
Retrieves folder criteria.

HIT LIST
Builds a list of folder items (search).

RETRIEVE
Retrieves a document.

BULK RTRIEV
Retrieves multiple documents in a single call to the API.

ANNOTATIONS
Retrieves the annotations associated with a document.

LOGOFF
Logs off of the Content Manager OnDemand server, and performs an implicit RELEASE request.

RELEASE
Frees all memory acquired by the various APIs.

RELEASEA
Frees the Annotations List Structure created by ANNOTATIONS.

RELEASEC
Frees the FolderCriteriaStructure created by FOLDER OPEN.

RELEASED
Frees the DocumentStructure created by RETRIEVE.

RELEASEH
Frees the HitListStructure created by HIT LIST.

RELEASEL
Frees the FolderListStructure created by LOGON.

At a minimum, the customer application program must perform these tasks:

- Control the flow of Content Manager OnDemand server requests/responses for each API user
- Include IBM-supplied copybooks that contain the layout of the Content Manager OnDemand server interface structures
- Build a request structure for each Content Manager OnDemand server function
- Issue API call to the Content Manager OnDemand server
- Process the Content Manager OnDemand server response
- Manage storage areas allocated to hold Content Manager OnDemand server response. Storage areas should be freed at the appropriate times during program execution. If you do not free storage, memory usage will continue to increase and it will appear as if the program is suffering from "memory creep."

Content Manager OnDemand server request

All API calls that are issued by the application to the Content Manager OnDemand server include a common structure (common area) of information that is established at LOGON and is maintained throughout the duration of each user dialog. Each user dailog, other than Bulk Retrieve, must begin with the LOGON request. Following LOGON, you must issue the FOLDER OPEN call before you issue the HIT LIST, RETRIEVE, or ANNOTATIONS requests. Multiple HIT LIST, RETRIEVE, and ANNOTATIONS requests can be issued after the folder is opened. You must issue a FOLDER OPEN request to access a different folder.

The BULK RTRIEV request is issued as a single function. It handles logging on, generating a hit list, returning the multiple requested documents, and logging off all in one function call.

At LOGON, a connection token is established in the common structure that associates all subsequent server requests with a particular logged on user. In addition to the connection token, the common structure contains other critical information such as the type of function being requested, the port number and IP address of the Content Manager OnDemand server, the port number and IP address of the mid-tier server as well as pointers to the data returned by the Content Manager OnDemand server to satisfy the request. You can find the common structure layout under the group item name CS-CommonStructure in member ARSZDCWS of the sample library.

Selected function requests require a second structure that contains additional, request-specific information required by the Content Manager OnDemand server. You can also find the request-specific structure layouts in sample library member ARSZDCWS under the following group item names:

- LO-LogonStructure
- SC-CriteriaStructure
- RS-RetrieveStructure
- AR-NotesStructure
- BR-BulkRetrieveStructure

An application can request services from more than one Content Manager OnDemand server asynchronously. Each server that is accessed requires a separate connection (LOGON), and each connection uses a unique set of request structures. It is the application coder's responsibility to create unique copies of the appropriate request structures from copybook ARSZDCWS (CS-CommonStructure, AL-NotesListStructure, LO-LogonStructure, SC-CriteriaStructure, RS-RetrieveStructure, BR-BulkRetrieveStructure) for each asynchronous connection.

Content Manager OnDemand server response

The Content Manager OnDemand server response, except for Bulk Retrieve, is returned in storage areas specific to the server request. Addressability to the returned storage areas is established by pointer fields that are defined in the API common structure, for example:

- CS-pFolderList
- CS-pFolderCriteria
- CS-pHitList
- CS-pDocument
- CS-pNotes

The detailed layouts of the structures returned by the Content Manager OnDemand server can be found in sample library member ARSZDCLS under the following group item names:

- FL-FolderListStructure
- FC-FolderCriteriaStructure
- HL-HitListStructure

- DS-DocumentStructure
- AL-NotesListStructure

The BULK RETRIEV function returns hit lists and documents into the hfs locations specified in the input parameters.

Content Manager OnDemand server flow

Communication with the Content Manager OnDemand server is established by the successful execution of a LOGON request, either explicitly or as part of the BULK RETRIEV function. A successful explicit logon returns a list of folders that are available to the user; the folder list contains only folder names that the user is authorized to access.

Subsequent function requests made using the API calls allow the user to select among the folder names returned at logon by using the FOLDER OPEN request, build a list of document index entries in the opened folder using the HIT LIST request with optional selection criteria, and RETRIEVE documents reflected in the hit list. Communication with the server is ended by successful execution of a LOGOFF request.

LOGOFF can be called directly (in which case all storage area are released) or the LOGOFF request can optionally be embedded in the RETRIEVE request (in which case the document storage areas will not be released). As storage areas become obsolete, the user must issue the appropriate RELEASE requests to free data areas. For each request sent to the Content Manager OnDemand server by using the common area, the API will receive a return code.

As functions are requested, values from prior requests are retained in the API common area and its associated request structures. Details about the fields required to have input values present for the execution of each server function are given in the following descriptions of each server request.

Note: The structures defined in the ARSZDCWS copybook contain fields used by the application program to pass information to the structured APIs and fields used to return information from the structured APIs to the application program reserved fields.

Reserved fields should never be changed by the application program. Return fields should not be changed unless this document describes when such changes are allowed.

The structures defined in copybook ARSZDCLS should not be changed by the application program.

Content Manager OnDemand server function requests

COBOL driver programs, written to execute in both z/OS environments, are provided in the sample library. The programs use the structures discussed in this section to pass data to and receive data from the Content Manager OnDemand server. The sample drivers programs are discussed in detail in Chapter 59, “Structured APIs sample drivers,” on page 283. Please refer to the sample driver programs for additional guidance when implementing the various API calls.

LOGON API

A LOGON request passes a user ID and associated password to the API in order to perform a logon for that user ID to the Content Manager OnDemand server. The user ID and password must be defined on the **General** tab of the user definition in the Content Manager OnDemand administrator database. As part of performing the LOGON function, the API will also request a list of folders from the Content Manager OnDemand server that the user is authorized to access. The folder list can be narrowed by specifying a fully or partially qualified folder name in the CS-FolderName field.

The fields listed in Table 16 on page 252 must be specified by the customer application in the CS-CommonStructure and LO-LogonStructure fields before issuing the LOGON function request.

Table 16. CS-CommonStructure and LO-LogonStructure fields for the LOGON function

Field	Description	Required or optional?
CS-SocketType	TCP/IP socket type. Valid values are: **0** Short socket . The socket is active only for REQUEST. This is the default value. **1** Persistent socket. The socket remains active from LOGON through LOGOFF.	Optional
CS-Eyeball	Common Structure Eyeball value of ARSLSCCO.	Required
CS-Length	Length of the CS-CommonStructure.	Required
CS-APILevel	Reserved for future use.	Reserved
CS-Request	Function request value of LOGON.	Required
CS-MidTierIPAddr	TCP/IP address for the mid-tier server. The "Host Name" representation of the TCP/IP address is supported in all environments.	Required
CS-MidTierPort	TCP/IP port number for the mid-tier server.	Required
CS-LibServIPAddr	TCP/IP address for the Content Manager OnDemand library server. The "Host Name" representation of the TCP/IP address is supported in all environments.	Required
CS-LibServPort	TCP/IP port number for the library server.	Required
CS-CodePage	Content Manager OnDemand server code page. Default value is 500. **Note:** The API does not edit this field for valid code page values.	Required
CS-FolderListMax	An input value of zero requests that the LOGON function return all folder names that the user is authorized to access that match the CS-FolderName input value.	Required
CS-FolderName	Folder name. Valid values are: • blank - return all folder names (up to maximum) the user has permission to access. • Folder name prefix ending with a percent (%) - return all folder names (up to maximum) with this prefix value that the user has permission to access. • full folder name - return only the folder indicated.	Optional

Table 16. CS-CommonStructure and LO-LogonStructure fields for the LOGON function (continued)

Field	Description	Required or optional?
CS-APITraceLevel	Midserver trace flag: **0** No trace **1** Summary tracing **2** Full trace	Required
CS-NumSecs-to-Wait	A non-zero number may be entered to specify the wait time for a LOGON request to complete. Default is 300 seconds. **Note:** Once a non-zero value has been specified, it remains in effect for all subsequent API requests until a new value is specified.	Optional
LO-Eyeball	Logon Structure Eyeball value of ARSZSCLO.	Required
LO-Length	Length of the LO-LogonStructure.	Required
LO-UserID	Content Manager OnDemand user ID. The user ID provided must match a user defined to the Content Manager OnDemand server database.	Required
LO-Password	User password. The password must match the password for the user defined to the Content Manager OnDemand server database for the user ID in field LO-UserID	Required

Table 17 on page 253 and Table 18 on page 254 list the fields that are returned from the Content Manager OnDemand server following the LOGON request. The fields are provided in the CS-CommonStructure and FL-FolderListStructure (pointed to by CS-pFolderList).

Values returned in the storage area addressed by pointer CS-pFolderList include a table whose entries consist of folder names and descriptions. A count field is supplied that indicates the number of entries in the table.

Note: Following a successful LOGON, the customer application program must execute the following statement to gain access to the fields: `Set address of FL-FolderListStructure to CS-pFolderList`

Table 17. CS-CommonStructure fields returned by the LOGON function

Field	Description	Field set by
CS-pFolderList	Pointer to the Content Manager OnDemand folder list. The folder list returned at this address is defined by the fields under group item name FL-FolderListStructure.	API

Table 17. CS-CommonStructure fields returned by the LOGON function (continued)

Field	Description	Field set by
CS-ReturnCode	Function Request Return Code. Valid values are: **0** LOGON and folder list were successful **4 or greater** LOGON and folder list were unsuccessful; examine the return message for further diagnostic information	API
CS-Message	This field contains a text string suitable for display. An informational message might be returned when CS-ReturnCode is equal to zero. **Note:** CS-Message should be set to "spaces" before each Structured APIs request.	API
CS-ConnectionID	Connection identifier associated with the signed-on user.	Content Manager OnDemand server

Table 18. FL-FolderListStructure fields returned by the LOGON function

Field	Description	Field set by
FL-Count	Number of folder names returned. The following fields represent a table of folders (equal to the FL-Count) to which the user has permission to access.	Content Manager OnDemand server
FL-Name	Folder name. This value is returned by selecting **Folders** > **General** > **Folder Name**in the Content Manager OnDemand administrative client.	Content Manager OnDemand server
FL-Description	Folder description. This value is matched against the description found when you select **Folders** > **General** > **Folder Description**in the Content Manager OnDemand administrative client.	Content Manager OnDemand server

LOGON example

Figure 23 on page 255 shows an example of a COBOL call routine to execute a LOGON request.

```
*
* The set of fields in the following lines must be populated for each
* call to perform a LOGON.
*
     Set  CS-RequestLogon              to True
     Move midtieraddressorname         to CS-MidTierIPAddr
     Move midtierportnumber            to CS-MidTierPort
     Move libraryserveraddressorname   to CS-LibServIPAddr
     Move libraryserverportnumber      to CS-LibServPort
     Move foldernamesearcharguement    to CS-FolderName
     Move userid                       to LO-UserID
     Move password                     to LO-Password
*
*  Values for  Fields CS-Eyeball, CS-APILevel, CS-Length, LO-Eyeball,
*  and LO-Length are defined in sample copybook ARSZDCWS.  For
*  re-entrant code, they must be populated dynamically.
*
     Move 'ARSLSCCO'                   to CS-Eyeball
     Move length of CS-CommonStructure to CS-Length
     Move current-level                to CS-APILevel
     Move 'ARSZSCLO'                   to LO-Eyeball
     Move length of LO-LogonStructure  to LO-Length
*
* The set of fields in the following lines are defined in sample
* copybook ARSZDCWS.  For re-entrant code, they must be populated
* dynamically.  Also, if you wish to override any of the default
* values, you may add code to do so.
*
     Move overridecodepage             to CS-CodePage
     Move overridefoldermax            to CS-FolderListMax
     Move overridetracelevel           to CS-APITraceLevel
     Move Spaces                       to CS-Message
*
* A TCP/IP persistent socket may be requested by using a socket
* type value of 1. The default value for CS-SocketType is 0,
* requesting a "short socket" that is active only for REQUEST.
*
     Move sockettypevalue              to CS-SocketType

     Call Structured-API-Program
        using
              CS-CommonStructure
              LO-LogonStructure

     If CS-Message > spaces
       Move CS-Message to Report-error-message
       Write Report-out
     End-if

     If      CS-ReturnCode   =     zero
         and CS-pFolderList  not = Null
         and CS-ConnectionID not = low-values

         Perform Logon-successful
     Else
         Perform Logon-errors
     End-If.
```

Figure 23. Example of a COBOL call routine to execute a LOGON request

FOLDER OPEN API

A FOLDER OPEN request passes a complete folder name to the API in order to perform an open of one specific folder on the Content Manager OnDemand server. This request can only be performed after a successful LOGON. A successful FOLDER OPEN request will return a table of criteria field names (up to a maximum of 128) and associated field information for the specified folder name. The returned criteria field names and logical operators are used to specify search arguments for subsequent HIT LIST and RETRIEVE function requests.

The fields listed in Table 19 on page 256 must be specified by the customer application in the CS-CommonStructure before issuing the FOLDER OPEN function request

Table 19. CS-CommonStructure fields for the FOLDER OPEN function

Field	Description	Required or optional?
CS-Request	Function request value of FOLDER OPEN.	Required
CS-FolderName	Fully qualified folder name. This value is matched against the value found when you select **Folders** > **General** > **Folder Name**in the Content Manager OnDemand administrative client. **Note:** This folder name must match a name returned in the FL-FolderListStructure. Users who run in Content Manager OnDemand Version 2 compatibility mode must make certain the value specified for CS-FolderListMax is large enough to allow the folder name requested for FOLDER OPEN to be returned FL-FolderListStructure.	Required
CS-NumSecs-to-Wait	A non-zero number might be entered to specify the wait time for a FOLDER OPEN request to complete. Default is 120 seconds. **Note:** If a non-zero value is specified, it remains in effect for all subsequent API requests until a new value is specified.	Optional

Table 20 on page 256 and Table 21 on page 257 list the fields that are returned from the Content Manager OnDemand server following the FOLDER OPEN request. The fields are provided in the CS-CommonStructure and FC-FolderCriteriaStructure (pointed to by CS-pFolderCriteria).

Values returned in the storage area addressed by pointer CS-pFolderCriteria include a table whose entries consist of folder criteria names and associated field values. A count field is supplied that indicates the number of entries in the table.

Note: Following the successful FOLDER OPEN, the customer application program must execute the following statement to gain access to the fields: `Set address of FC-FolderCriteriaStructure to CS-pFolderCriteria`

Table 20. CS-CommonStructure fields returned by the FOLDER OPEN function

Field	Description	Field set by
CS-pFolderCriteria	Pointer to the Content Manager OnDemand folder criteria structure. The folder criteria returned at this address is defined by the fields under group item name FC-FolderCriteriaStructure.	API
CS-ReturnCode	Function Request Return Code. Valid values are: **0** FOLDER OPEN was successful **4 or greater** FOLDER OPEN was unsuccessful. Examine the return message for further diagnostic information	API

Table 20. CS-CommonStructure fields returned by the FOLDER OPEN function (continued)

Field	Description	Field set by
CS-Message	This field will contain a text string suitable for display. An informational message may be returned when CS-ReturnCode is equal to zero. **Note:** CS-Message should be set to "spaces" before each Structured APIs request.	API

Table 21. FC-FolderCriteriaStructure fields returned by the FOLDER OPEN function

Field	Description	Field set by
FC-Count	The number of folder criteria entries in the table. The following fields represent a table of folder criteria fields (1 to 128) for the requested folder.	Content Manager OnDemand server
FC-CriteriaName	The folder criteria name. The name correlates to the value you find when you select **Folders** > **Field Information** > **Field Name**in the Content Manager OnDemand administrative client.	Content Manager OnDemand server

Table 21. FC-FolderCriteriaStructure fields returned by the FOLDER OPEN function (continued)

Field	Description	Field set by
FC-PermOPs	The valid search operators for the folder criteria. The appropriate bit will be on for each query operator allowed. **X'0001'** Equal **X'0002'** Not Equal **X'0004'** Less Than **X'0008'** Less Than or Equal **X'0010'** Greater Than **X'0020'** Greater Than or Equal **X'0040'** In (not supported) **X'0080'** Not In (not supported) **X'0100'** Like **X'0200'** Not Like **X'0400'** Between **X'0800'** Not Between The values correlate to the values you find in the **Operators** box when you select **Folders** > **Field Information**in the Content Manager OnDemand administrative client.	Content Manager OnDemand server
FC-DefOP	The default search operator for the criteria. The valid values are the same as those displayed above for field FC-PermOPS. Only one bit is set in this field. The value correlates to what you find in the **Default** section of the **Operator** box when you select **Folders** > **Field Information**in the Content Manager OnDemand administrative client.	Content Manager OnDemand server
FC-CritMaxEntry	The maximum length of the search criteria value.	Content Manager OnDemand server

Table 21. FC-FolderCriteriaStructure fields returned by the FOLDER OPEN function (continued)

Field	Description	Field set by
FC-DefValueCount	The number of default search argument values for this criteria. A criteria that provides default search values is assumed by the APIs to be a required search criteria. **0** No default values **1** Single default value **2** Two default values (supported only with the Between and Not Between operators) This value correlates to what you find when you select **Folders** > **Field Information** > **Default Value 1** in the Content Manager OnDemand administrative client.	Content Manager OnDemand server
FC-DefValue1	The default value of the first search (or only) search criteria value. This value correlates to what you find when you select **Folders** > **Field Information** > **Default Value 1** in the Content Manager OnDemand administrative client.	Content Manager OnDemand server
FC-DefValue2	The default value of the second search criteria value. This value correlates to what you find when you select **Folders** > **Field Information** > **Default Value 2** in the Content Manager OnDemand administrative client.	Content Manager OnDemand server
FC-SortDefault	The default sort value. The valid values are: **A** Ascending **D** Descending This value correlates to what you find in the **Ascending** check box when you select **Folders** > **Field Information** in the Content Manager OnDemand administrative client.	Content Manager OnDemand server

FOLDER OPEN example

The following shows an example of a COBOL call routine to execute a FOLDER OPEN request.

```
*
* The set of fields in the following lines must be populated for
* each call to perform a FOLDER OPEN.
*
     Set  CS-RequestFolderOpen          to True
     Move fullfoldername                to CS-FolderName
     Call Structured-API-Program
          using
                CS-CommonStructure

     If CS-Message > spaces
       Move CS-Message to Report-error-message
       Write Report-out
     End-if
```

```
If      CS-ReturnCode      =     zero
    and CS-pFolderCriteria not = null

    Perform Folderopen-Successful
Else
    Perform Folderopen-Errors
End-If.
```

HIT LIST API

A HIT LIST request passes from the Midserver to the ODWEK API with specific search operators and values for criteria field(s) defined. These values apply to a folder and the request takes place after a successful FOLDER OPEN has been performed on that folder. Criteria field(s) and their valid search operators are found on the field information tab of the folder definition on the Content Manager OnDemand administrative client.

A successful HIT LIST request returns a table of entries in database order which represent instances of report segments, or other data, that have been loaded to the Content Manager OnDemand server database. The HIT LIST returned from the API may be truncated by specifying that a maximum number of hit list entries (MaxHits) be returned. The Midserver always specifies OD_SORT_LOCATION_MIDTIER on the ODWEK Java API `ODFolder.search` call and the Content Manager OnDemand server returns the MaxHits results in database order (not sorted) to ODWEK. ODWEK then sorts the list on the midtier before returning the sorted list to the Midserver. An entry from the returned HIT LIST must be specified for each subsequent RETRIEVE request for the opened folder. For more information, see Appendix C of the ODWEK Java API Reference.

The fields listed in Table 22 on page 260 must be specified by the customer application in the CS-CommonStructure and SC-CriteriaStructure fields before issuing the HIT LIST function request.

Table 22. CS-CommonStructure and SC-CriteriaStructure fields for the HIT LIST function

Field	Description	Required or optional?
CS-Request	Function request value of HIT LIST.	Required
CS-HitListMax	The maximum number of hits to be returned by Content Manager OnDemand server. The valid values are: **0** Return all entries for the requested folder. **N** Numeric value indicating the maximum number of hits that can be returned.	Required
CS-NumSecs-to-Wait	A non-zero number might be entered to specify the wait time for a HIT LIST request to complete. The default is 900 seconds if the current value is zero. **Note:** If a non-zero value is specified, it remains in effect for all subsequent API requests until a new value is specified.	Optional
SC-Eyeball	Search Criteria Eyeball value of ARSZSCSC.	Required
SC-Length	Length of the SC-CriteriaStructure.	Required

Table 22. CS-CommonStructure and SC-CriteriaStructure fields for the HIT LIST function (continued)

Field	Description	Required or optional?
SC-Count	The number of Criteria Fields to be used in the Search. This is the number of entries in the following table. Values passed to the Content Manager OnDemand server on the Selection Criteria storage area include a table of entries composed of seven fields, and a count field that indicates the number of entries in the table.	Required
SC-CriteriaName	The Criteria Name as returned in the FC-CriteriaName field following the FOLDER OPEN.	Required
SC-OP	The single search operator as returned in the FC-DefOP or FC-PermOPS fields following the FOLDER OPEN.	Required
SC-SearchValueCount	The number of search values to be used in the search. The valid values are: **1** Single value (search specifies SC-SearchValue1 only) **2** Two values (search specifies both SC-SearchValue1 and SC-SearchValue2)	Required
SC-ValueLen	The length of the search value(s). This value equals the FC-CritMaxEntry value returned with FOLDER OPEN.	Required
SC-SearchValue1	The first search value	Required
SC-SearchValue2	The second search value (required for Between or Not Between) search value.	Optional
SC-SortOption	The sort option. The valid values are: **A** Ascending **D** Descending	Required

Table 23 on page 262 and Table 24 on page 262 list the fields that are returned from the Content Manager OnDemand server following the HITLIST request. The fields are provided in the CS-CommonStructure and HL-HitListStructure (pointed to by CS-pHitlist).

Values returned in the storage area addressed by pointer CS-pHitList include three major data areas listed in Table 25 on page 264.

Note: Following the successful HIT LIST request, the customer application program must execute the following statement to gain access to the fields: `Set address of HL-HitListStructure to CS-pHitList`

Table 23. CS-CommonStructure fields returned by the HIT LIST function

Field	Description	Field set by
CS-pHitlist	Pointer to the Content Manager OnDemand hit list structure. The hit list returned at this address is defined by the fields under group item name HL-HitListStructure.	API
CS-ReturnCode	Function Request Return Code. Valid values are: **0** HITLIST was successful **4 or greater** HITLIST was unsuccessful. Examine the return message for further diagnostic information.	API
CS-Message	This field contains a text string suitable for display. An informational message might be returned when CS-ReturnCode is equal to zero. **Note:** CS-Message should be set to "spaces" before each Structured API request.	API

Table 24. HL-HitListStructure fields returned by the HIT LIST function

Field	Description	Field set by
HL-Eyeball	The hitlist eyeball ARSZSCHL.	API
HL-Length	The length of hit list storage area.	API
HL-Count	The number of hits returned. This value will not exceed the CS-HitListMax.	API
HL-DocIDsLen	The size (in bytes) of the DocID value storage area returned following the HL-Entry array in HL-DocIDs.	API
HL-CritCount	The number of criteria values returned in each hit list entry.	Content Manager OnDemand server
HL-CritEntry	HL-CritEntry is the group name containing the fields describing the document Criteria. HL-CritEntry is a variable occurrence array of 1 to 128 entries controlled by the value returned in field HL-CritCount.	
HL-CritName	The Criteria Name.	Content Manager OnDemand server
HL-MaxDisplayChars	The maximum length of criteria value that can be displayed.	Content Manager OnDemand server
HL-Entry	HL-Entry is the group name containing the fields describing the Criteria values for each hit. HL-Entry is a variable occurrence array controlled by the value returned in field HL-Count.	
HL-oDocID	Relative position of the DocID value for this hit in the HL-DocIDs area following the HL-Entry array. First DocID value will be at position one.	API

Table 24. HL-HitListStructure fields returned by the HIT LIST function (continued)

Field	Description	Field set by
HL-DocType	The document type. The valid values are: **A** AFP **B** BMPl **C** SCS-EXT **E** E-MAIL **F** JFIF (JPEG) **G** GIF **L** LINE **M** Metacode **N** None **O** ODDOC **P** PDF **Q** PNG **S** SCS **T** TIFF **U** USRDEF **X** PCX **Important:** Document type SCS and SCS-EXT apply to data created on an iSeries system. These document types may be retrieved by the structured APIs by specifying an iSeries library server at LOGON, or through any library server that has an iSeries data server defined.	Content Manager OnDemand server
HL-Values	The hit list values (128 entries) that are associated with the matched search value. The length of each value must be set to either the maximum display length of the associated criteria in array HL-CritEntry or zero if the value represents a null criteria (one not used in this folder).	Content Manager OnDemand server

Table 24. HL-HitListStructure fields returned by the HIT LIST function (continued)

Field	Description	Field set by
HL-SlackBytes	HL-SlackBytes is a variable length field whose length must be calculated by the application program. The HL-HitListStructure is built by a C program, and the numeric binary fields in the structure will be placed on the boundary alignment associated with the field type. COBOL has does not support this alignment for variable length groups, so the field alignment must be controlled by programming. A programming sample is provided in the copybook ARSZDCLS.	Calculated by the application program
HL-DocIDLen	The length of the HL-DocID for this hit list entry.	API
HL_NotesIndicator	Specifies the number of text annotations for this document. (The Structured APIs does not have access to image annotations.)	API
HL_DocIDs	A variable length field containing all the HL-DocID values for all the HLEntry rows. The length of the field is controlled by the value in field HLDocIDsLen.	
HL-DocIDs	A variable length field containing all the HL-DocID() values for the HitList. HL-DocID(1) begins at position one and is of length HL-DocIdLen(1). For all other HL-DocID(*n*), HL-oDocID(*n*) is equal to HL-oDocID(*n*-1) + HL-DocIDLen(*n*-1), and its length is HL-DocIDLen(*n*).	API

Table 25. Values returned in the storage area addressed by pointer CS-pHitList

Field	Description
HL-CritEntry	An variable length array whose entries consist of the search criteria fields defined to the folder (1 to 128 entries). The number of entries is controlled by the value in HL-CritCount.
HL-Entry	A variable length array containing an entry for each row in the Content Manager OnDemand application group table that matched the search criteria. The number of entries is controlled by the value in HL-Count. Each entry in the table defines all criteria values, but the criteria values corresponding to the criteria fields defined in HL-CritEntry will contain values. Criteria values not used for the Hit List must have their lengths set to zero.
HL-DocIDs	A variable length field containing all the HL-DocID values for all the HL-Entry rows. The length of the field is controlled by the value in field HL-DocIDsLen.

HIT LIST example

The following shows an example of a COBOL call routine to execute a HIT LIST request.

```
Hit-List-Routine.
*
* The set of fields in the following lines must be populated for
* each call to perform a HIT LIST request.
*
    Set CS-RequestHitList               to True
*
* The set of fields in the following lines must be populated for each
* criteria field you want to use to specify selection values, for any
* critera N its appropriate search value(s), search operand and sort
* option.  Field SC-Count will communicate how many criteria are
* being passed to the OnDemand system.
*
    Set address of FC-FolderCriteriaStructure
        to CS-pFolderCriteria

    Move 0                              to SC-Count
    Perform varying FC-Ndx from 1 by 1 until FC-Ndx > FC-Count
     If FC-CriteriaName(FC-Ndx)   =  criterianameN
       Add 1                            to SC-Count
       Set SC-Ndx                       to SC-Count
       Move FC-CriteriaName(FC-Ndx)     to SC-CriteriaName(SC-Ndx)
       Move FC-CritMaxEntry(FC-Ndx)     to SC-ValueLen(SC-Ndx)
       Move searchvalue1N               to SC-SearchValue1(SC-Ndx)
       Move searchvalue2N               to SC-SearchValue2(SC-Ndx)
       Move searchcountN                to SC-SearchValueCount(SC-Ndx)
       Move opcodeN                     to SC-OP(SC-Ndx)
       Move FC-SortDefault(FC-Ndx)      to SC-SortOption(SC-Ndx)
     End-if
    End-Perform
*
* Limiting the number of hits returned by overriding the default
* number of max hits is optional
*
    Move hitlistmaxoverride to CS-HitListMax
    Move Spaces             to CS-Message

    Move length of SC-CriteriaStructure  to SC-Length
    Call Structured-API-Program
         using
               CS-CommonStructure
               SC-CriteriaStructure
    If CS-Message > spaces
       Move CS-Message to Report-error-message
       Write Report-out
     End-if

     If     CS-ReturnCode =     zero
       and CS-pHitList   not = null
       Perform Hit-List-Successful
     Else
       Perform Hit-List-Errors
     End-If.
```

Figure 24. Example of a COBOL call routine to execute a HIT LIST request

RETRIEVE API

A RETRIEVE request passes "hit" data from a table entry returned by a prior successful HITLIST request. A successful RETRIEVE request returns an instance of a report, report segment, or other data that is loaded to the Content Manager OnDemand server archive.

Multiple RETRIEVE requests might be necessary to return all the data for a document. If the document has been loaded (captured) as a large object, it is stored as multiple segments. The document type (large object or single segment) is not known until the completion of the initial RETRIEVE request, and the field RS-DocumentType must be checked to determine document type.

For a large object, each segment must be retrieved by a separate RETRIEVE request. Each RETRIEVE request implicitly frees the storage obtained for the previous RETRIEVE request. The fields RS-SegmentCount, RS-SegmentRetrieved, and RS-SegmentNext must be initialized to zero before the first RETRIEVE request for a document, and will have values returned only for a large object document.

The segment numbers for large object documents are zero based, and the first segment returned is segment zero. Following a RETRIEVE request RS-SegmentRetrieved contains the segment number of the

retrieved segment, and RS-SegmentNext contains RSSegmentRetrieved+1. To retrieve large object segments sequentially, subsequent RETRIEVE requests do not need to alter the RS-SegmentNext value. If a specific segment is desired, set the value of RS-SegmentNext to the segment number before the next RETRIEVE request.

There is no explicit end of document indicator for a large object document. The end of document is recognized when the value of RS-SegmentNext equals the value of RS-SegmentCount.

The fields listed in Table 26 on page 266 must be specified by the customer application in the CS-CommonStructure and RS-RetrieveStructure fields before issuing the RETRIEVE function request.

Table 26. CS-CommonStructure and RS-RetrieveStructure fields for RETRIEVE function

Field	Description	Req / Opt
CS-Request	Function request value of RETRIEVE.	Required
CS-NumSecs-to-Wait	A non-zero number may be entered to specify the wait time for a RETRIEVE request to complete. Default is 1800 seconds. **Note:** If a non-zero value is specified, it remains in effect for all subsequent API requests until a new value is specified.	Optional
RS-Eyeball	Retrieve Structure Eyeball value of ARSZSCRS.	Required
RS-Length	The length of the retrieve storage area.	Required
RS-SegmentCount	The segment count for large object documents. Initialize this value to zero for the first API RETRIEVE call. Subsequent calls should not alter this value.	Required
RS-SegmentRetrieved	The segment number of the large object segment retrieved. Initialize this value to zero for the first API RETRIEVE call. Subsequent calls should not alter this value.	Required
RS-SegmentNext	The next segment number to be retrieved for documents stored as large object documents. For targeted large object document retrievals, the value might be changed to select a specific large object document segment.	Required
RS-LogoffFlag	Flag indicating whether LOGOFF request should be issued after the document is retrieved. The valid values are: **Y** Perform LOGOFF request after the document is retrieved. This will release all memory except for the document memory. When appropriate, you will need to perform a RELEASED request to release the document memory. **N** Do not perform LOGOFF after the document is retrieved. No memory is released.	Required

Table 26. CS-CommonStructure and RS-RetrieveStructure fields for RETRIEVE function (continued)

Field	Description	Req / Opt
RS-DocumentFlag	For documents stored as Large Documents, indicates whether the document should be retrieved as a single document (all pieces combined) or as separate pieces. The valid values are: **D** Retrieve the Large Document as a single document **S** Retrieve the Large Document as multiple pieces	Reserved for future use
RS-DocIDLen	The document length.	Required
RS-DocID	The document identifier.	Required

Table 27 on page 267, Table 28 on page 268, and Table 29 on page 269 list the fields that are returned from the Content Manager OnDemand server following the RETRIEVE request. The fields are provided in the CS-CommonStructure, RS-RetrieveStructure, and DS-DocumentStructure (pointed to by CS-pDocument).

Note: Following the successful RETRIEVE request, the customer application program must execute the following statement to gain access to the fields: `Set address of DS-Document Structure to CS-pHitList`

The values returned in the storage area addressed by pointer CS-pDocument include a field of 1000000 bytes and a length field that indicates the length of the returned document storage area.

Table 27. CS-CommonStructure Fields returned by RETRIEVE function

Field	Description	Field set by
CS-pDocument	Pointer to the Content Manager OnDemand document structure. The document structure returned at this address is defined by the fields under group item name DS-DocumentStructure.	API
CS-ReturnCode	Function Request Return Code. Valid values are: **0** RETRIEVE was successful **4 or greater** RETRIEVE was unsuccessful. Examine the return message for further diagnostic information	API
CS-Message	This field contains a text string suitable for display. An informational message might be returned when CS-ReturnCode is equal to zero. **Note:** CS-Message should be set to "spaces" before each Structured APIs request.	API

Table 28. RS-RetrieveStructure Fields returned by RETRIEVE function

Field	Description	Field set by
RS-DocumentFlag	The value returned indicates whether the document being retrieved is a single segment document or a large object document. The valid values are: **D** Single segment document **S** Large object document	
RS-SegmentCount	Number of segments in the large object document	
RS-SegmentRetrieved	Number of segments in the large object document	
RS-SegmentNext	Number of next segment to be retrieved for large object document	
RS-DocLen	The length of the document segment that has been returned by a RETRIEVE request in field DS-Document.	Content Manager OnDemand server
RS-RecLen	For fixed length line data documents, this is the length of a line within the retrieved document.	Content Manager OnDemand server
RS-RecDelimLen	For line data documents only, this is the length of the delimiter string.	Content Manager OnDemand server
RS-LastSegmentFlag	Reserved for future use.	
RS-LargeDocFlag	Reserved for future use.	
RS-AFPDataFlag	For AFP data, the value returned indicates when the retrieved data is an AFP resource or AFP data segment. The valid values are: **D** AFP data **R** AFP resource	Content Manager OnDemand server
RS-RecFormat	The value returned indicates the format of the retrieved line data document. The valid values are: **F** Fixed **V** Variable **S** Stream	Content Manager OnDemand server

Table 28. RS-RetrieveStructure Fields returned by RETRIEVE function (continued)

Field	Description	Field set by
RS-CC	The value returned indicates the type of carriage control in the retrieved document. The valid values are: **A** ANSI **M** Machine **N** No carriage control	Content Manager OnDemand server
RS-TRC	The value returned from the API in this field indicates whether the document uses TRC codes. The valid values are: **1** Uses TRC codes **0** Does not use TRC codes	Content Manager OnDemand server
RS-RecDelim	The value returned from the API in this field is the delimiter for stream data documents.	Content Manager OnDemand server

Table 29. DS-DocumentStructure Fields returned by RETRIEVE function

Field	Description	Field set by
DS-Length	The length of the document storage area.	Content Manager OnDemand server
DS-Document	The document segment returned by a RETRIEVE request. Individual bytes in this string are addressable using field name DS-DocChar indexed by DS-NDX. DS-Document is a variable length character field whose length is determined by the value in field DS-Length.	Content Manager OnDemand server

RETRIEVE example

The following shows an example of a COBOL call routine to execute a RETRIEVE request.

```
RETRIEVE-ROUTINE.
*
* The request field must be populated for each call to perform a
* RETRIEVE.
*
     Set  CS-RequestRetrieve             to True
*
*  Values for  Fields RS-Eyeball and RS-Length are defined in sample
*  copybook ARSZDCWS.  For re-entrant code, they must be populated
*  dynamically.
*
     Move 'ARSZSCRS'                to RS-Eyeball
*
* These 3 numeric fields must be initialized to zero before the
* retrieve call.
*
     Move 0  to RS-SegmentCount RS-SegmentRetrieved RS-SegmentNext
*
*  Fields RS-LogoffFlag and RS-DocumentFlag must be initialized
*  before the retrieve call.
*
     Set RS-LogoffNo                     to True
     Set RS-SingleDoc                    to True
*
* For the selected Hit List entry N the document id and length must
* be filled in.
*

     Set address of HL-HitListStructure
         to CS-pHitList

     Move HL-DocIDLen(N)                 to RS-DocIDLen
     Move HL-oDocID(N)                   to Tally
     Move HL-DocIDs(Tally:RS-DocIdLen)   to RS-DocID
     Move Spaces                         to CS-Message

     Compute RS-Length = length of RS-RetrieveStructure
     Call Structured-API-Program
         using
             CS-CommonStructure
             RS-RetrieveStructure

     If CS-Message > spaces
       Move CS-Message to Report-error-message
       Write Report-out
     End-if

     If     CS-ReturnCode =     0
        and CS-pDocument  not = null
        Perform Retrieve-Successful
     Else
        Perform Retrieve-Errors
     End-If.
```

Figure 25. Example of a COBOL call routine to execute a RETRIEVE request

ANNOTATIONS API

An ANNOTATIONS request passes “hit” data from a table entry returned by a prior successful HIT LIST request. A successful ANNOTATIONS request returns a structure containing all the text annotations for a document.

The fields listed in Table 30 on page 270 must be specified by the customer application in the CSCommonStructure and AR-NotesStructure before issuing the ANNOTATIONS function request.

Table 30. CS-CommonStructure and AR-NotesStructure fields for ANNOTATIONS function

Field	Description	Required or optional?
CS-Request	Function request value of ANNOTATIONS.	Required

Table 30. CS-CommonStructure and AR-NotesStructure fields for ANNOTATIONS function (continued)

Field	Description	Required or optional?
CS-NumSecs-to-Wait	A non-zero number might be entered to specify the wait time for an ANNOTATIONS request to complete. The default is 120 seconds. **Note:** If a non-zero value is specified, it remains in effect for all subsequent API requests until a new value is specified.	Optional
AR-Eyeball	Structure Eyeball value of ARSZSCAN	Required
AR-Length	The length of the AR-NotesStructure	Required
AR-DocIDLen	The document identifier length	Required
AR-DocID	The document identifier	Required

Table 31 on page 271, and Table 32 on page 271 list the fields that are returned from the Content Manager OnDemand server following the ANNOTATIONS request. The fields are provided in the CS-CommonStructure and AL-NotesListStructure (pointed to by CS-pNotes).

Note: Following the successful ANNOTATIONS request, the customer application program must execute the following statement to gain access to the fields: `Set address of AL-NotesListStructure to CS-pNotes`.

Table 31. CS-CommonStructure Fields returned by ANNOTATIONS function

Field	Description	Field set by
CS-pNotes	Pointer to the Content Manager OnDemand Annotations List Structure. The Annotations List Structure returned at this address is defined by the fields under group item name AL-NotesListStructure.	API
CS-ReturnCode	Function Request Return Code. Valid values are: **0** ANNOTATIONS was successful **4 or greater** ANNOTATIONS was unsuccessful; examine the return message for further diagnostic information	API
CS-Message	This field contains a text string suitable for display. An informational message might be returned when CS-ReturnCode is equal to zero. **Note:** CS-Message should be set to “spaces” before each Structured APIs request.	API

Table 32. AL-NotesListStructure Fields returned by ANNOTATIONS function

Field	Description	Field set by
AL-Length	The length of the AL-NotesListStructure	API
AL-Count	Number of text annotations returned	API

Table 32. AL-NotesListStructure Fields returned by ANNOTATIONS function (continued)

Field	Description	Field set by
AL-NotesLen	The length of the AL-NotesText field following the array of notes descriptions	API
AL-Entry	AL-Entry is the group name for the array entries that describe the annotations returned. AL-Entry is a variable occurrence array, the number of entries depending on the value in AL-Count	
AL-oNote	Starting position in AL-NotesText for the text of this annotation	API
AL-Page	Page number for this annotation	API
AL-OffsetY	The vertical position of this annotation, where position 0 is the top of the page.	Content Manager OnDemand server
AL-OffsetX	The horizontal position of this annotation, where position 0 is the left margin.	Content Manager OnDemand server
AL-Number	This number represents the relative age of the annotation, from oldest to newest.	API
AL-UserID	The user ID of the operator that created this annotation	Content Manager OnDemand server
AL-GroupName	The Group Name to which this userid is assigned	Content Manager OnDemand server
AL-DateTime	The time and date the annotation was created	Content Manager OnDemand server
AL-Type	This field is not currently supported by the Java API interface used by the structured APIs. When available it identifies the type of the annotation as text or image. The Java APIs only support text annotations.	
AL-Color	The color of the annotation tag that appears on the GUI Client.	Content Manager OnDemand server
AL-Copy	OK to copy annotation to another server **0** No **1** Yes	Content Manager OnDemand server
AL-Public	Annotation is Public **0** No **1** Yes	Content Manager OnDemand server
AL-NoteLen	Length of the text for this annotation in AL-NotesText	API
AL-NotesText	AL-NotesText contains the text for all the annotations.	

Table 32. AL-NotesListStructure Fields returned by ANNOTATIONS function (continued)

Field	Description	Field set by
AL-NotesText	The length of this variable character field depends on the value in AL-NotesLen. The text for each annotation will begin at position AL-oNote, for length AL-NoteLen.	API

ANNOTATIONS example

The following shows an example of a COBOL call routine to execute an ANNOTATIONS request.

```
NOTES-ROUTINE.
*
* The Annotations Structure must be populated for each call to
* perform an ANNOTATIONS request
*
* Value for fields AR-Eyeball is defined in sample copybook ARSZDCWS.
*
For reentrant code, it must be populated dynamically.
* Move 'ARSZSCAN' to AR-Eyeball
*
* For the selected Hit List entry "N" the document id and length must
  be filled in.
*
        Set address of HL-Hit List Structure to CS-pHitList
        Set CS-RequestAnnotations to True
        Move HL-DocIDLen(N) to AR-DocIDLen
        Move HL-oDocID(N) to Tally
        Move HL-DocIDs(Tally:AR-DocIdLen) to AR-DocID
        Move Spaces to CS-Message

        Call Structured-API-Program
             using
                  CS-CommonStructure
                  AR-NotesStructure

        If CS-Message > spaces
           Move CS-Message to Report-error-message
           Write Report-out
        End-if

        If CS-ReturnCode = 0 and CS-pNotes not = null
           Perform Annotations-Successful
        Else
           Perform Annotations-Errors
        End-If.
```

LOGOFF API

A LOGOFF request passed to the API requests that the user who was logged on by the prior successful LOGON request be logged off of the Content Manager OnDemand server.

Table 33 on page 273 lists the field that must be specified by the customer application in the CS-CommonStructure before issuing the LOGOFF function request.

Table 33. CS-CommonStructure field for the LOGOFF function

Field	Description	Required or optional?
CS-Request	Function request value of LOGOFF.	Required
CS-NumSecs-to-Wait	A non-zero number may be entered to specify the wait time for a LOGOFF request to complete. Default is 120 seconds. **Note:** Once a non-zero value has been specified, it will remain in effect for all subsequent API requests until a new value is specified.	Optional

Table 34 on page 274 lists the fields that are returned from the Content Manager OnDemand server following the LOGOFF request. The fields are provided in the CS-CommonStructure.

Table 34. CS-CommonStructure Fields returned by the LOGOFF function

Field	Description	Field set by
CS-ReturnCode	Function Request Return Code. Valid values are: **0** LOGOFF was successful **4 or greater** LOGOFF was unsuccessful; examine the return message for further diagnostic information	API
CS-Message	This field will contain a text string suitable for display. An informational message may be returned when CS-ReturnCode is equal to zero. **Note:** You should always set CS-Message to "spaces" before each structured APIs request.	API

LOGOFF example

The following shows an example of a COBOL call routine to execute a LOGOFF request.

```
*
* The request field must be populated for each call to perform a LOGON.
*
     Set CS-RequestLogoff  to True
     Move Spaces           to CS-Message

     Call Structured-API-Program
         using
             CS-CommonStructure.

     If CS-Message > spaces
       Move CS-Message to Report-error-message
       Write Report-out
     End-if

     If     CS-ReturnCode =     0
        Perform Logoff-Successful
     Else
        Perform Logoff-Errors
     End-If.
```

RELEASE (RELEASEA, RELEASEC, RELEASED, RELEASEH, RELEASEL) API

A RELEASE request passed to the API requests that the storage areas obtained by the API and returned to the application be freed. The storage areas implicitly freed by this request are:

- FL-FolderListStructure
- FC-FolderCriteriaStructure
- HL-HitListStructure
- DS-DocumentStructure
- AL-Annotations List Structure

If a RETRIEVE request with the implicit logoff flag set to Y is processed, a LOGOFF call is made and invokes a RELEASE request after the document is retrieved. The RELEASE request then frees all storage areas except the Document Structure. You need to make a separate RELEASED request to release the Document Structure.

A RELEASEA request passed to the API requests the AL-Annotations List Structure created in a prior ANNOTATIONS request be freed.

A RELEASEC request passed to the API requests the FC-FolderCriteriaStructure created in a prior FOLDER OPEN request be freed.

A RELEASED request passed to the API requests the DS-DocumentStructure created in a prior RETRIEVE request be freed.

A RELEASEH request passed to the API requests the HL-HitListStructure created in a prior HIT LIST request be freed.

A RELEASEL request passed to the API requests the FL-FolderListStructure created in a prior LOGON request be freed.

Table 35 on page 275 lists the field that must be specified by the customer application in the CS-CommonStructure before issuing the RELEASE function request.

Table 35. CS-CommonStructure field for the RELEASE function

Field name	Description	Required or optional?
CS-Request	Function request value of RELEASE, RELEASEA, RELEASEC, RELEASED, RELEASEH, or RELEASEL	Required

Table 36 on page 275 lists the fields that are returned from the Content Manager OnDemand server following the Release request. The fields are provided in the CS-CommonStructure.

Table 36. CS-CommonStructure fields returned by the Release function

Field	Description	Field set by
CS-ReturnCode	Function Request Return Code. Valid values are: **0** RELEASE was successful **4 or greater** RELEASE was unsuccessful; examine the return message for further diagnostic information	API
CS-Message	This field will contain a text string suitable for display. An informational message may be returned when CS-ReturnCode is equal to zero. **Note:** You should always set CS-Message to "spaces" before each structured APIs request.	API

RELEASE example

The following shows an example of a COBOL call routine to execute a RELEASE request.

```
*
* The request field must be populated for each call to perform
* a RETRIEVE.
*
     Set  CS-RequestRelease             to True
     Move Spaces                        to CS-Message

     Call Structured-API-Program
         using
             CS-CommonStructure

     If CS-Message > spaces
       Move CS-Message to Report-error-message
       Write Report-out
     End-if
```

```
If     CS-ReturnCode =     0
   Perform Release-Successful
Else
   Perform Release-Errors
End-If.
```

RELEASEA, RELEASEC, RELEASED, RELEASEH and RELEASEL APIs
A RELEASEA, RELEASEC, RELEASED, RELEASEH or RELEASEL request passed to the API requests that a specific storage area obtained for a returned structure in the execution of a prior request be freed. Other than the request value placed in CS-Request, these requests use the same CommonStructure fields as the RELEASE request. Below are the differences for each request.

1. For a RELEASEA the Annotations List Structure will be freed. The following COBOL statement sets the request: `Set CS-RequestReleaseA to True`
2. For a RELEASEC the FolderCriteriaStructure will be freed. The following COBOL statement sets the request: `Set CS-RequestReleaseC to True`
3. For a RELEASED the Document Structure will be freed. The following COBOL statement sets the request: `Set CS-RequestReleaseD to True`
4. For a RELEASEH the HitListStructure will be freed. The following COBOL statement sets the request: `Set CS-RequestReleaseH to True`
5. For a RELEASEL the FolderListStructure will be freed. The following COBOL statement sets the request: `Set CS-RequestReleaseL to True`

BULK RETRIEVE API

A BULK RETRIEV request provides for a single function call that performs all of the required actions to generate the retrieval of multiple documents. A query is specified in the BR-SQL parameter. The hit list associated with a query is placed in the HFS file pointed to by the BR-FListName parameter. The documents retrieved for the hit list are placed in the HFS file pointed to by the BR-ThreadDir parameter. Up to 10 sets of values can be specified for these three parameters.

You need to delete previous hit lists and documents stored by the Bulk Retrieve API into the directories specified in the BR-FListNames and BR-ThreadDir parameters before each execution of Bulk Retrieve. If multiple Bulk Retrieves are to be run concurrently, each must specify unique directories.

The Bulk Retrieve function can be multi-threaded. For each different value specified in the BR-ThreadDir parameter, a new thread is created. All queries associated with the same BR-ThreadDir parameter value are run under the same thread.

The BR-NumSQL parameter controls how many sets of values are specified for the BR-FListName, BR-ThreadDir, and BR-SQL parameters. The BR-NumThread parameter identifies how many different values are specified for the BR-ThreadDir parameters.

The documents returned in the HFS file or files specified in the BR-ThreadDir parameters must be parsed to reconstruct the documents. If the document was stored with fixed length records, then that record length value must be used to parse the records in the hfs file. If the document was stored with variable length records, then each logical record in the hfs physical records is preceded by a two-byte length value. This length value can be used to parse the document records.

The fields described in Table 37 on page 276 must be specified by the customer application in the CS-CommonStructure and BR-BulkRetrieveStructure fields before issuing the BULK RETRIEV function request.

Table 37. CS-CommonStructure and BR-BulkRetrieveStructure fields for BULK RETRIEV function

Field	Description	Required or optional?
CS-Eyeball	Common Structure Eyeball value of ARSLSCCO.	Required

Table 37. CS-CommonStructure and BR-BulkRetrieveStructure fields for BULK RETRIEV function (continued)

Field	Description	Required or optional?
CS-Length	Length of the CS-CommonStructure.	Required
CS-APILevel	CS-APILevel API level value of 8.4.	Required
CS-Request	Function request value of BULK RETRIEV.	Required
CS-MidTierIPAddr	TCP/IP address for the mid-tier server. The 'Host Name' representation of the TCP/IP address is supported in all environments.	Required
CS-MidTierPort	TCP/IP port number for the mid-tier server.	Required
CS-LibServIPAddr	TCP/IP address for the Content Manager OnDemand library server. The 'Host Name' representation of the TCP/IP address is supported in all environments.	Required
CS-LibServPort	TCP/IP port number for the library server.	Required
CS-CodePage	The Content Manager OnDemand Server code page. Default value is 500. **Important:** The API does not edit this field for valid code page values.	Required
CS-FolderName	A full folder name; returns only the folder specified.	Required
CS-APITraceLevel	Trace flag: **0** No trace **1** Summary tracing **2** Full trace	Required
CS-SocketsRequested	Number of sockets requested. If unspecified, blank, or less than 100, the default is 1200.	Optional

Table 37. CS-CommonStructure and BR-BulkRetrieveStructure fields for BULK RETRIEV function (continued)

Field	Description	Required or optional?
CS-NumSec-to-Wait	A non-zero number that is entered to specify the wait time for a send/receive request to complete. The default is 300 seconds. **Important:** After a non-zero value is specified, it remains in effect for all subsequent API requests until a new value is specified.	Optional
Bulk Retrieve Structured Fields		
BR-Eyeball	Logon Structure Eyeball value of ARSZSCBR.	Required
BR-Length	Length of the BR-BulkRetrieveStructure.	Required
BR-UserID	Content Manager OnDemand user ID. The user ID provided must match a user defined in the Content Manager OnDemand server database for the user ID in field LO-UserID.	Required
BR-Password	User password. The password must match the password for the user defined in the Content Manager OnDemand server database for the user ID in field LO-UserID.	Required
BR-AG	Application group name.	Required
BR-Language	Language. Default is en.	Optional
BR-Country	Country code. Default is US.	Optional
BR-MaxHits	Maximum number of hits to be returned by SQL.	Required

Table 37. CS-CommonStructure and BR-BulkRetrieveStructure fields for BULK RETRIEV function (continued)

Field	Description	Required or optional?
BR-DiskW	**0** Write retrieved document data directly to disk. Uses less memory and is slightly faster, but the program fails if the same document (document name) is retrieved twice on the same thread or the same directory. BR-DiskW can be used only if all document names on a thread or in a directory are unique. **1** Retrieves document data in memory, then writes the file (document data) to disk. Uses more memory and is slightly slower. This option overwrites existing files if they already exist; thus, if the same file (file name) is downloaded 10 times (on the same thread) only one copy remains in the designated directory.	Required

Table 37. CS-CommonStructure and BR-BulkRetrieveStructure fields for BULK RETRIEV function (continued)

Field	Description	Required or optional?
BR-Debug	Debugging parameters: **0** No debug statements **1** Print thread statistics summary **2** Print thread statistics detail **3** Print configuration file values that are read **4** Echo SQL arrays that are read from the configuration file **5** Print thread startup values (in main) **6** General debug ArsBulkRetrieve class **7** Debug document retrieval Each higher debug level includes debugging statements from all lower levels.	Required
BR-SQLSec	Wait time between thread startups.	Required
BR-NumThread	The number of threads. An integer between 1 and 10.	Required
BR-NumSQL	The number of SQL statements to be executed.	Required
The structure contains locations for 10 of each of the following three fields. The number of entries that will be read is equal to the BR-NumSQL value.		
BR-FListName	File name that will contain a list of the files retrieved by the SQL statement. There is one entry for each SQL statement. If BR-NumSQL is *n*, then there must be *n* entries, one file per thread.	Required

Table 37. CS-CommonStructure and BR-BulkRetrieveStructure fields for BULK RETRIEV function (continued)		
Field	**Description**	**Required or optional?**
BR-ThreadDir	The directory containing the output when a SQL statement is executed by a thread. There is one entry for each SQL statement. If BR-NumSQL is *n*, there must be *n* entries. SQL statements with the same directory name execute under the same thread.	Required
BR-SQL	SQL statement to be executed. There is one entry for each SQL statement. If BR-NumSQL is *n*, there must be *n* entries.	Required

Table 38 on page 281 describes the fields that are returned from the Content Manager OnDemand server following the BULK RETRIEV request. The fields are provided in the CS-CommonStructure.

Table 38. CS-CommonStructure fields returned by BULK RTRIEV function		
Field	**Description**	**Set by**
CS-ReturnCode	Function request return code. Valid values are: **0** RETRIEVE was successful. **4 or greater** RETRIEVE failed. Examine the return message for further diagnostic information.	API
CS-Message	This field contains a text string suitable for display. An informational message might be returned when CS-ReturnCode = 0. **Important:** CS-Message should be set to spaces before each structured API request.	API

Bulk retrieve example

This is an example of a COBOL call routine to execute a BULK RTRIEV request.

```
BULK RETRIEVE ROUTINE
*
* The set of fields in the following lines must be populated for each call
* to perform a bulk retrieve.
*
    Set CS-RequestBulkRetrieve to True
    Move midtieraddressorname                        to CS-MidTierIPAddr
    Move midtierportnumber                           to CS-MidTierPort
    Move libraryserveraddressorname                 to CS-LibServIPAddr
    Move libraryserverportnumber                     to CS-LibServPort
    Movefoldernamesearchargument                     to CS-FolderName
    Move userid                                             to BR-UserID
```

```
    Move password                                          to BR-Password
*
* Values for fields CS-Eyeball, CS-APILevel, CS-Length, BR-Eyeball, and
* BR-Length are defined in a sample copybook ARSZDCWS. For re-entrant code,
* they must be populated dynamically.
*
    Move "                                                  to CS-Eyeball
    Move length of CS-CommonStructure            to CS-Length
    Move current-level                                to CS-APILevel
    Move "                                                  to BR-Eyeball
    Move length of BR-BulkRetrieveStructure    to BR-Length
*
* The set of fields in the following lines are defined in sample copybook
* ARSZDCWS. For re-entrant code, they must be populated dynamically. Also,
* if you want to override any of the default values, you may add code to do so.
*
    Move overridecodepage                             to CS-CodePage
    Move overridetracelevel                         to CS-APITraceLevel
    Move input-NumSecs-to-Wait-value            to CS-NumSecs-to-Wait
    Move input-SocketsRequested-value           to CS-SocketsRequested
    Move Spaces                                              to CS-Message
    Move input-AG-value                                 to BR-AG
    Move overridelanguage                             to BR-Language
    Move overridecountry                              to BR-Country
    Move input-DiskW-value                              to BR-DiskW
    Move input-NumThread-value                    to BR-NumThread
    Move input-NumSQL-value                         to BR-NumSQL
    Move input-SQLSec-value                         to BR-SQLSec
    Move input-MaxHits-value                          to BR-MaxHits

    Call Structured-API-Program
        using
            CS-CommonStructure
            BR-BulkRetrieveStructure

    If CS-Message > spaces
        Move CS-Message to Report-error-message
        Write Report-out
    End-if
```

How to report conditions detected by the APIs and the Content Manager OnDemand server

About this task

The structured APIs return error and informational messages in the common structure, in field CS-Message defined in the sample copybook ARSZDCWS. Messages up to 160 bytes in length are returned to indicate function-related conditions found.

Messages returned to indicate data-related conditions will give specific information about the criteria field names, values, and the reason they might be invalid.

Refer to the structured API Messages chapter in *Messages and Codes* for messages returned by the structured APIs.

Chapter 59. Structured APIs sample drivers

IBM provides two sample driver programs for the structured APIs: ARSZDAPB and ARSZDAPC. ARSZDAPB is designed to execute in a z/OS batch environment. ARSZDAPC runs exclusively in a z/OS CICS environment.

The drivers serve the following purposes:

- ARSZDAPB is used during the structured API installation to verify that the systems environment is set up properly
- Both drivers provide sample code that prepares the API requests, invokes the API calls, and processes the information returned by Content Manager OnDemand.

Sample Batch Driver - ARSZDAPB

The sample COBOL batch driver program, ARSZDAPB, interfaces with the structured API and can be run to access Content Manager OnDemand data defined to the server. You can find the sample program in member ARSZDAPB of the USERSAMP library.

You can use ARSZDAPB to perform the following functions, which are described in detail in Chapter 58, “Using the structured APIs,” on page 249:

- LOGON
- FOLDER OPEN
- HIT LIST
- RETRIEVE
- ANNOTATIONS
- RELEASE
  - RELEASEA
  - RELEASEC
  - RELEASED
  - RELEASEH
  - RELEASEL
- LOGOFF
- BULK RTRIEV

ARSZDAPB performs the following tasks:

1. Reads the REQFILE
2. Reads the BULKRTV file (for bulk retrieve requests only)
3. Validates the input
4. Performs the requested structured APIs requests

The input records to ARSZDAPB contain REQUEST records identifying the Structured API request to be performed, followed by the input records supplying values to build the request. When ARSZDAPB reads the REQUEST record, the request being built performs as if no input errors were detected. REQFILE records beginning with an asterisk (*) are treated as comments and are printed in the REPORT file.

For bulk retrieve requests, the BULKRTV file is read to obtain values for the BR-FListName, BR-ThreadDir, and BR-SQL parameters. Each of these parameters can be longer than the existing 80–byte record length of the REQUEST file.

The called program, ARSZSTOP, is used to validate a requested Operation (OP) code contained in the Permitted Operation codes for the criteria. The Operation code and the Permitted Operations are unsigned small integer fields in C-language used as binary bit flags. The program, ARSZSTOP, validates the bit flag value for the requested Operation as a valid bit flag in the Permitted Operations.

The called program, ARSPTR2X, converts a COBOL Pointer variable to display in HEX format.

File description

ARSZDAPB accesses the following files:

File	Description
REQFILE	Contains 80-byte, fixed-length records that specify the batch driver input requests
BULKRTV	Contains 32,752–byte, variable length records that specify the batch driver input requests for certain bulk retrieve parameters.
REPORT	Contains 133-byte, fixed-length records that report informational and error messages regarding the batch driver execution
SAPIOUT	Contains 133-byte, fixed-length records that log the REQFILE input records. Information is written to this file by the COBOL DISPLAY command.

The next three sections contain a more detailed description of each of the files.

REQFILE

Input records to the REQFILE are 80-byte, fixed-length. The bytes 1-19 contain the label field and bytes 20-80 contain the value field.

Values that are entered in the label field are left justified and are validated for specific values. When the value in the label field (columns 1-19) is REQUEST, the corresponding value field (columns 20–80) must be one of the following character strings:

- LOGON
- FOLDER OPEN
- HIT LIST
- RETRIEVE
- ANNOTATIONS
- RELEASE
- LOGOFF
- BULK RTRIEV

REQFILE records are grouped by REQUEST. The REQUEST record is entered first, followed by a set of records that provide the necessary information that pertains to the specific API function being requested. For example, when the REQUEST is LOGON, then all the records between REQUEST...LOGON and REQUEST...FOLDER OPEN are one group. The label and value portions of the records that follow the REQUEST record are validated based on the function being requested. Order does not matter within each REQUEST group.

The following table shows the relationship between the REQFILE records and their associated API request fields. The REQFILE does not provide a value for every field passed by the API; many API fields are populated with values returned to the batch driver by Content Manager OnDemand by a previous function request. See Chapter 58, “Using the structured APIs,” on page 249 for details regarding the API call parameters.

Table 39. Shows the relationship between the structured API request to the REQFILE field label

REQFILE function request	REQFILE field label	Structured API request
LOGON	MIDSERVER	CS-MidTierIPAddr
	MIDPORT	CS-MidTierPort
	LIBSERVER	CS-LibServIPAddr
	LIBPORT	CS-LibServPort
	FOLDERNAME	CS-FolderName
	TRACELEVEL	CS-APITraceLevel
	LOGONID	LO-UserID
	PASSWORD	LO-Password
	CODEPAGE	CS-CodePage
	FLISTMAX	CS-FolderListMax
	HITLISTMAX	SC-HitListMax[6]
	SKIPNOTESCOUNT	CS-SkipNotesCountValue[7] • NO=Return notes count • YES=Do not return notes count
	WAITTIME[4]	CS-NumSecs-to-Wait
FOLDER OPEN	FOLDERNAME	CS-FolderName
HIT LIST	HITLISTMAX	CS-HitListMax
	CRITERIA#[1]	SC-CriteriaName
	CRITERIA[1]	SC-CriteriaName
		SC-ValueCount
		SC-SearchValue1
		SC-SearchValue2
		SC-SortOption
	SCOP	SC-OP
		SC-ValueCount[2]
	SCVALUE1	SC-SearchValue1
	SCVALUE2	SC-SearchValue2
	SCSORTOPT	SC-SortOption • A=Ascending • D=Descending
	SKIPNOTESCOUNT	CS-SkipNotesCountValue[7] • NO=Return notes count • YES=Do not return notes count
	WAITTIME[4]	CS-NumSecs-to-Wait

Table 39. Shows the relationship between the structured API request to the REQFILE field label (continued)

REQFILE function request	REQFILE field label	Structured API request
RETRIEVE	RSLOGOFLA	RS-LogoffFlag • N=No implicit LOGOFF • Y=implicit LOGOFF
	RSDOCFLAG	RS-DocumentFlag • D=Retrieve Document
	HIT#[3]	RS-DocIDLen
		RS-DocID
	WAITTIME[4]	CS-NumSecs-to-Wait
ANNOTATIONS	HIT#[3]	RS-DocIDLen
		RS-DocIDLen
	WAITTIME[4]	CS-NumSecs-to-Wait
RELEASE[5]	none	
LOGOFF	none	
BULK RTRIEV	MIDSERVER	CS-MidTierIPAddr
	MIDPORT	CS-MidTierPort
	LIBSERVER	CS-LibServIPAddr
	LIBPORT	CS-LibServPort
	CODEPAGE	CS-CodePage
	FOLDERNAME	CS-FolderName
	TRACELEVEL	CS-APITraceLevel
	SOCKETSREQUESTED	CS-SocketsRequested
	WAITTIME	CS-NumSecs-to-Wait
	LOGONID	BR-UserID
	PASSWORD	BR-Password
	AGNAME	BR-AG
	LANGUAGE	BR-Language
	COUNTRY	BR-Country
	HITLISTMAX	BR-MaxHits
	DISKW	BR-DiskW
	DEBUGLEVEL	BR-Debug
	SQLSEC	BR-SQLSec
	NUMTHREAD	BR-NumThread
	NUMSQL	BR-NumSQL

Table 39. Shows the relationship between the structured API request to the REQFILE field label (continued)

REQFILE function request	REQFILE field label	Structured API request

Note:

1. Folder Criteria Number (CRITERIA#) or Folder Criteria Name (CRITERIA). Identify the Folder Name entry in the table of criteria values returned by the FOLDER OPEN request. CRITERIA# identifies the entry by index value. CRITERIA identifies the entry by Folder Field Name.
2. This value is based on the SC-OP setting. SC-OPs of L (Like) or B (Between) must have an SC-ValueCount = 2, and values in SC-SearchValue1 and SC-SearchValue2. The remaining SC-OPs must have an SC-ValueCount = 1, and a value in SC-SearchValue1.
3. Hit list Number. This structured API value must represent the index into the hitlist table returned by the HIT LIST request and is used to populate the structured API fields.
4. Number of seconds to wait for server to complete request. If this value is never specified, default wait times are issued for each request. If a non-zero value is specified, it is used for all subsequent requests until a new value is specified.
5. RELEASE applies to RELEASEA, RELEASEC, RELEASED, RELEASEH AND RELEASEH.
6. The HLISTMAX control card is supported for LOGON.
   - If the value for CS-HitListMax is zero, or no HITLISTMAX value is specified for LOGON, the program logic is unchanged.
   - If the value for CS-HitListMax is NOT zero, or a HITLISTMAX value is specified for LOGON, the HitListMax value will be the default HitListMax value for any HIT LIST request that does not specify a HITLISTMAX value. If a HITLISTMAX value is specified, it is used only for the current HIT LIST request.
7. The SKIPNOTESCOUNT control card might be used to signal the MidServer to return the notes count per hit (NO), or skip the notes count per hit (YES).

   The batch driver treats the value specified in the field CS-SkipNotesCountValue or the specification in the SKIPNOTESCOUNT control card used at LOGON as the default value for any HIT LIST request that does not include a SKIPNOTESCOUNT control card.

   If a SKIPNOTESCOUNT control card is specified for a HIT LIST request, it applies only to that HIT LIST request

REQFILE example

In Figure 26 on page 288, a user who uses userid1 with password passw1 accesses a Content Manager OnDemand library server at TCP/IP port number 1111 on the library server that has a TCP/IP address of 9.999.99.999 through the Content Manager OnDemand MidServer that has a TCP/IP address of 8.888.888.88 with port 3333. The user enters the file in h l. to query folder CHKSTMNT for report document with an account number (CRITERIA1) that starts with 123 posting date (CRITERIA4) between 2003-01-01 and 2003-08-06.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
REQUEST            LOGON
MIDSERVER          8.888.888.88
MIDPORT            3333
LIBSERVER          9.999.99.999
LIBPORT            1111
FOLDERNAME         CHK%
TRACELEVEL         1
LOGONID            userid1
PASSWORD           passw1
REQUEST            FOLDER OPEN
FOLDERNAME         CHKSTMNT
REQUEST            HIT LIST
CRITERIA#          1
SCOP               L
SCVALUE1           123
SCSORTOPT          A
CRITERIA#          4
SCOP               B
SCVALUE1           2003-01-01
SCVALUE2           2003-08-06
SCSORTOPT          A
REQUEST            RETRIEVE
RSLOGOFLAG         N
RSDOCFLAG          D
HIT#               2
REQUEST            RELEASE
REQUEST            LOGOFF
```

Figure 26. An example of a REQFILE with REQUEST records

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
REQUEST                BULK RTRIEV
MIDSERVER         8.888.888.8
MIDPORT           3333
LIBSERVER         9.999.999.9
LIBPORT           4444
CODEPAGE          500
FOLDERNAME        CHKSTMNT FOLDER
TRACELEVEL        1
SOCKETSREQUESTED  2400
WAITTIME          300
LOGONID           userid1
PASSWORD          passw1
AGNAME            CHKSTMNT
LANGUAGE          en
COUNTRY           US
HITLISTMAX        1000
DISKW             1
DEBUGLEVEL        7
SQLSEC            10
NUMTHREAD         2
NUMSQL            3
```

Figure 27. An example of a REQFILE for the BULK RTRIEV function

BULKRTV file

The BULKRTV file is only used by the BULK RTRIEV function. It holds parameters that can be longer than the limit of the REQUEST file.

For the three parameters specified in Table 40 on page 288, each can be specified up to 10 times. The number of times each is specified is set by the BR-NumSQL parameter in the REQUEST file. Each parameter must be specified the same number of times as the value of BR-NumSQL.

Input records to the BULKRTV file are 32,752 bytes, variable length. Bytes 1 through 9 contain the label field, and bytes 10 through 32,752 contain the value field. Values entered in the label field are left justified, and are validated for specific values.

Table 40. The relationship between the structured API request and the BULKRTV field label

BULKRTV field label	**Structured API request**
FLISTNAME	BR-FListName

Table 40. The relationship between the structured API request and the BULKRTV field label (continued)	
BULKRTV field label	**Structured API request**
THREADDIR	BR-ThreadDir
SQLCLAUSE	BR-SQL

Figure 28 on page 289 shows an example of a BULKRTV file. The first two queries run in one thread because they both use the same THREADDIR value. The third query runs in a second thread.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----9--------0
FLISTNAME /u/ussuser/bulkrtv/flist01
THREADDIR /u/ussuser/bulkrtv/dir01
SQLCLAUSE WHERE ACCOUNT_NUMBER IN ('123000022', '123000055', '123000261') AND POSTING_DATE = 12854
FLISTNAME /u/ussuser/bulkrtv/flist02
THREADDIR /u/ussuser/bulkrtv/dir01
SQLCLAUSE WHERE ACCOUNT_NUMBER IN ('123000287', '123000469') AND POSTING_DATE = 12854
FLISTNAME /u/ussuser/bulkrtv/flist03
THREADDIR /u/ussuser/bulkrtv/dir02
SQLCLAUSE WHERE ACCOUNT_NUMBER IN ('123000535', '123000709') AND POSTING_DATE = 12854
```

Figure 28. Example of a BULKRTV RTRIEV function

REPORT file

The REPORT output file lists each input record and indicates whether it passed validation. Rejected input prints on the report with one asterisk (*). messages from the ARSZDAPB program print is prefixed by two asterisks (**), and messages returned from the API call print on the report prefixed by three asterisks (***).

```
 ARSZDAPB Report of REQFILE input

*  = Edit error of input detected by ARSZDAPB
** = Message issued by program ARSZDAPB
***= Message returned by the API
   REQUEST            LOGON
   MIDSERVER          hostname.address
   MIDPORT            3333
   LIBSERVER          9.99.999.99
   LIBPORT            1111
   FOLDERNAME         CHK%
   TRACELEVEL         1
   LOGONID            adminid
   PASSWORD           admpass

***Folder List for 1069624575357-9 successful- user has access to <4> folders.
          Maximum requested = <50> folders returned = 4
   REQUEST            FOLDER OPEN
   FOLDERNAME         CHKSTMNT
   FOLDERNAME         CHKSTMNT

***Open Folder for 1069624575357-9 successful
   REQUEST            HIT LIST
   CRITERIA#          1
   SCOP               L
   SCVALUE1           123
   SCSORTOPT          A
   CRITERIA#          4
   SCOP               B
   SCVALUE1           2003-01-01
   SCVALUE2           2003-08-06
   SCSORTOPT          A

***Hit List for 1069624575357-9 successful - user has access to <29> hits.
          Maximum requested=<50> hits returned = 29
   REQUEST            RETRIEVE
   RSLOGOFLAG         N
   RSLOGOFLAG         N
   RSDOCFLAG          D
   HIT#               2

***Retrieve for 1069624575357-9 successful
   REQUEST            RELEASE
   REQUEST            LOGOFF

***Logoff 1069624575357-9 successful

 ** ARSZDAPB Normal end of job
```

SAPIOUT file

The SAPIOUT output file displays the values in the interface structures before and after the API calls is issued.

The following shows the common structure and folder list values returned from the LOGON request using the REQFILE information from Figure 26 on page 288.

```
Display of CommonStructure returned fields
         RC = 0000000000
   connectID = {1069624575357-9.....}
   connectID upper= {FFFFFFFFFFFFF6F00000}
                    {10696245753570900000}
   pFolderList = 06EAA1A0
    pFolderCrieria = 00000000
       pHitList = 00000000
       pDocument = 00000000
         fn = 0000000003
 short-sock = 0000000000
     socket = 0000000000
   arraynum = 0000000000
    payload = 0000000000
  numstruct = 0000000000

     Display of FolderListStructure
    eyeball = {ARSZSCFL}
   Display of 4 entries of the FolderListStructure
    Folder Name(1) = CHK390lo
    Folder Desc(1) = FLDR-CHECKING ACCOUNT STATEMT-390 Large Object

    Folder Name(2) = CHK390so
    Folder Desc(2) = FLDR-CHECKING ACCOUNT STATEMT-390 Indexer Small Object

    Folder Name(3) = CHKACIF FOLDER
    Folder Desc(3) = FLDR-CHECKING ACCOUNT STATEMT

    Folder Name(4) = CHKSTMNT
    Folder Desc(4) = FLDR-CHECKING ACCOUNT STATEMT
```

Figure 29. Example of an SAPIOUT file returned from a LOGON request

ARSZDABP execution

You must compile and link-edit the IBM-supplied sample ARSZDABP program before it can be executed. You can find the IBM-supplied JCL to compile the batch driver in member ARSZDCBJ of the USERJCL library. For more information, see Chapter 57, “Content Manager OnDemand structured API installation instructions,” on page 235.

You can find the IBM-supplied JCL to execute the batch driver in member ARSZDABJ of the USERJCL library. The following shows the JCL:

```
//JOBCARD
//* ------------------------------------------------------------------
//*
//* APAR PQ81183: INITIAL DISTRIBUTION
//*
//* SMP/E LIBRARY: SARSINST
//* LOCAL LIBRARY: USERJCL
//*        MEMBER: ARSZDABJ
//*
//* THIS JCL MAKES USE OF THE ONDEMAND "USER" DATSETS ALLOCATED VIA
//* JCL MEMBERS "ARSUALOC" AND "ARSUALOL".
//*
//*
//* THIS JCL EXECUTES THE BATCH DRIVER PROGRAM TO RUN THE IVP TEST
//*
//*
//*  CHANGE THE OPERANDS ON THE FOLLOWING SET STATEMENTS TO REFERENCE
//*  THE DATA SET NAMES APPROPRIATE FOR YOU INSTALLATION.
//*
//PROCESS   OUTPUT  DEFAULT=YES,CLASS=*,JESDS=ALL,OUTDISP=HOLD
//USERPRE   SET     USER='ARS.V10R5M0'
//SARSPRE   SET     SARS='ARS.V10R5M0'
//*
//* The 'TCPIP.DATA' file is needed by the TCPIP requests issued by
//* the Structured API's interface module 'ARSAPIIX'. The 'TCPIP.DATA'
//* file is discussed in the TCPIP Customization and Admin publication
//* or the z/OS Communications Server: IP Configuration Guide.
//*
//* Change 'TCPIP.DATA' in the SYSTCPD SET statement to match the file
//* name in your system.
//*
//* If 'TCPIP.DATA' is a PDS, change '(TCPDATA)' in the TCPDATA SET
//* statement to match the member name for the 'TCPDATA'. If
//* 'TCPIP.DATA' is a sequential file, change '(TCPDATA)' to ".
//*
//SYSTCPD    SET    SYSTCPD='TCPIP.DATA'
//TCPDATA    SET    TCPDATA='(TCPDATA)'
//*
//* ------------------------------------------------------------------
//*
//STEP1    EXEC PGM=ARSZDAPB,PARM='/ABTERMENC(ABEND)',REGION=0M
//STEPLIB  DD DISP=SHR,DSN=&USER..USERLOAD
//         DD DISP=SHR,DSN=&SARS..SARSLOAD
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=134,BLKSIZE=134)
//REQFILE  DD DISP=SHR,DSN=&USER..USERPARM(ARSZUIVP)  IVP
//BULKRTV  DD DISP=SHR,DSN=&USER..USERBRET(ARSZUIVP)  IVP
//SAPIOUT  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//REPORT   DD SYSOUT=*
```

Sample CICS® Driver - ARSZDAPC

The CICS sample COBOL driver program, ARSZDAPC, interfaces with the structured API and can be run to access Content Manager OnDemand data defined to the server. You can find the sample program in member ARSZDAPC of the USERSAMP library.

You can use ARSZDAPC to perform the functions described in detail in Chapter 58, “Using the structured APIs,” on page 249:

- LOGON
- FOLDER OPEN
- HIT LIST
- RETRIEVE
- ANNOTATIONS
- RELEASE
- LOGOFF

The following table describes the CICS screens that display as they relate to the API call (request and returned data):

Table 41. CICS screens that display as they relate to the API call

Function Request	ARSZDAPC Request Screens MAP NAME - Description	ARSZDAPC results panel MAP NAME - Description
LOGON	• ARSLOGN - LOGON • ARSXUSR - Expanded UserID & Password • ARSUCTR - Establish Terminal Case	ARSFLST - Folder List
FOLDER OPEN	ARSFLST - Folder List	ARSCRIT - Folder Criteria
HIT LIST	• ARSCRIT - Folder Criteria • ARSXCRI - Expanded Criteria	ARSHITS - Hit List
RETRIEVE	ARSHLST - Hit List	ARSDOC - Retrieved Document
ANNOTATIONS	ARSHLST -Hit List	ARSNOTE - Annotations List
LOGOFF	All	ARSLOGN - Logon

Program ARSZDAPC is designed to demonstrate programming techniques to use the Structured APIs. The program proceeds from "LOGON" through "LOGOFF" in sequence by structured APIs requests. A document is retrieved (if viewable or printable) but not displayed.

To view the retrieved document, do the following steps:

1. Use CICS CEDF and set stop points for SEND.
2. Proceed through the SAPI transaction until the Retrieve Document screen is returned.
3. Press F5 to view Working Storage and see the displayed address for the Document Structure.
4. Type the address into the storage address to be displayed and view the Document Structure.

Each screen presented to the operator requires a positive response from the operator to initiate the structured APIs request. The LOGON and FOLDER CRITERIA screens require that you press the F5 key to initiate the request. The FOLDER LIST and HIT LIST require that you use an S to select a Folder or Hit to initiate the request.

The major processes to be performed are placed in COBOL sections. Each process follows a similar format:

1. Initialization begins at entry.
2. The PREPARE through a PROCESS paragraph follows (with an inclusive SEND paragraph) until processing is complete or a terminal error is encountered.
3. The PREPARE paragraph builds the BMS symbolic MAP.
4. The SEND paragraph executes the BMS SEND and RECEIVE.
5. The PROCESS paragraph checks the AID keys used and enters text. Each AID key has its own processing requirements, but all (except F3) causes the screen input values to be checked. If any screen values are in error, the AID key function is ignored.

On any screen where a list is displayed, you can use function keys F7 and F8 to scroll backward and forward through the list if the list does not fit on a single screen. If the list is positioned to the beginning of the list, the F7 key and text does not display. If the list is positioned to the end of the list, the F8 key and text does not display.

If the list is wider than the screen display, you can use F9 and F10 keys to shift left and right across the list. If the list is at the left margin, then the F9 key and the text does not display. If the list is at the right margin, then the F10 key and the text does not display.

In addition to the scroll and shift keys, you can enter an exact row or column for the list position in the Row and Col entry fields at the top right of the panel. You might also enter a relative row/column number by preceding the number with a "+" (plus sign) or a "-" (minus sign). A "+" causes the scroll to be advanced towards the bottom of the list, and the shift to be advanced toward the right margin. A "-" causes the scroll to be positioned towards the top of the list, and the shift to be positioned toward the left margin. Both entry fields allow you to enter the row or column number with a blank following the number. The program ignores any values after the blank. This action allows you to enter the row or column number without doing an erase EOF.

If you are allowed to shift list left and right, a value is displayed below the Col entry, showing the shifting that is possible. These displayed values include:

>< (greater than lesser than)
The complete list entry is displayed on the screen.

>> (greater than greater than)
The list entry is at the left margin.

<> (lesser than greater than)
The list entry is between the left and right margins.

<< (lesser than less than)
The list entry is at the right margin.

If you are allowed to perform scroll operations, values are placed to the left of the beginning row number. These displayed values include:

+- (plus minus)
The complete list is displayed on the panel.

++ (plus plus)
The top of the list.

-+ (minus plus)
The list is between the top and the bottom.

-- (minus minus)
The bottom of the list.

Routines and programs used by ARSZDAPC

The external program ARSZSTOP is used to validate a requested Operation (OP) code contained in the Permitted Operation codes for the criteria. The Operation code and the Permitted Operations are unsigned small integer fields in the C programming language that are used as binary bit flags. The program ARSZSTOP validates the bit flag value for the requested Operation as a valid bit flag in the Permitted Operations.

The external program ARSZSXOP is used to convert the Permitted Operations to an array of Operation codes. The Permitted Operations and Operation codes are unsigned small integer fields in the C programming language used as binary bit flags. The program examines the bit flags in the Permitted Operations and prepares a single bit value Operation code field to match each permitted operator. The number of Operation code fields created is returned in ReturnCode.

The internal program ARSPTR2X converts a COBOL Pointer variable to display in HEX format (eight characters in length). Also, the internal program ARSADR2X converts a COBOL data item address to display in HEX format (eight characters in length).

The internal program NumberCheckAndJustify is used to validate an input field for numeric values and returns the numeric value justified.

- Justify right returns the significant number with leading zeroes converted to leading spaces.
- Justify left returns the significant number justified left with trailing spaces.
- Justify none returns the number with leading zeroes.
- Justify signed returns the number with leading zeroes. If the input number has a leading sign (+ or -), the number will be returned with as a signed number.

The routine allows the input number to be two numeric values separated by spaces, in that case, the leftmost number returns and the rightmost number are ignored. By changing the justification type used for the different number fields on the screens, you can change the justification of the numbers on the panel.

Although the sample CICS driver ARSZDAPC is written as a conversational program, the structured APIs support pseudo-conversational tasks.

The sample program ARSZDAPC is written to execute calls to the structured APIs in a specific sequence:

1. LOGON
2. FOLDER OPEN
3. HIT LIST
4. RETRIEVE
5. ANNOTATIONS
6. LOGOFF

Multiple requests for retrieve would require multiple executions of the SAPI CICS transactions: one retrieve per SAPI execution.

The following sections contain the CICS Drive sample CICS screens. Some of the screen layouts have been compressed (blank lines removed) for inclusion in this document. All of the screens can display an informational message at the bottom of the screen (above the function keys). These message might overlay some screen text. If you receive an information message when a new screen displays, press any key to clear the message and display the full screen.

LOGON

The CICS transaction SAPI invokes ARSZDAPC, which displays the LOGON Panel. The information specified on the LOGON panel is used to build the structured API LOGON request (see Figure 30 on page 295).

```
                    Structured API Sample CICS Program
                            LOGON Panel

           UserID: ________
         Password:
      Folder Name: ____________________________________________________________
       List Limit:    50
   Hit List Limit:    50
   Library Server: 9.99.999.99
             Port: 1111
       API Server: hostname.address
             Port: 3333
        Code Page: 00500

<Enter> Validate  <F3> End   <F4> Case   <F5> Logon    <F6> Expand
```

Figure 30. Example of the LOGON panel

You can perform one of the five functions on the LOGON Panel:

Function	Function	Description
Enter	Validate	Validates the LOGON Panel input fields and identifies input errors.
F3	End	Ends the SAPI transaction.
F4	Case	Invokes the **Establish Terminal Case** panel
F5	Logon	Logs onto the system.

Function	Function	Description
F6	Expand	Expands the user ID and password. If the user ID and password values are longer than what the data entry fields on the LOGON panel allows, then press F6 from the LOGON panel to display the **Expanded UserID and Password** screen (see Figure 31 on page 296).

```
                      Structured API Sample CICS Program
                        Expanded UserID and Password

   The UserID and Password may be a maximum of 128 characters with some Servers
     This screen allows you to enter the maximum length UserID and Password


    UserID: ____________________________________________________________________
____________________________________________________________


Password:


               <Enter> Validate  <F3> End    <F12> Return
```

Figure 31. Example of the Expanded UserID and Password screen

Function	Description
Enter	Validates the Expanded Userid Panel input fields and identifies input errors.
F3	Ends the SAPI transaction.
F12	Returns to the LOGON Panel. You must provide a user ID and password before pressing F12.

```
                      Structured API Sample CICS Program
                          Establish Terminal Case

  The OnDemand Server is sensitive to text case for values used for folder
  selection. Use the <F> keys below to set text case conversion.

 <F4> Set terminal to convert transaction code only to upper case
 <F5> Set terminal to convert all text to upper case
 <F6> Set terminal to not perform any text upper case conversion

             <F3> Exit                   <F12> Return
```

Figure 32. Example of the Establish Terminal Case screen

You can perform one of the five functions on the Establish Terminal Case screen:

Function	Description
F3	Ends the SAPI transaction.
F4	Converts transaction code only to upper case.
F5	Converts all text to upper case.
F6	Does not perform any text upper case conversion.
F12	Returns to LOGON Panel.

FOLDER OPEN

```
                     Structured API Sample CICS Program

                             Folder List Panel               Row     1 of    50
 A  Folder Name                                              Col     1 of   120
 C  Folder Description                                                   >>
 _  390AFPSO
    AFP INSURANCE STATEMENTS
 _  CHK390lo
    FLDR-CHECKING ACCOUNT STATEMT-390 Large Object
 _  CHK390so
    FLDR-CHECKING ACCOUNT STATEMT-390 Indexer Small Object
 _  CHKACIF FOLDERFLDR-CHECKING ACCOUNT STATEMT
 _  CHKSTMNT
    FLDR-CHECKING ACCOUNT STATEMT

Folder List for 1070329324323-9 successful- user has access to <51> folders. Max
imum requested = <50> folders returned = 50                              RC: 00000

 <Enter> Process  <F3> LOGOFF           <F8> Down             <F10> Right
```

Figure 33. Example of a Folder List Panel

The following message indicates that the logon was successful:

```
Folder List for 1070329324323-9 successful- user has access to <51> folders.  Maximum requested = <50> folders returned
= 50
```

Figure 34. Logon successful message

The value 1070329324323-9 is the connection ID associated with this user logon. Further, it indicates that 50 folders were returned for the user. Press any key to remove the message.

You can perform one of the following functions on the Folder List panel:

Function	Display	Description
Enter	Validate	Validates the values entered on the panel
F3	Logoff	LOGOFF, return to LOGON panel
F7	-	Scroll back (towards beginning of list). The "<F7> -" is displayed only when a scroll back is allowed.
F8	+	Scroll forward (towards the end of the list). The "<F8> +" is displayed only when a scroll forward is allowed.
F9	<-	Scroll left. "<F9> <-" is displayed only when a scroll left is allowed.
F10	->	Scroll right. "<F10> ->" is displayed only when a scroll left is allowed.
F12	Return	Return to LOGON Panel

The Folder List panel has an entry for each folder that the user is authorized to access. The list might be narrowed depending on whether a Folder Name (fully or partially qualified and indicated by a percent sign (%)) was provided at LOGON. Each Content Manager OnDemand folder takes up two lines of the screen. The first line holds the Folder Name. The second line holds part of the Folder Description. You select a folder for further processing by typing an S beside the Folder Name and pressing ENTER (see Figure 35 on page 298).

```
                     Structured API Sample CICS Program

                              Folder List Panel               Row     1 of    50
 A  Folder Name                                               Col     1 of   120
 C  Folder Description                                                        >>
 _  390AFPSO
    AFP INSURANCE STATEMENTS
 _  CHK390lo
    FLDR-CHECKING ACCOUNT STATEMT-390 Large Object
 _  CHK390so
    FLDR-CHECKING ACCOUNT STATEMT-390 Indexer Small Object
  _ CHKACIF FOLDER
    FLDR-CHECKING ACCOUNT STATEMT
 S  CHKSTMNT
    FLDR-CHECKING ACCOUNT STATEMT

 Type an "S" in the A/C column and <Enter> to OPEN one folder

 <Enter> Validate  <F3> LOGOFF            <F8> Down              <F10> Right
```

Figure 35. Example of a Folder List Panel

A successful folder open displays the Folder Criteria panel.

FOLDER CRITERIA

The Folder Criteria Panel lists the key fields and valid search operators for the folder search fields:

```
                    Structured API: Folder Criteria Panel

                           Folder Name: CHKSTMNT

A EXP Criteria Name                              + Row 1 of 4
C OP Criteria Value                              -
     ACCOUNT NUMBER
_ L  _________
     SSN / TAX-ID
_ =  _________
     CUST NAME
_ =  _________
  +  POSTING_DATE
_ B  **********



Enter an "S" in the AC column to expand criteria options
The OP column must specify a valid OPerator value
Enter spaces in the Value(s) to remove a Criteria from Query

   <Ent> Validate <F3> Logoff <F5> Query                        <F12> Return
```

Figure 36. Example of a Folder Criteria Panel

Note: The EXP/OP column indicates whether each Criteria requires the use of the Expanded Criteria Panel to make updates, and displays the current OPerator. A "+" above the OPerator indicates that the Expanded Criteria Panel is required for this Criteria. If the "+" is present, the OPerator is displayed but might not be changed. The Criteria Value might not be changed, and if it has a value other than spaces, the value is displayed as asterisks.

You can perform any of the following functions on the Folder Criteria Panel:

Function	Display	Description
Enter	Validate	Validates the values entered on the panel
F3	Logoff	LOGOFF, return to LOGON panel
F5	Query	Search for documents matching the Folder Criteria values
F7	-	Scroll back (towards beginning of list). "<F7> -" is displayed only when a scroll back is allowed.
F8	+	Scroll forward (towards bottom of list). "<F8> +" is displayed only when a scroll back is allowed.

Function	Display	Description
F12	Return	Return to LOGON Panel

Valid values for the OP column on the Folder Criteria Panel and the Permitted Operators fields on the Expanded Criteria Panel include:

=
Equal

^=
Not Equal

<
Less Than

<=
Less Than or Equal

>
Greater Than

>=
Greater Than or Equal

L
Like

^L
Not Like

B
Between

^B
Not Between

You can key in the values in the OP column from this panel, or you can select from the Permitted Operators listed on the Extended Criteria Panel.

You can enter a request to build a hit list of report documents from the Folder Criteria Panel by entering the first value and the search operator in the Folder Criteria Panel.

If the Criteria specifies the "Between" or "Not Between" OPerators, the Expanded Criteria Panel must be used to alter any values for the Criteria. Always enter the first and second search values required for the "Between" and "Not Between" operators in the Expanded Criteria Panel. You can also use the Expanded Criteria Panel to enter search argument values, select the sort option, and change the query operator. Provide these values from the Folder Criteria Panel by entering an S in the leftmost column of the detail line that displays the criteria that you want to change, and pressing ENTER. The Expanded Criteria Panel is displayed when the Operator value for a Criteria entry on the Folder Criteria Panel is changed to "Between" or "Not Between". The following figure shows after an S is typed beside the Insured Name search field and ENTER is pressed:

```
                    Structured API: Expanded Criteria Panel

                             Folder Name: CHKSTMNT
Name: POSTING_DATE
Sort: D
OP Permitted OPerators
B = B
Value 1:
1990-01-25
Value 2:
2005-01-24
<F4> - Position cursor to select from Permitted OPs or OP field for Default OP
<Ent> Validate <F3> Logoff <F4> OP <F6> Reset <F12> Return
```

Figure 37. An example of the Expanded Criteria Panel

You can perform any of the following functions on the Folder Criteria Panel:

Function	Display	Description
Enter	Validate	Validates the values entered on the panel
F3	Logoff	LOGOFF, return to the LOGON Panel.
F4	OP	When the cursor is positioned in the OP entry field, pressing F4 replaces the current OPerator with the Default OPerator. When the cursor is positioned to one of the Permitted OPerators, pressing F4 replaces the current OPerator with the selected Default OPerator.
F6	Reset Restore	Pressing F6 once resets the current values in the Expanded Criteria Panel to the values at entry. The text for the F6 key is changed to "Restore". Pressing F6 twice in succession restores the current values in the Expanded Criteria Panel with the default values at Folder Open.
F12	Return	Return to the Folder Criteria Panel. If there are errors found in the Expanded Criteria Panel, the text for the F12 key is changed to "Cancel". Pressing the F12 key twice in succession will return to the Folder Criteria Panel with the Criteria reset to their values when the Expanded Criteria Panel was entered.

To change the logical operator, do the following steps:

1. Place the cursor on a Permitted Operator value and press F4 to move that value into the OP field. This change is reflected on the Folder Criteria Panel.
2. Place the cursor on the OP field and press F4 to move the default Operator to the OP field. This change is reflected on the Folder Criteria Panel.

HIT LIST

A successful hit list request displays a Hit List Panel that contains the document list matching the search criteria. The criteria field names display as column headers on the Hit List Panel (see Figure 38 on page 300).

```
                  Structured API Sample CICS Program
                                                                +  Row 1 of 50
                           Hit List Panel                       + Col 1 of 104
A                                                                 >>
C Doc Type++Notes++POSTING_DATE++ACCOUNT NUMBER++SSN / TAX-ID++ CUST NAME ++
LINE            4   1992-12-18    123000022      176-74-6868   RICHARD M LA
LINE            0   1992-12-18    123000055      619-60-6781   NANCY C JOHN
LINE            0   1992-12-18    123000261      150-62-4182
LINE            0   1992-12-18    123000287      801-74-4313
LINE            0   1992-12-18    123000352      435-02-0453
LINE            0   1992-12-18    123000469      801-54-5350   JIMMIE L PYL
LINE            0   1992-12-18    123000535      435-28-2521
LINE            0   1992-12-18    123000709      801-64-9117   W O POWELL
LINE            0   1992-12-18    123000733      435-28-4255   MR A B WATT
LINE            0   1992-12-18    123001004      150-82-0730   DEBBIE E MOR
LINE            0   1992-12-18    123001038      801-46-1260   EMMIE E MILL
LINE            0   1992-12-18    123001053      987-32-3217   SUSAN M MILL
LINE            0   1992-12-18    123001111      435-28-9485
LINE            0   1992-12-18    123001137      694-46-0974   STEPHEN CARL
LINE            0   1992-12-18    123001178      777-58-2777   MR JAMES T S
Hit List for 1070329324323-9 successful - user has access to <165> hits. Maxim
um requested=<50> hits returned = 50 RC: 00000
<Ent> Select     <F3> Logoff          <F8> +          <F10> -> <F12> Return
```

Figure 38. An example of the Hit List Panel

You can perform any of the following functions on the Folder Criteria Panel:

Function	Display	Description
Enter	Select	Processes the value in the action column. **S** Select Document **N** Select Annotations
F3	Logoff	LOGOFF, return to LOGON Panel
F7	-	Scroll back (towards beginning of list). "<F7> -" is displayed only when a scroll back is allowed.
F8	+	Scroll forward (towards beginning of list). "<F7> -" is displayed only when a scroll back is allowed.
F9	<-	Scroll left (towards left margin). "<F9> <-" is displayed only when a scroll left is allowed.
F10	->	Scroll right (towards right margin). "<F10> ->" is displayed only when a scroll right is allowed.
F12	Return	Return to the Folder Criteria Panel

RETRIEVE

You can enter a request to retrieve a report document and its associated information from the Hit List Panel by typing an S under the Action code column and then pressing ENTER. A successful retrieve displays a Retrieved Document Panel:

```
                     Structured API Sample CICS Program
                        Retrieved Document Panel

      Folder Name: CHKSTMNT
    Document Type: LINE
 Document Address: 07627D0C
  Document Length: 11,571
    Record Format: Fixed
               CC: ASA
              TRC: Yes
    Record Length: 133

Retrieve for 1070329324323-9 successful
                                                                  RC: 00000

                     <F3> LOGOFF
```

Figure 39. An example of the Retrieve Document Panel

You can perform any of the following functions on the Retrieve Document Panel:

Function	Display	Description
F3	Logoff	LOGOFF, return to LOGON Panel
F5	Notes	Toggle to the annotations for this document. "<F5> Notes" is displayed only when there are annotations for the document.
F12	Return	Return to the Hit List Panel

Annotations

You can enter a request to retrieve annotations from the Hit List Panel by typing an N under the Action code column and pressing ENTER. A successful request displays the Notes List Panel:

```
                      Structured API: Notes List Panel
Folder: Annotafp
                                                                  + Row: 1 of 4
A                                                                 - Sort: D
C    Note#    Note Text
-    4        Note 4, Private, Copy, Lower left hand corner
-    3        Note 3, Private, No Copy, Lower right hand corner
-    2        Note 2, Public, Copy, Upper right hand corner
S    1        Note 1, Public, Copy, Upper left hand corner



Notes retrieval for 1070329324323-9 successful - notes returned = 4
                                                                    RC: 00000
<Ent> Select         <F3> Logoff       <F5> Document     <F8> + <F12> Return
```

Figure 40. An example of the Annotations List Panel

You can perform any of the following functions on the Notes List Panel:

Function	Display	Description
Enter	Select	Select one or more Notes for detailed display
F3	Logoff	LOGOFF, return to LOGON Panel
F5	Document	Toggle to the document for these annotations
F12	Return	Return to the Hit List Panel

If you select a note for detailed display, the note will be displayed in the Note Display Panel:

```
                      Structured API: Note Display Panel
Folder: IVP 1 TEST
  Note: 1
  Page: 1       Color: Yellow       Public: Yes      Copy: No     X: 15    Y: 8
UserID: ARSIVPU

 Group:

  Date: 22:52:32 PM UTC Monday January 24, 2005
                                                          Col: 1 of 44
  Text: ><
Note 1, Public, Copy, Upper left hand corner



       <F3> LOGOFF                                  <F12> Return
```

You can perform any of the following functions on the Note Display Panel:

Function	Display	Description
F3	Logoff	LOGOFF, return to LOGON Panel
F12	Return	Return to the Annotations List

LOGOFF

You can enter a request to logoff from any panel by pressing the F3 key.

Notices

This information was developed for products and services that are offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at www.ibm.com/privacy and IBM’s Online Privacy Statement at www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM

Software Products and Software-as-a-Service Privacy Statement" at www.ibm.com/software/info/product-privacy.

Index

A

B

C

D

E

F

H

I

J

L

M

N

O

P

R

S

T

U

V

W

Product Number: 5697-CM1

SC19-3363-03