

CA Workload Automation CA 7® Edition

Best Practices Guide

Version 12.0.00



This Documentation, which includes embedded help systems and electronically distributed materials (hereinafter referred to as the "Documentation"), is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Workload Automation CA 7® Edition, (CA WA CA 7 Edition), formerly CA Workload Automation SE and CA 7® Workload Automation
- CA 7® Web Client
- CA ACF2™
- CA Chorus™ Software Manager (CA CSM)
- CA Chorus™
- CA Mainframe Software Manager™ (CA MSM)
- CA Critical Path Monitor (CA CPM)
- CA Datacom/AD
- CA General Transaction Server (CA GTS)
- CA Integrated Agent Services (CA IAS)
- CA JCLCheck™ Workload Automation (CA JCLCheck)
- CA Service Desk
- CA Top Secret®
- CA Workload Automation Restart Option for z/OS Schedulers (CA WA Restart Option), formerly CA 11™ Workload Automation Restart and Tracking (CA 11)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Best Practices Guide Process

These best practices are based on customer experience reported through interviews with development, technical support, and technical services. Therefore, many of these best practices are a collaborative effort stemming from customer feedback.

To continue to build on this process, we encourage you to share common themes of product use that might benefit other users. Please [consider sharing](#) your best practices with us.

To share your best *practices*, contact us at techpubs@ca.com and preface your email subject line with "Best Practices for product name" so that we can easily identify and categorize them.

Chapter 1: Documentation Changes

- [Introduction](#) (see page 9)—Streamlined and improved.
- Your Product Installation and Configuration Best Practices > [Implement a proactive Preventive Maintenance Strategy](#) (see page 13)—Added to the guide.

Contents

Chapter 1: Documentation Changes	4
Chapter 2: Introduction	9
Chapter 3: Installation Best Practices	11
General Installation	11
CA Datacom/AD Guidelines.....	12
Implement a Proactive Preventive Maintenance Strategy	13
Chapter 4: Configuration Best Practices	15
Review the IBM Workload Manager (WLM) Service Class	15
Update JES MAS Tuning	16
Use XCF for SMF Record Notification	16
Perform File Maintenance.....	17
Reconsider JCL Files.....	18
Understand LOAD Processing.....	19
Evaluate Schedule Scan Options	20
Exclude Some Data Set SMF Feedback.....	21
Parallel Submission Processing	21
Chapter 5: Customization Best Practices	23
Review the Region Size	23
Update the Initialization File	24
Startup Procedures	24
Order for Starting Programs.....	25
Add VTAM Terminals.....	26
Add Batch Terminals	27
Add CAICCI Terminals.....	28
Add TCP/IP Terminals.....	28
Add JCL Libraries	29
Move or Reallocate the Log Files	30
Move or Reallocate Batch Terminal Files	30
Move or Reallocate the Checkpoint and Queue Files	31
Move or Reallocate the Communications Data Set	31
Move or Reallocate the Database Files.....	32

Maintain the CA7AGNT VSAM File	32
Implement or Delete Submit Data Sets.....	33
Implement Workload Balancing (WLB)	34

Chapter 6: Interfaces and Integration Points Best Practices 37

CA WA Restart Option (CA 11)	37
CA JCLCheck.....	39
Email.....	40
Batch Terminal Interface	42
CAICCI Interface	43
TCP/IP Terminal Interface	45
JFM Interface.....	46
CA Driver or Global Variables.....	48
IBM WLM.....	49
CA Service Desk	51

Chapter 7: Maintenance Best Practices 53

Reorganize the Database	53
Define and Use Logs	54

Chapter 8: Define and Maintain the Workload Best Practices 57

Jobs.....	57
Job Load Processing	58
Job Definition Defaults.....	59
Late Prompting for Jobs	59
Maintenance Jobs	60
Condition Code Testing	60
JCL Overrides for Jobs	61
Nonexecutable Jobs	63
Schedules	64
Time Calculations for Jobs in the Queue.....	64
SCHONLY Field.....	66
DAILY and ROLL Options.....	67
Schedule a Specific Day of the Month	67
Repeat Jobs	68
Third Shift Schedules.....	69
Calendars and RESOLV	69
Create Calendars.....	69
Global Schedule RESOLV	71
Job Requirements.....	72

How Input Data Sets Are Added to Jobs	73
Job Requirements	73
Job Requirements for Jobs in the Queue	74
Data Set Requirements for Jobs in the Queue	75
SIRD-11 Message.....	76
Use VRM	80
Use ARF	82
Activate ARF	83
ARF High Return Code (HRC) Test	84
Define an ARFSET to Catch Any Bad Job Completion.....	84
Define an ARFSET to Limit Submission Time	85
Define an ARFSET for an LA Condition	85
Use ARF Variables	85

Chapter 9: Security Best Practices **87**

Make RACF Changes.....	87
Use UID Security.....	88
Multiple Job Security.....	90

Chapter 10: Reporting Best Practices **91**

Use BTI or CAICCI.....	91
Use Program SASSHIS8.....	91
Use CA Earl or CA Easytrieve	92
Use APA Graphs.....	93
Use SQL Reports	93

Chapter 11: Disaster Recovery Best Practices **95**

DR Mode.....	95
Disaster Recovery for a Scheduled Outage	97
Failover and Recovery for an Unscheduled Outage	101
Quiesce Job Submission	103

Chapter 12: Troubleshooting Best Practices **105**

Negative Dependencies.....	105
Tracking	106
Locked in LOAD Status.....	107
Not Tracking	108
SASSXTRK Program.....	109
Dumps of CA7ONL.....	111



Chapter 2: Introduction

The guide introduces the CA Technologies mainframe management strategy and features, and describes the best practices for installing and configuring your product.

The intended audience of this guide is systems programmers and administrators who install, maintain, deploy, and configure your product.

Chapter 3: Installation Best Practices

General Installation

Before an installation or an upgrade, read this document, understand, and incorporate the best practices. Also, follow the steps that are outlined in the *Installation Guide* and any provided ancillary documentation. A best practice is to install the CA WA CA 7 Edition product using CA CSM. Additionally, apply all CA WA CA 7 Edition maintenance available using CA CSM during the initial installation.

If you currently have CA WA CA 7 Edition installed and are installing a new release, you are upgrading and must follow the upgrade steps in the *Installation Guide*. Even if you are putting a new image on an LPAR for testing purposes, use the upgrade steps and copies of the production files for the testing phase.

Install all provided programs even though you are not using them (for example, CA General Transaction Server (CA GTS), CA Critical Path Monitor (CA CPM), and so forth). If you later decide to use them, the installation procedures make these programs available for use.

Verify that the following libraries are APF authorized:

- CA WA CA 7 Edition load library
- CA WA CA 7 Edition JFM load library
- CA CPM load library
- CA GTS load library
- CA IAS load library
- CA Datacom/AD MUF libraries

Business Value:

Following these installation best practices ensures a successful upgrade or a new installation of the software.

Additional Considerations:

No IPL is required to install CA WA CA 7 Edition. CAIRIM initializes the required system modules.

CA Datacom/AD Guidelines

We strongly recommend that you use CA Datacom/AD as the database system instead of CA Datacom/DB. You must use CA Datacom/AD r14 or higher.

When using CA Datacom/AD r14, ensure the following PTFs are applied. These PTFs address items specific for CA WA CA 7 Edition:

- RO49754
- RO50149
- RO52117
- RO54284
- RO54667
- RO55905
- RO55447
- RO59917
- RO65436

CA WA CA 7 Edition can use a CA Datacom/AD MUF with similar CA Technologies products like CA 11. We do not recommend running CA WA CA 7 Edition in the same MUF as CAIENF or CA CSM.

We recommend that the CA Datacom/AD MUF executes on the same system as CA7ONL. This method eliminates the use of IBM Cross-Systems Communication Facility (XCF), thus improving overall performance. If you use a dormant CA7ONL copy, have the Shadow MUF and dormant copy of CA7ONL execute on the same system.

Place the MUF in a WLM service class that is the same or higher performance as the CA7ONL started task. For simplicity, put both the CA7ONL and the CA Datacom/AD MUF in the same WLM service class.

Business Value:

Using CA Datacom/AD with recommended levels and parameters provides for the best optimization. Using CA7ONL with other similar Workload Automation products ensures better performance than sharing with other products.

Implement a Proactive Preventive Maintenance Strategy

CA Technologies formerly delivered product maintenance using Service Packs. We have replaced this model with [CA Recommended Service \(CA RS\) for z/OS](#), which provides more flexibility and granular application intervals. CA RS is patterned after the IBM preventive maintenance model, Recommended Service Upgrade (RSU). With CA RS, you can install preventive maintenance for most CA Technologies z/OS-based products in a consistent way on a schedule that you select (for example, monthly, quarterly, annually).

CA Technologies periodically releases Service Updates. A Service Update is a product installation file and all PTFs preapplied up to the last CA RS level.

We recommend that you develop and implement a proactive preventive maintenance strategy whereby you regularly apply maintenance. You could follow the same schedule that you use to apply IBM maintenance, or you could implement a schedule for CA Technologies products only.

Business Value:

Keeping your products current with maintenance helps your team remain productive and minimize errors while safely protecting your systems. If you do not install preventive maintenance regularly, you risk encountering known problems for which we have published and tested fixes.

Our mainframe maintenance philosophy is predicated upon granting you the flexibility to maintain your sites and systems consistent with industry best practices and site-specific requirements. Our philosophy focuses on two maintenance types. Understanding each type can help you maintain your systems in the most efficient manner.

Note: This philosophy applies to the [CA Chorus Software Manager Enabled Products](#). For legacy products, contact CA Support for maintenance details.

Corrective Maintenance

Helps you address a specific and immediate issue. This type of maintenance is necessary after you encounter a problem. We may provide a test APAR when a new problem is uncovered, and a confirmed PTF when the problem has been resolved. Your primary goal is to return your system to the same functional state that it was before you experienced the issue. This type of maintenance is applied on an as-needed basis.

Preventive Maintenance

Lets you bring your system up to a more current level. Doing so helps you avoid problems that others have reported. This maintenance may also contain compatibility fixes for hardware and software. You may have experienced the issues that each PTF addresses. CA RS provides a way to identify all published maintenance that has been successfully integration-tested. This maintenance has been tested with other CA Technologies products, current z/OS releases, and IBM subsystems, such as CICS and DB2. CA RS levels are published monthly that include PTFs, HIPERs and PRPs (PE-resolving PTFs). Before you download, apply, and test a new CA RS level, we recommend that you accept the previous CA RS level.

You can initiate a maintenance installation activity at any time. You can then install the current CA RS level of maintenance (recommended) or an earlier level. Additionally, you can install maintenance to support a new hardware device, software upgrade, or function using the [FIXCAT](#) method.

For all maintenance, *before* you initiate any maintenance action, obtain the current SMP/E HOLDDATA.

Important! [CA Chorus™ Software Manager \(CA CSM\)](#) - formerly known as CA Mainframe Software Manager™ (CA MSM) - is an intuitive web-based tool that can automate and simplify many CA Technologies product installation and maintenance activities. We strongly recommend that you use CA CSM to maintain your CA Technologies z/OS-based products.

More Information:

To apply preventive maintenance using CA CSM or from CA Support Online on <http://ca.com/support>, see the *Installation Guide* for your product and the CA CSM online help.

Chapter 4: Configuration Best Practices

This section contains the following topics:

[Review the IBM Workload Manager \(WLM\) Service Class](#) (see page 15)

[Update JES MAS Tuning](#) (see page 16)

[Use XCF for SMF Record Notification](#) (see page 16)

[Perform File Maintenance](#) (see page 17)

[Reconsider JCL Files](#) (see page 18)

[Understand LOAD Processing](#) (see page 19)

[Evaluate Schedule Scan Options](#) (see page 20)

[Exclude Some Data Set SMF Feedback](#) (see page 21)

[Parallel Submission Processing](#) (see page 21)

Review the IBM Workload Manager (WLM) Service Class

We strongly suggest that you make the following equal to or no more than one service class below JES:

- CA7ONL
- ICOM
- CA Datacom/AD Multi-User Facility (MUF) used by CA WA CA 7 Edition
- If using, CA WA Restart Option

The timely submission of jobs to the internal reader is probably the most important role of CA WA CA 7 Edition in your environment. After the implementation of z/OS 1.7, much of the internal reader processing is done in the address space allocating the reader. An address space that is not optimally qualified in the WLM Service Policy does not always get access to the processor for timely job submission.

Business Value:

This best practice ensures better CA7ONL performance and more efficient, quicker job submission.

Update JES MAS Tuning

Consider updating the MASDEF specifications for Hold and Dormancy.

In a JESplex, the JES system running within a sysplex, each LPAR is called a member. For a JESplex, the SPOOL and JES Checkpoint data sets are shared among the members, which are called the Multi-Access Spool or MAS. Only one member of the MAS can hold the checkpoint at one time.

If a MAS member holds the checkpoint too long, accessing the JES2 checkpoint can cause problems. This situation sometimes slows the CA WA CA 7 Edition job submission process. See the *JES2 Availability and Performance Redbook*. That guide helps you determine the correct specifications based on the number of members in the MAS, and what the members are doing. The member that has the CA7ONL address space (that is, the one submitting the jobs) must have a higher Hold value than the other members. The systems with less JES2 activity can benefit by adjusting the MINDORM and MAXDORM values.

Business Value:

This best practice ensures more efficient, quicker job submission.

Use XCF for SMF Record Notification

Change your initialization so that CA7ONL and ICOM join an XCF group.

CA WA CA 7 Edition tracks the progress of jobs on the operating system using SMF for z/OS work and pseudo SMF for cross-platform work. Records are generated for the job start, step end, data set create, and job end. ICOM is the facility that picks up the feedback and communicates it to CA7ONL. Before r11.3, all ICOMs wrote SMF feedback to the communications data set.

The SMF initialization file statement has a keyword of SMFXCF= with the default of NO. As a best practice, specify YES and CA7ONL joins an XCF group. The group name defaults to the instance name or can be specified on the SMFXCF keyword.

For ICOM, the PARM, XCF, determines whether ICOM uses XCF to notify CA7ONL that the SMF records are ready to process. The default is XCF=NO. As a best practice, specify NOTIFY to indicate that ICOM joins an XCF group. To specify a group name other than the default, add the keyword XCFGRP=*grpname*.

If XCF=NOTIFY is used for ICOM, XCF is not used for the SMF feedback process, but XCF notifies CA WA CA 7 Edition when SMF data is written to the COMMDS. If the LPAR that runs this ICOM does not have much activity, this setting can make SMF feedback processing quicker.

You can run ICOMs using this feature simultaneously other ICOMs are running without this feature, communicating to the same CA WA CA 7 Edition.

Business Value:

This best practice ensures more efficient tracking of work in CA7ONL.

Perform File Maintenance

Keep the CA WA CA 7 Edition files reorganized, sized, and placed correctly.

The size and location of CA WA CA 7 Edition files have an impact on the performance of the product. CA WA CA 7 Edition files can benefit from the fiber attachment, DASD Fast Write and the caching attribute on storage devices. Do not place CA WA CA 7 Edition files on the DASD volumes with other high activity product files.

The browse data set usually has a ddname in CA7ONL of BROWSE and is a sequential file that contains messages that CA7ONL produces. The messages are buffered according to the block size before they are written to the file. You can use a larger block size to decrease the number of writes to the file, which can improve the overall performance. If you have periods of inactivity in CA7ONL, the browse messages are not always written in a timely manner because the buffer must fill before the writing. To force the writing of the block to the browse file, log in to CA7ONL. Issue the `/MSG,LT=MASTER,M=message` enough times to fill the buffer. If you had a buffer size of 800, you would have to issue this 10 times to write the block to the browse file. Because typically CA7ONL is continuously busy, this activity is not an issue.

The communications data set, which has a ddname of UCC7CMDS, is the focal point of communication between CA7ONL and ICOM. The tracking information that the system collects is sent to CA7ONL by ICOM through the communications data set. The XCF option is available to pass SMF data from system to system instead of writing to the communications data set. This option can alleviate the major source of contention. Any trailer or U7SVC commands are also fed to CA7ONL through this file. In multi-CPU environments, it is imperative for performance that this file is on a low accessed volume. This file is the only CA WA CA 7 Edition file that has explicit reserves done on it for synchronization.

Business Value:

The correct placement, use, and definition of CA WA CA 7 Edition files promote better performance of CA7ONL.

More information:

[Use XCF for SMF Record Notification](#) (see page 16)

Reconsider JCL Files

Use procedures instead of inline JCL for performance purposes.

CA WA CA 7 Edition places a copy of execution JCL for job submission in the trailer queue when a job comes into the request queue for processing. PDS access can be degraded when there are large numbers of members within the file for either or both of the following:

- If you have contention for the JCL file
- If JCL files need compressing

Using PDSMAN or any other PDS performance product can help by placing the PDS directory in storage. Also, a large JCL member can take longer to submit.

Business Value:

Your jobs enter the CA7ONL request queue faster.

Understand LOAD Processing

Schedule a batch job to issue top line LOAD commands at nonpeak processing periods and only LOAD a few jobs at a time.

The first time that CA WA CA 7 Edition submits a job or if the RELOAD option is Y on the DB.1 panel, a LOAD step is inserted and executes with the JCL. The LOAD step causes the defining to the database of the step, DD, and data set information within the JCL of the job. The execution of many LOAD steps can affect system performance and overall CA7ONL performance.

If you do not use the data set information from the database, you can turn off the adding of data set information. Use the LOADDSENS keyword on the DBASE statement in the initialization file. You can also set a job to never reload by placing an X in the RELOAD field on the DB.1 panel.

Note: XPJOBS and AGJOBS do not use LOAD processing because no data sets are created on the z/OS system.

Business Value:

This best practice produces quicker, more efficient processing of batch work.

Evaluate Schedule Scan Options

Using shorter spans means fewer jobs coming into the queues during a scan, which can improve performance.

Schedule scan runs on a timer basis, bringing jobs into the queues for a specified time frame. The SCNINCR and QDWELL values from the SCHEDULE statement in the initialization file are added to determine the span of time scanned at each wakeup.

Schedule scan also drives prompting. During a prompt cycle, messages are written to the browse data set regarding late jobs or jobs in restart status. For performance purposes, set the SCHEDULE statement REPROMPT keyword to one hour, or turn it off with REPROMPT=0.

Jobs go into RETRY status in the request queue because a dynamic allocation error occurs accessing the file that contains the JCL for the job.

When a nonzero value for the RETRY keyword on the SCHEDULE statement is coded, CA WA CA 7 Edition tries to reattach the JCL from the library.

Use a longer RETRY value for performance purposes.

The XPRETRY keyword lets XPJOBS automatically retry submission when the node or alternate node is in an offline, stopped, or failed state.

Business Value:

This best practice helps ensure more efficient job processing in CA7ONL and less work in the queues to monitor.

Exclude Some Data Set SMF Feedback

You can selectively exclude specific SMF 15 record processing for data sets that CA7ONL or any job that it submits does not use.

CA WA CA 7 Edition collects various SMF records. Job initiation and job completion records are collected to determine when a job starts and ends. Data set creation records are also collected to satisfy data set requirements and for data set triggers. The collection and processing of the SMF 15 data set creation/update records is sometimes resource-intensive. Use the SASSXU83 exit to exclude specific data sets (not pick up or process the SMF 15 data for the specified data sets). If you do not use data set triggers or data set requirements within CA WA CA 7 Edition, you can turn off collection of the SMF type 15 records during the CAIRIM initialization.

Business Value:

This best practice ensures more efficient use of SMF feedback for data sets.

More Information:

For more information about the SASSXU83 exit, see the *Systems Programming Guide*.

Parallel Submission Processing

Use PSP=YES in the OPTIONS statement of the initialization file.

Parallel Submission Processing (PSP) lets CA WA CA 7 Edition open more than one JES Internal Reader at one time to submit the work to JES. This processing is activated using the OPTION,PSP=Y specification. The default number of subtasks to use is 5, although the MAXSUBOUT parameter on the OPTIONS statement can specify from 1 to 15. This setting indicates the number of jobs that CA WA CA 7 Edition can submit at one time.

Business Value: Faster submission of jobs to JES improves the job throughput through the CA WA CA 7 Edition system.

Chapter 5: Customization Best Practices

The installation macros for CA WA CA 7 Edition cause various specifications to occur (such as file space and names, JCL library designations, number of terminals, and so on). After the installation, you can change any of the items that the install macros configure.

This section contains the following topics:

- [Review the Region Size](#) (see page 23)
- [Update the Initialization File](#) (see page 24)
- [Startup Procedures](#) (see page 24)
- [Order for Starting Programs](#) (see page 25)
- [Add VTAM Terminals](#) (see page 26)
- [Add Batch Terminals](#) (see page 27)
- [Add CAICCI Terminals](#) (see page 28)
- [Add TCP/IP Terminals](#) (see page 28)
- [Add JCL Libraries](#) (see page 29)
- [Move or Reallocate the Log Files](#) (see page 30)
- [Move or Reallocate Batch Terminal Files](#) (see page 30)
- [Move or Reallocate the Checkpoint and Queue Files](#) (see page 31)
- [Move or Reallocate the Communications Data Set](#) (see page 31)
- [Move or Reallocate the Database Files](#) (see page 32)
- [Maintain the CA7AGNT VSAM File](#) (see page 32)
- [Implement or Delete Submit Data Sets](#) (see page 33)
- [Implement Workload Balancing \(WLB\)](#) (see page 34)

Review the Region Size

Define a region size of at least 50 MB and MEMLIMIT=25G if necessary for 64-bit storage use. A best practice is leaving REGION=0M and MEMLIMIT=NOLIMIT when possible.

Earlier releases of the CA7ONL procedure contained the value REGION=6144K in the installation JCLLIB. Although this region is probably sufficient for testing, we recommend that you use a value of at least 50 MB when running in production. Depending on the options and interfaces that you use, your installation could require 120 MB or more. The most current CA7ONL procedure contains REGION=0M.

Some z/OS users have reworked the standard IBM IEALIMIT exit. These users must verify that this exit has not reduced this large region value for CA7ONL.

With Version 12.0, many areas are now allocated in 64-bit storage. These areas include the active workload queues and work files that are used in performing forecasts and sorts. Also, if you are using the agent job interface, CA Integrated Agent Services (CA IAS), 64-bit memory is used. This amount is coded as MEMLIMIT on the CA7ONL EXEC statement. For CA IAS, a minimum of 10 MB is required. Depending on the installation requirements at your site, add MEMLIMIT=25G to the EXEC statement.

Business Value:

This best practice results in a better overall performance for CA7ONL.

Update the Initialization File

Use the batch edit function to help ensure the accuracy of any initialization file changes.

When CA WA CA 7 Edition is installed, the JCLLIB contains a sample initialization file. The UCC7IN DD in the CA7ONL procedure with member name ONLINE points to this file. When the initialization file is discussed, this file is the file that is referenced. To initiate this function, execute the program UCC7 with a PARM of TYPE=EDIT:

```
//EDIT EXEC PGM=UCC7,PARM='TYPE=EDIT'
```

This execution verifies the contents of the initialization file without having to start CA7ONL for any other functions. Run this program any time that you make initialization file changes to verify accuracy before implementing the changes in a production mode. This practice is especially helpful when terminals are added to or deleted from the initialization file. Running this batch job does not require that CA7ONL is already active.

Business Value:

This best practice results in no downtime due to incorrect changes to the initialization file.

Startup Procedures

When starting CA WA CA 7 Edition, we recommend an ERST type start (unless otherwise noted).

The default start is WARM. To change the start type, edit the CA7ONL PROC replacing TYPE=WARM with TYPE=ERST. Alternately, you can use the TYPE=ERST on the start command (S CA7ONL,TYPE=ERST).

You can also change the TYPE parameter in the initialization file on the INIT statement. Remember that the TYPE parameter specified in the procedure or on the start command overrides the TYPE parameter that is specified in the initialization file.

Business Value:

Correct startup of CA7ONL assures seamless scheduling of batch work.

Additional Considerations:

Two cold-type starts, COLD and FORM, are available. Use these cold-type starts rarely, especially with CA Datacom/AD. The COLD start resets the active workload queues as empty. The FORM start resets the active workload queues and the data that is contained in the Prior Run table as empty.

When performing a cold-type start of the CA 7 Online system, a WTOR asks for a confirmation of the cold-type start. If you want to perform the cold start, reply OK to the WTOR. If a cold-type start is not wanted, reply to the WTOR with a warm start option. You can suppress the WTOR in the initialization file by coding COLDWTOR=NO on the RESIDENT statement.

More Information:

For more information about the different types of start and initialization file options, see the *System Programming Guide*.

Order for Starting Programs

We recommend the following starting order (after CA Common Services CAIRIM or CAS9 has executed):

- CA Datacom/AD MUF region
- CA WA Restart Option (if using)
- ICOM
- CA7ONL
- CPM
- JFM (if using the initialization file option CPM=JFM)

Business Value:

This best practice ensures proper tracking of work and correct processing of batch work for possible restarts.

Add VTAM Terminals

As you give new users access to CA7ONL, you sometimes need to add terminals. If you need additional terminals in CA7ONL, update the initialization file.

- If you have used our defaults, change the VTM#*nnn* (where *nnn* is the number of terminals available) on the LINE and TERM statements. After this change, perform a specified ERST type start.
- If you have not used the defaults for virtual terminal naming, you have a VTNAME=*xxx* on the UCC7VTAM statement in the initialization file. Change the LINE and TERM statement with *xxx#nnn* so that the *nnn* is changed to the number of terminals available online.
- If you use the ISPF interface and want to have more terminals available to it, change the CA7CLIST SESSAPL keyword. You can use SESSAPL(TSO9999) to enable the use of all available virtual terminals in the ISPF interface.

Changing terminal definitions requires a specified ERST type start of CA7ONL.

Business Value:

Additional personnel can require more terminal definitions.

Add Batch Terminals

Have at least one set of larger batch data sets to use for commands such as RESOLV or generic listings because these commands can produce much output.

You can have up to eight batch terminal definitions. If you currently do not have eight defined, you can add batch terminals by updating the initialization file and the CA7ONL procedure. The Batch Terminal Interface (BTI) input and output data sets are pointed to in CA7ONL. Each data set has a ddname that matches the GROUP name in the initialization file. The first seven characters of these ddnames must match the initialization file GROUP statement NAME parameter. The last (eighth) character is either an I or O indicating either an input or output data set. In the initialization file, each batch terminal data set pair is defined as a batch terminal by having its own GROUP, LINE, TERM, and STATIONS statement. BTI input files are known as BATCHIN and output files as BATCHOUT. The LINE statement for each batch terminal must have the BUFSIZE keyword match the block size (BLKSIZE) of the BATCHIN file.

The PARM on the execution of SASSBSTR (program executed for BTI) determines the number of the batch terminal (and corresponding BATCHIN and BATCHOUT data sets) that are used. This number corresponds to the order of the batch terminal definitions in the initialization file. If 0 is used in the PARM, you are using dynamic allocation of the BTI terminals and BATCHIN and BATCHOUT.

Changing terminal definitions requires a specified ERST type start of CA7ONL.

Business Value:

This change results in faster processing of batch terminal input commands and fewer BATCH TERMINAL IN USE messages.

More information:

[Batch Terminal Interface](#) (see page 42)

Add CAICCI Terminals

Many different types of access to CA7ONL use CAICCI terminals. Adding CAICCI terminals as this access grows is sometimes necessary. If you need additional CAICCI terminals, update the initialization file. Add a GROUP, LINE, TERM, and STATIONS statement for each added terminal as follows:

```
GROUP, NAME=CCIGRP2, DEVICE=CCI, LNAME=CCILN2
LINE, NAME=CCILN2, BUFSIZE=3120, TNAME=CCIT2
TERM, NAME=CCIT2, DEVICE=CCI, NLINE=60, LINLEN=137, CONS=ALTRN
*
GROUP, NAME=CCIGRP3, DEVICE=CCI, LNAME=CCILN3
LINE, NAME=CCILN3, BUFSIZE=3120, TNAME=CCIT3
TERM, NAME=CCIT3, DEVICE=CCI, NLINE=60, LINLEN=137, CONS=ALTRN
.....
STATIONS, TRMID=CCIT2, STANIDS=(CCI2)
STATIONS, TRMID=CCIT3, STANIDS=(CCI3)
```

These statements are positional. Add the GROUP, LINE, and TERM statements after the existing ones for CAICCI terminals. Add the STATIONS statements after the SECURITY statement (grouped with the other STATIONS statements).

Changing terminal definitions requires a specified ERST type start of CA7ONL.

Note: Make the total number of CAICCI terminals defined the number of terminals needed for CAICCI plus the number of terminals needed for TCP/IP. If either is heavily used, define at least 20 CAICCI terminals.

Business Value:

This best practice results in quicker processing of CAICCI terminal input commands.

Add TCP/IP Terminals

Many different types of access to CA7ONL use TCP/IP terminals. TCP/IP terminals are defined in the same way as CAICCI terminals. No distinction exists between the two in the initialization file. Follow the same directions as for adding CAICCI terminals.

Define the total number of CAICCI terminals defined as the number of terminals needed for CAICCI plus the number of terminals needed for TCP/IP. If either is heavily used, define at least 20 CAICCI terminals.

To activate the TCP/IP terminal interface, code the TCPTPORT parameter on the initialization file UCC7VTAM statement to define the TCP/IP port on which CA7ONL listens for messages that are sent to CA WA CA 7 Edition. The TCPTMOUT parameter specifies the number of seconds before the TCP/IP terminal session times out. The default is 10 seconds.

Business Value:

This best practice results in quicker processing of TCP/IP terminals.

Add JCL Libraries

For performance purposes, consider using more JCL libraries with fewer members in each library.

During installation, JCL files are defined to CA7ONL in the initialization file with JCL statements. You can add or remove the JCL statements as needed by making the initialization file changes and shutting down and restarting CA7ONL.

To add dynamically a JCL library without bouncing CA7ONL, use the following top line command:

```
/JCL,OPT=ADD,INDEX=indexc,DSN=your.jcl.library
```

The INDEX keyword defines the symbolic index that is associated with the JCL statement. A symbolic index is referred to as a JCLLIB on the DB.1 panel. A symbolic index consists of an ampersand (&) followed by up to 15 alphanumeric characters.

You can also define the JCL library as an alternate library to an existing JCL library by using the ALT=xxx keyword. The xxx defines the INDEX value from a previously defined JCL library that is searched before this one.

```
/JCL,OPT=ADD,INDEX=indexc,DSN=your.jcl.library,ALT=001
```

After you define a JCL library dynamically, an index value of 255 is assigned. To display all the JCL libraries that are defined to CA7ONL including the dynamically added libraries, issue the following top line command:

```
/DISPLAY,ST=JCLVAR
```

You also can delete or update the current dynamically allocated library using the /JCL command. Remember, this library is a dynamically added library. If you restart CA7ONL, the JCL library is only defined when you specify JCLDEFS=VSAM on the RESIDENT statement in the initialization file.

Business Value:

This best practice ensures flexibility in JCL library use.

Move or Reallocate the Log Files

Shut down CA7ONL before running the log dump jobs.

The primary log file is pointed to by the UCC7LOG DD in the CA7ONL procedure and in the initialization file on the ALOG1 statement. The secondary log file is named on the ALOG2 statement and can be pointed to by the UCC7LOGS DD statement. These two files must reside on the same volume. As a best practice, when moving these files, shut down CA7ONL and then run the log dump jobs. When the log dump jobs are complete, you can then reallocate the log files and can perform an ERST type start of CA7ONL.

Business Value:

This best practice results in less down time when reallocating files.

Move or Reallocate Batch Terminal Files

To move or reallocate batch terminal input (BATCHI#*n*) files, output (BATCHO#*n*) files, or both, shut down CA7ONL. Perform the move or reallocation. Next, restart CA7ONL with a specified ERST type start.

Business Value:

This best practice results in less down time when reallocating files.

Move or Reallocate the Checkpoint and Queue Files

Shut down to move or reallocate the checkpoint file or the SCRQ and DQTQ files when those files are not allocated to a VIO (virtual input/output) device.

The queue files are pointed to in the CA7ONL procedure with the ddnames UCC7SCRQ and UCC7DQTQ. Using VIO reduces the maintenance needs for these two files.

The checkpoint file is pointed to in the CA7ONL procedure with the ddname UCC7CKPT.

The queue files and the checkpoint file must reside on the same device type when not using VIO for SCRQ and DQTQ.

Business Value:

This best practice results in less down time when reallocating files.

Move or Reallocate the Communications Data Set

If you are moving the communications data set to a different device type or enlarging it, reallocate and reformat instead of using IEBGENER.

The communications data set is pointed to by the UCC7CMDS DD in CA7ONL and in ICOM. You must have both down to reallocate or move the file. You can move the communications data set using utility IEBGENER. As a best practice, if you are moving the communications data set to a different device type or enlarging it, reallocate and reformat it. Do not copy using IEBGENER.

To reformat the communications data set, see member CA07N700 in JCLLIB.

If you are not copying the existing communications data set, stop job submission and let all active jobs complete.

Business Value:

This best practice results in less down time when reallocating files.

More information:

[Quiesce Job Submission](#) (see page 103)

Move or Reallocate the Database Files

Back up, move, or reallocate the database files using the DBUTILITY supplied with CA Datacom/AD. We recommend you consult with CA Datacom/AD support when you have questions.

Back up the CA Datacom/AD database on a daily basis using a *hot* backup. Perform a full backup on a periodic basis such as before a planned system outage. The CAL2JCL members AL2DBHOT and AL2DBKUP provide samples for this backup process.

Moving or expanding the database again uses the DBUTLTY utility with different commands. For the best methods to move the database, see the CA Datacom documentation and contact CA Support.

Business Value:

This best practice results in less down time when reallocating files.

More information:

For more information, see the *Systems Programming Guide*.

Maintain the CA7AGNT VSAM File

Use the provided procedures to back up and restore the agent database file (ddname CA7AGNT). JCLLIB members CA07N513 and CA07N514 contain samples of the execution of these procedures. Alternately, use IDCAMS REPRO to move or reallocate this database file.

The CA7AGNT file contains agent job data from executed jobs. Remove these entries using the /DELAGENT command in CA 7. The removal of old entries prevents the file from expanding too large. The agent purges the job-related data on a periodic basis, so there is no need to keep the data in the CA7AGNT file.

If you are using the agent job interface, you can also back up and can move the CA IAS checkpoint file. Use the CA IAS CIASJCL members IASCKPBK (backup) and IASCKPRL (reload). These members use the IDCAMS REPRO utility.

If you must maintain the password, CIASJCL member IASCKPRP uses a utility. The utility extracts any passwords from the backed-up file and reloads only those entries instead of all agent and message entries. Use this process when any problems exist on the CA IAS checkpoint file.

Business Value:

This best practice results in less down time when reallocating files.

More information:

For more information, see the *Agent Cookbook*.

Implement or Delete Submit Data Sets

If you do not have JES shared spool *and* have more than one LPAR where CA7ONL submits jobs, add submit data sets.

In a JES shared spool environment, CA7ONL submits jobs to the internal reader on the LPAR it is running on and JES decides where the job executes. We do not recommend that ICOM submit in this environment for the following reasons:

- JES decides the LPAR where the job actually executes.
- Performance considerations occur with the overhead of opening and reading files between systems.

To route jobs to specific LPARs in a JES shared spool environment, use route execute statements (or /*JOBPARM SYSAFF=).

If you do not have a shared JES environment, you can use ICOM to submit jobs on LPARS other than the one running CA7ONL. CA WA CA 7 Edition uses submit data sets to let ICOM do job submission. A submit data set is a physical sequential file that contains 80-byte records and resides on DASD that is shared among systems. If you decide to use submit data sets, initialize the communications data set (ddname UCC7CMD5) with the submit data sets ddnames. The ddname is required to be UCC7SUBx (the x is a number from 1 to 7) and corresponds to a DD statement in CA7ONL and ICOM. Setting the DD PARM data for ICOM to the number on its submit data set DD statement enables ICOM to do job submission. CA7ONL knows which submit data set to write the JCL of a job to by the designation of MAINID on the DB.1 panel with a SYx value. The SYx is also defined in the initialization file on a CPU statement. When it is time to submit a job, the JCL is written to the designated submit data set. ICOM picks up the JCL from the submit data set and writes it to the internal reader on the LPAR where it is running, and JES takes over from there.

If you want to remove the submit data sets and have CA7ONL submit the jobs (instead of ICOM), change the initialization file, CA7ONL, and ICOM. In the initialization file, remove the CPU statement that has the DDNAME=xxxxxxx (where xxxxxxx is the ddname of the submit data set in CA7ONL). Remove the DD in CA7ONL and ICOM, and change the ICOM PARM DDNAME= to *NONE*. Shut down or stop CA7ONL, ICOM, or both, and restart to get the changes into effect. Only stop or start the ICOM task that referred to the removed submit data set.

Business Value:

You can submit jobs to an LPAR where the CA7ONL is not running (only ICOM) when no JES is shared between the LPARs.

Implement Workload Balancing (WLB)

Use the Workload Balancing (WLB) feature that is designed to select the best mix of jobs to submit to the host system to meet defined objectives. WLB has numerous benefits including the class barriers (CA WA CA 7 Edition class, not JES), tape drive usage (if a CPU job), and the prioritized submission of jobs.

Start by defining what you want Workload Balancing to do. Do you want it to control tape drives? Do you want it to control the number of jobs submitted by class? Do you want it to control the submission of jobs by a user-defined priority? The *System Programming Guide* contains a table that discusses the steps for implementing Workload Balancing. Read this entire topic thoroughly to determine whether Workload Balancing can help you reach your job submission objectives. After you have set the goals, you are ready to do some preliminary tasks.

Answer the Workload Balancing Questionnaire (in the *System Programming Guide*). The answers to this questionnaire you establish and define your Workload Balancing goals. If you do not use Workload Balancing for tape drives or initiators, code the corresponding macros anyway. The TAPE1, TAPE2, and INITR macros have defaults that can stop the submission of jobs. For example: You want to use only the class barrier feature of Workload Balancing. Code the TAPE1 macro with the highest permitted value for 'total available.' If you do not code the TAPE1 macro, the total available tape drives default to zero, and jobs using a tape device are not submitted.

After you have coded the macros and assembled and link edited the Workload Balancing module, you have a load module named UCC7Rxxx. The xxx is the MODNAME value from the WLBDEF macro. After adding the module as a job on the DB.1 panel (job name UCC7Rxxx), you can schedule this job the same as any job. Consider scheduling different Workload Balancing modules to be in effect at different times of the day or after certain processes complete. When the job UCC7Rxxx goes to the ready queue, the Workload Balancing module by that name is loaded and the UCC7Rxxx job flushes from the queues (like a nonexecutable job). (Obviously, UCC7R is a special job name prefix that can only be used for Workload Balancing modules.)

Once a Workload Balancing module is created, the values for that module can be viewed, modified, or both with online commands. The LWLB command lets you list the values for a specific Workload Balancing module. The XWLB command lets you modify the values of the current Workload Balancing module. A /WLB command can be issued using the Batch Terminal Interface, U7SVC, or the trailer step to modify the current Workload Balancing module from JCL or user-written programs. You can modify an individual job's Workload Balancing associated values in the following ways:

- Permanently on the DB.1 panel for number of tape drives used and CA WA CA 7 Edition job class.
- The XUPD command for a one-time change after the job is scheduled to run (currently in the queue).
- A #RES statement in the JCL of a job to change the resource values of the job.

Workload Balancing affects jobs waiting submission in the ready queue. Before you implement Workload Balancing, jobs usually remained in the ready queue only long enough to be submitted and initiated unless they were held by a virtual resource. After you turn on Workload Balancing, some jobs could stay in the ready queue longer. To determine the status of a job in the ready queue, look at the output from an LQ command. For jobs in the ready queue, a date and time in the column headed by SUB/START is the time that the job was submitted to JES. If *NONE* in this field, the job has not yet been submitted. To view the nonsubmittal reason, issue the following command:

```
LRDYP, JOB=jobname/job#, LIST=STATUS
```

Once the module is loaded and in effect, it can be too late to discover that you have not set all the options correctly. You can test your Workload Balancing modules with Workload Planning. Workload Planning is a modeling process that can test what-if scenarios. You can run Workload Planning with the default Workload Balancing module and then with the module you want to test to see how the new module affects processing.

Business Value:

This best practice helps ensure more efficient processing of batch work.

More Information:

For more information about Workload Balancing, see the *Systems Programming Guide*. For more information about Workload Planning, see the *Report Reference Guide*.

Chapter 6: Interfaces and Integration Points Best Practices

This section contains the following topics:

[CA WA Restart Option \(CA 11\)](#) (see page 37)

[CA JCLCheck](#) (see page 39)

[Email](#) (see page 40)

[Batch Terminal Interface](#) (see page 42)

[CAICCI Interface](#) (see page 43)

[TCP/IP Terminal Interface](#) (see page 45)

[JFM Interface](#) (see page 46)

[CA Driver or Global Variables](#) (see page 48)

[IBM WLM](#) (see page 49)

[CA Service Desk](#) (see page 51)

CA WA Restart Option (CA 11)

Use the CA WA CA 7 Edition interface with CA WA Restart Option (CA 11). This interface supports the ARTS command monitor, ensures the automatic RMS step insertion, and provides automatic CMT updating when jobs are restarted, forced to complete, or canceled.

With CA WA Restart Option installed, CA7ONL establishes the interfaces as jobs are ready to submit, restart, or cancel (also Force Complete). If CA WA Restart Option becomes inactive, a WTO is produced:

```
CA-7.730 - CA-11 INACTIVE, DO YOU WANT JOBS SUBMITTED? (YES/NO/NO11)
```

You have the options to turn off submit until you get CA WA Restart Option activated, continue to submit work with the CA WA Restart Option step inserted, or to submit work without the CA WA Restart Option step inserted. Usually, you want to suspend submission until you can activate CA WA Restart Option.

If CA WA Restart Option is activated after CA7ONL, you can redrive the CA WA Restart Option with a CA7ONL command of SUBMIT. The CA WA CA 7 Edition top line SUBMIT command with no keywords causes the interface to CA WA Restart Option to be initialized. If CA WA Restart Option is not active, you receive the CA-7.730 WTO again. The message requires a reply before processing continues.

With CA WA Restart Option and CA WA CA 7 Edition, a parameter, CONDCHK=YES, on the initialization file RESTART statement, indicates that CA WA CA 7 Edition controls the restart step. The relative job step number in which to restart the job is passed to CA WA Restart Option. CA WA Restart Option then evaluates the true restartable step (based on temporary data sets and other items). Also, CA WA CA 7 Edition displays the JES node where the job executed and the CA WA Restart Option subsystem name on the restart panel.

Jobs with remote nodes that use CA WA Restart Option r11 or higher require one of the following items:

- The no cost CA GTS product that is configured for CA WA Restart Option.
- The CA WA Restart Option configuration option LOCALGTS=NO when all nodes are in the same SYSPLEX.

Business Value:

Using CA WA Restart Option ensures that job restarts and reruns that are correct without any manual intervention.

CA JCLCheck

Use the CA WA CA 7 Edition interface for CA JCLCheck which supports enhanced validation of JCL in the CA7ONL editor and on the LJCK command. CA JCLCheck also supports batch validation of JCL for a stream of forecasted jobs.

In the online environment, the interface displays procedure expansions, so that error messages are reported in a meaningful context. Also, the interface supports checking for the existence of data sets that can prove useful in testing production job streams.

The interface also includes a batch utility that uses the FJOB command to forecast a series of jobs. Next, the utility uses the LJCK command to extract JCL for them so that CA JCLCheck can validate the JCL for the stream as a whole. This utility is requested from an ISPF panel that is provided with CA JCLCheck.

The CA JCLCheck Common Component is a subset of the CA JCLCheck product. CA JCLCheck Common Component is shipped free of charge with CA WA CA 7 Edition to validate JCL statement syntax and job execution within CA7ONL.

CA JCLCheck Common Component is shipped with other CA products. We recommend that you install CA JCLCheck Common Component into its own SMP zone. This installation prevents duplicate installation and duplicate libraries, and eliminates new installation if the primary CA product is upgraded.

The recommended best practice is to place the CA JCLCheck Common Component load library in the linklist and above all other CA products. If it cannot be placed in the linklist, the CA7ONL started task must have a STEPLIB DD pointing to the CA JCLCheck Common Component load library.

If you do not use the AUTOPROC option, add a SYSPROC DD to the CA7ONL started task. This DD statement points to the procedure libraries that you want CA JCLCheck to search for PROC expansion.

Business Value:

Using CA JCLCheck results in fewer JCL errors.

More Information:

For more information, see the CA JCLCheck documentation.

Email

Use the email feature for an automatic problem notification. The email feature of CA WA CA 7 Edition provides the capability to generate and send emails from the address space using standard SMTP protocol. This feature is primarily intended for an automatic problem notification, such as job abends. However, the feature can notify end-users or other staff about any status.

Business Value:

Using email provides quicker notification of job abends and better communication within your organization.

Additional Considerations:

To implement the email feature:

Add the following DD to your CA7ONL procedure.

```
//SYSTCPD DD DSN=VTAM.TCPIP.DATA,DISP=SHR
```

The SYSTCPD DD statement points to the data set containing systemwide TCP/IP definitions. If only one TCP/IP system is executing in the environment, you can omit this DD statement from the CA7ONL procedure. If you have multiple TCP/IP environments, code the SYSTCPD DD statement pointing to the definition library for the TCP/IP.

Note: If TCP/IP is not in the linklist, add the TCP/IP STEPLIB data sets to the CA7ONL STEPLIB concatenation.

The following references to AL2xxxx members are for releases starting with r11.3. Release 11.1 uses member names that start with CAL2xxxx that reside in the CAL2OPTN library.

Next, add the following statements in your initialization file with the appropriate values:

```
EMAIL,ESERVER=email.com,  
EADDRLIB=CA7.AL2EADR,  
EMAILLIB=CA7.AL2EMAIL,  
EFROM=CA7,  
EREPLY=CA7@do.not.reply,  
EPORT=25,  
ETIMEOUT=10
```

If you do not already have one, build the data set referenced by EADDRLIB. The data set is a fixed block, 80-byte PDS. Members of the EADDRLIB PDS are used to build the list of email addresses to which emails are sent. The TO=xxxxxxx keyword on the /EMAIL command specifies the member name. Also, provide a TXT keyword on the /EMAIL command to provide the message text to send. We provide four sample messages to send for job failures, abends, late jobs, and jobs ending. Member AL2EMAIL in the CAL2JCL library contains these samples.

A batch utility, AL2EMLT, is available to test the email interface in an isolated environment. The utility can be used to test different SMTP (email) servers, new templates, new email address members, or all. AL2EMLT provides information about a "dummy" job so that email templates with variables referencing job information are shown with realistic data. See the AL2EMAIL member in the CAL2OPTN library.

Verify that you have an L2SCEMAL resource that is defined to the external security exactly like you would have for other L2xxxx commands.

To generate an automatic email notification for job failures, you have an ARFSET set up to respond with a /EMAIL command. A sample ARFSET follows. The ARFSET gets invoked for a job when a return completion rc = 0004. The ARFSET then issues the /EMAIL command to send an email to the email addresses listed in the member named SUPPORT with the message found in the library AL2EMAIL member @JOBEND.

```

----- ARF CONDITION EDIT FOR SUPPORT -----
FUNCTION: LIST (ADD,DELETE,EXIT,FORMAT,LIST,REPL,SAVE,SR,SS) DEFCT:
TYPE: JC   SYS EQ *           SID EQ 0   RSTC GE 0   EM EQ * DEFID: 1
      FROM: 01011975 0001     TO: 12312074 2359

JC, SC TST: STEP EQ *           PROC EQ *           PGM EQ *
CC/ABENDS : CC EQ 0004 ___ ??? GE 000 ___ ??? GE 000 ___
           ??? GE 000 ___ ??? GE 000 ___ ??? GE 000 ___
EC, EE, IS, LB, LE, LS TST:
RO:      DATE:                TIME:                A0:      INT/ADJ:
RESPONSES:
1: AC,M=/EMAIL,TO=SUPPORT,TXT=@JOBEND,JOB=&JOBNAME
2:
3:
FINAL -- DISP : N CA-11?: N BYPGDG: N USAGE:  PROCESS:  CC:
      START :                END :
PROGRAM: AR32 MSG-INDX: 00 -- AR.3.1 -- yy.ddd / hh:mm:ss

```

Batch Terminal Interface

Use the Batch Terminal Interface (BTI) program SASSBSTR to send commands to CA7ONL in batch and receive the output from the commands in SYSPRINT format.

We highly recommend that you run any long running command (such as RESOLV) or commands that have many lines of output through BTI. Also, processing database additions or updates in batch let you retain the commands used in case you have to recreate the process.

The following is the sequence of events when executing SASSBSTR:

If you are using Dynamic BTI Management, the BTI program dynamically allocates the BATCHIN and BATCHOUT files associated with a specific batch terminal. Typically, the BTI program constructs the data set names for these files using the data set name prefix from the communication data set (UCC7CMDS DD). The suffix BATCHI#*n*/BATCHO#*n* is appended to the prefix to construct the full data set names (where *n* is the associated batch terminal number).

The constructed BATCHIN and BATCHOUT data set names must match the data set names in the CA7ONL procedure.

Note: If a specific batch terminal (1-8) is specified on the PARM and the BATCHIN and BATCHOUT DDs are in the BTI JCL, no dynamic allocation occurs.

SASSBSTR writes SYSIN to the BATCHIN data set.

SASSBSTR then verifies the status flags in the UCC7CMDS (communications data set) to see whether CA7ONL is up and whether the requested batch terminal is available.

If CA7ONL is up and the batch terminal is not in use--SASSBSTR turns on a setting in the UCC7CMDS file that tells CA7ONL it has commands to process from this batch terminal.

CA7ONL reads the settings in the UCC7CMDS and processes the commands from the appropriate BATCHIN data set.

As CA WA CA 7 Edition processes the commands from BATCHIN, it writes the output from the commands into the appropriate BATCHOUT data set.

When the commands are completed, CA7ONL turns on a setting in the UCC7CMDS file noting this completion.

The SASSBSTR program has been *waking* every few seconds to check the setting. When it sees that CA7ONL has completed the processing, it writes BATCHOUT to SYSPRINT and ends.

Never cancel a BTI job that is executing. If the program SASSBSTR is canceled, the batch terminal used may not be accessible again until you shut down and restart CA7ONL.

Business Value:

BTI input can be saved and used to recreate the database updates or changes and commands that produce much output can be processed.

CAICCI Interface

Use the CAICCI interface for short running commands that have minimal output.

The CAICCI interface uses the CA Common Communications Interface (CAICCI) to send batch commands to, and receive command output from CA7ONL. The interface can be executed from batch, from a REXX address environment, or in a program-to-program mode. Because the interface uses CAICCI, there is no requirement for shared DASD between the MVS images where CA WA CA 7 Edition executes and where the CAICCI interface environment executes.

Using the CAICCI interface heavily sometimes requires that you define up to 20 CAICCI terminals.

CAICCI terminals are defined in the initialization file. When CA7ONL starts, it recognizes the CAICCI terminal definitions. This recognition causes the CAICCI Terminal interface to initialize. You then see the following messages in the SYSLOG:

CA-7.XTM0 - SASSXTM0 initialization in progress

CA-7.XTM0 - SASSXTM0 initialization complete

CA-7.XTM0 - CCI Interface initialized.

CA-7.XTM0 - CTI Receiver is: #ENFNODE .CA-7 XTM CA71

CAL2C900I CA-7 CCI Initialization Complete

Note: The initialization file OPTIONS statement CTIMSGS=NO can suppress some of these messages.

Use the following JCL to execute the CAICCI batch interface:

```
//jobname JOB (user job parms)
//JS1 EXEC PGM=CAL2X2WB,PARM='ENFNODE,CA71'
//STEPLIB DD DISP=SHR,DSN=cai.ca7.ca12load
//SYSPRINT DD SYSOUT=*
//ERRORS DD SYSOUT=*
//SYSIN DD *
.....
(CA WA CA 7 Edition commands go here)
.....
//
```

The command input is in Batch Terminal Interface format. The PARM information provided on the execute statement is positional and optional. The first position is CAICCI NODE, where CA7ONL runs. The second position is the CA7ONL CAICCI receiver name. Both can be found in the CA7ONL startup message:

CA-7.XTM0 - CTI Receiver is: #ENFNODE .CA-7 XTM CA71

Note: Most clients run in a multiple LPAR environment, and batch jobs run on different LPARs. If so, connect each LPAR through CAICCI to the LPAR where the CA7ONL started task resides. For more information about connecting LPARS, see the CA Common Services documentation.

REXX and a Program-to-Program call can also use the CAICCI interface.

Business Value:

Using the CAICCI interface lets you process short running commands in batch quickly. You are not limited to eight simultaneous executions like the batch terminal interface (BTI).

More information:

[Add CAICCI Terminals](#) (see page 28)

TCP/IP Terminal Interface

Use the TCP/IP terminal interface for short running commands that have minimal output.

The TCP/IP terminal interface uses TCP/IP to send batch commands to, and receive command output from CA7ONL. The interface can be executed from batch, from a REXX address environment, or in a program-to-program mode. Because the interface uses TCP/IP, there is no requirement for shared DASD between the MVS images where CA WA CA 7 Edition executes and where the TCP/IP interface environment executes.

TCP/IP terminals are defined in the initialization file as CAICCI terminals.

Update the initialization file on each instance of CA7ONL that is to use the TCP/IP interface. Add the following keyword to the UCC7VTAM statement:

```
TCPTPORT=nnnnn
```

where *nnnnn* is a number from 1 to 65535. This number is the TCP/IP port number on which CA7ONL listens for incoming messages. The port number must be registered in the profile of the running TCP started task.

Also, on the UCC7VTAM statement is the TCPTMOUT parameter. This parameter specifies the number of seconds in which to receive responses. The default is 10 seconds.

When CA7ONL starts, it recognizes the TCPTPORT keyword and the CAICCI terminal definitions. This recognition causes the TCP/IP terminal interface to initialize. You next see messages similar to the following in the SYSLOG:

```
CAL2T050I TCP/IP host name is USCOMP31
```

```
CAL2T051I TCP/IP host addr is 155.222.67.89
```

```
CAL2T065I TCP/IP port number is 4071
```

Use the following JCL to execute the TCP/IP batch interface:

```
//jobname JOB (user job parms)
//JS1 EXEC PGM=CAL2X2TB,PARM='USCOMP31.CO.COM,4071,CA71'
//STEPLIB DD DISP=SHR,DSN=cai.ca7.caL2load
//SYSPRINT DD SYSOUT=*
//ERRORS DD SYSOUT=*
//SYSIN DD *
.....
(CA 7 commands go here)
.....
//
```

The command input is in Batch Terminal Interface format.

The PARM information provided on the execute statement is positional and is required. The first position is the TCP/IP address where CA7ONL runs. The second position is the TCP/IP port number where CA7ONL listens. The third position is the name of the CA7ONL instance.

Note: Most customers run in a multiple LPAR environment and batch jobs run on different LPARs. If so, connect each LPAR using TCP/IP to the LPAR where the CA7ONL started task resides.

Business Value:

Using the TCP/IP interface lets you process short running commands using TCP/IP quickly.

JFM Interface

Jobflow Monitor, JFM, monitors the current and near future workload, and provides a small historical view. You can use JFM for more than one application, such as CA Critical Path Monitor (CPM) and CA 7 Web Client. JFM lets these applications make queries while collecting the job status information from the CA 7 Online system.

We recommend separate JFM tasks for each application, but separate tasks are not required. The start-up options and parameters are unique by application. Also, with each application having its own log, JFM records deal only with that application and facilitates problem isolation.

Note: For more information about JFM, see the *Interface Reference Guide*.

Additional Considerations:

If your installation has changed from the default Language Environment (LE) CEEOPTS, consider adding the following DD statement to the JFM JCL:

```
//CEEOPTS DD DISP=SHR,DSN=cai.CA7.useroptions(AL2JFM0P)
```

Where this member contains the following statement:

```
DYNDUMP(*USERID,DYNAMIC,TDUMP),TERMTHDACT(UADUMP),
RPTOPTS(ON),RPTSTG(OFF),
HEAPCHK(OFF,1,0,0,10),TRACE(OFF)
```

This setting turns off several options. This setting also improves performance for the JFM address space.

CA CPM Application: After starting the CA 7 Online system (CA7ONL), then start CA CPM before JFM. In the CAL2OPTN member JBFLWMN, the recommended parameter settings for a CA 7 instance (CA7n) are:

```
MONITOR(INSTANCE(CA7n) TYPE(WARM) REFRESH(60) SPAN(1) HISTORY(0))
```

Use a WARM start to allow automatic recovery (CAIENF events) so that CA CPM is synchronized with JFM. The history value is set to zero, because any data needed is obtained from the recorded CAIENF events. The REFRESH and SPAN values can be changed based on the needs of an installation. Considerations must include whether long running flows span more time and how much processing is being done with flows.

In the CAL2OPTN member JBFLWIP, set the IP communications for CA CPM as follows:

```
PORT(NAME(CPMPORT) NUMBER(7173))
```

Port number 7173 is the default for the CA CPM port for JFM. You can override the default by specifying a different port number.

The CA7 Web Client Application: After starting the CA 7 Online system (CA7ONL), then start JFM before starting the CA 7 Web Client. This method lets JFM get ready for query requests from the CA 7 Web Client interface. In the CAL2OPTN member JBFLWMN, the recommended parameter settings for a CA 7 instance (CA7n) are:

```
MONITOR(INSTANCE(CA7n) TYPE(COLD) REFRESH(60) SPAN(1) HISTORY(1))
```

The CA 7 Web Client is interactive with queries for current data. For this reason, perform a cold-type start because the data needs to be only from this point in time. A history period is given such that the CA 7 Web Client queries can obtain data from the past number of hours and forecast (SPAN) for the next number of hours. This period is the moving window to encompass the rolling period of workload.

In the CAL2OPTN member JBFLWIP, set the IP communications for CA 7 Web Client as follows:

```
PORT (NAME (CPMPORT) NUMBER (NONE) )  
PORT (NAME (QRYPORT) NUMBER (nnnnn) )
```

This setting disables the CPMPORT communications, as this instance of JFM does not need to exchange data with CA CPM. The QRYPORT is a valid port number for use by JFM to listen for CA 7 Web Client requests.

Business Value:

JFM is used with CA CPM to monitor critical SLAs, with reporting and updating job status.

For the CA 7 Web Client, JFM provides a graphical interface to determine job flows, triggers, and requirements.

CA Driver or Global Variables

If you have JCL substitutions that are done consistently for jobs to process correctly, look into using either CA Driver or global variables (or perhaps both).

CA Driver is provided with CA WA CA 7 Edition and can be used to perform JCL substitutions. The global variables feature lets users perform variable substitution without requiring the use of CA Driver procedures. The feature lets users define global variables and assign values to those variables. A fixed set of reserved global variables is always available (for example, current date, job name, and more). You can add user-defined variables as needed.

If your substitution is straightforward enough to do so, we recommend that you use global variables instead of CA Driver procedures.

If calculations are part of the substitution (such as a date calculation), we recommend CA Driver.

Business Value:

This best practice automates JCL substitutions.

IBM WLM

CA7ONL can facilitate the use of WLM for balancing work on the mainframe.

CA WA CA 7 Edition provides features which can facilitate the use of IBM WLM (Workload Management System) scheduling environments to manage your batch workload. A scheduling environment is a list of WLM resource names and their required states (ON, OFF, or RESET). Through CA WA CA 7 Edition Virtual Resource Management (VRM) definitions, you can assign IBM WLM scheduling environments to individual jobs and have CA7ONL automatically add the appropriate SCHENV= keyword at job submission time. These VRM definitions use the variable (VAR) resource type and can be associated based on schedule ID (SCHID). When CA7ONL prepares to submit a job, it scans VRM variable definitions for the SCHID of the job. If an appropriate VRM variable definition is found, it is used to set the value of the SCHENV keyword to insert.

Business Value:

This best practice lets you exploit WLM to control batch work.

Additional Considerations:

CA WA CA 7 Edition supports the definition of VRM variables using the RM panels. The sole purpose of a VRM variable definition is to control SCHENV keyword insertion. SCHENV keyword insertion can be tailored to a specific run of a job by defining a VRM variable in the resource profile of the job.

A VRM WLM variable is indicated when VAR is specified as the resource type in the definition on the RM.1 panel. When VAR is specified, the FREE value is not used and is always set to N. The resource name in the definition must begin with 'SCHENV.'. This signifies that the variable is used for SCHENV insertion. The second node of the resource name designates the WLM scheduling environment that must be available for the job to run. This value must conform to IBM naming conventions for scheduling environments.

The setting of the WLMSE keyword on the OPTIONS statement in the initialization file governs SCHENV keyword insertion. IF N or NO is coded, CA7ONL does not attempt SCHENV keyword insertion. WLMSE=Y or WLMSE=YES indicates that CA7ONL is to insert the SCHENV keyword. If a value other than Y or YES is coded, CA7ONL assumes that it is a global default. The global default is inserted when CA7ONL is unable to determine a more specific value for the job.

First, CA7ONL scans for an entry with an SCHID value matching that of the job being submitted. If a match occurs, the value from the resource name is inserted on the JOB statement. If no match occurs, a scan is made for an entry with an SCHID value of 0. If found, this value is the SCHENV value used on the JOB statement. If neither is present, CA7ONL attempts to use the global value defined in the initialization file, if one is available. If no global value has been defined on the WLMSE keyword of the OPTIONS statement, the job is submitted without any SCHENV keyword insertion.

Note: All scheduling environments must be defined to the IBM WLM.

If the JCL for a job already has an SCHID keyword on the JOB statement, it is not overridden or validated. If a job is submitted with an SCHID keyword that specifies a scheduling environment that is not part of the current policy, the job terminates with a preexecution JCL error.

More Information:

For more information about resource types and examples, see the *CA WA CA 7 Edition Database Maintenance Guide*.

CA Service Desk

Use the CA Service Desk feature to open problem issues automatically. This feature is primarily intended to open issues for abnormal job completions or when selected queues reach 80% full.

To implement the CA Service Desk feature

1. Install and configure the CA Service Desk using CAISDI/els from CA Common Services. CAICCI must also be present between the system where CAISDI/els and the CA WA CA 7 Edition are installed.
2. Customize the AL2CNTL member found in the CA WA CA 7 Edition CAL2EVNT library.
3. Review and if necessary create CAISDI/els events in the CAL2EVNT library.
4. Define the product and event library to CAISDI/els system.
5. Configure the CA Service Desk to recognize the information contained in the AL2CNTL member.
6. Add the SERVICEDESK initialization statement to CA WA CA 7 Edition.
7. Add SERVDESK and SERVLOG DD statements to the CA7ONL procedure. The SERVDESK statement contains the rules for the creation and customization of CA Service Desk events.

Business Value:

Using CA Service Desk enables the recording and processing of problem issues.

More Information:

For more information, see the *Interface Reference Guide*.

Chapter 7: Maintenance Best Practices

If you used CA CSM to install the CA WA CA 7 Edition product, we recommend that you install the CA WA CA 7 Edition maintenance using CA CSM.

This section contains the following topics:

[Reorganize the Database](#) (see page 53)

[Define and Use Logs](#) (see page 54)

Reorganize the Database

Reorganize the database files as frequently as your usage requires. The CA WA CA 7 Edition database consists the CA Datacom/AD database and the agent database file (CA7AGNT DD).

Back up all files at least daily.

The reorganization needs differ based on the site. If many jobs and data sets are added, deleted, or both to the database frequently, reorganization usually occurs more often. If jobs and data sets are added, deleted, or both on a quarterly basis or more infrequent time, reorganization need not occur as frequently. You can list the data set attributes through the CA Datacom/AD DBUTLTY utility. If any one of the data areas has gone into space allocation extents, review the data set placement. Keeping the data sets allocated in one extent is more efficient when possible.

To reorganize the files, use the CA Datacom/AD DBUTLTY utility.

Business Value:

Following these guidelines provides for seamless and timely processing of your workload through CA7ONL.

More Information:

For more information about the reorganization, see the *CA WA CA 7 Edition Systems Programming Guide*. For more information about CA Datacom/AD and the DBUTLTY utility, see the CA Datacom/AD guides.

Define and Use Logs

Define logs in your initialization file with ALOG1 and ALOG2 statements. CA WA CA 7 Edition writes many different log records to the log files to track its activity. As a best practice (to help ensure no log records are lost), primary and secondary logs are defined in the initialization file with the ALOG1 and ALOG2 statements. The CA7ONL procedure has a DD for the primary log with the ddname UCC7LOG. This DD statement causes the allocation of the primary log file. You can use a DD statement for the secondary log (you can call it UCC7LOGS) to allocate this log file also. This naming helps ensure that neither of the log files is moved during any DFSMS utility operations.

The DBASE statement in the initialization file JOB= keyword names a 7-character log dump job. The job is automatically submitted when the log fills to archive that data while recording to the alternate log file. A P or an S is appended to the seven character job name to denote the primary (P) or secondary (S) file to be dumped. The default job names are CA07LOGP and CA07LOGS. These jobs execute program SASSHIS5, which writes the log records to a history file.

These log dump jobs are expected to complete typically within CA WA CA 7 Edition control. If there are problems with the log dump jobs, such as abends or being canceled, then the message CA-7.200 BOTH LOG FILES FULL is issued. Because CA7ONL creates many log records, normal processing is suspended when the WTOR is issued. Jobs are not submitted to the request queue. A user cannot log on. Jobs do not move between queues, and so forth. CA7ONL tracks the log file status in the checkpoint data set. CA7ONL determines which log file is currently being written and the successful completion of the last log dump job. This knowledge is why the successful completion of log dump jobs is important.

If you receive the CA-7.200 WTOR, the first step is to try to execute one of the log dump jobs outside of CA7ONL (because it has suspended processing). If the log dump job is successful, reply that file identifier to the WTOR (LOGP for CA07LOGP or LOGS for the CA07LOGS job). CA7ONL then starts recording on the file specified and resets the checkpoint to show the new information.

Business Value:

The capability to recall log records when needed is critical for reporting purposes and in the event product or production troubleshooting is required.

Additional Information:

The SASSHIS5 program can also remove log records from the log history file and place them on the log archives file so the log history file does not become too large. This transfer can occur either while the log records are being extracted from the log files or as a separate operation. Make the DD for the log files a DD DUMMY, and use the PARM operand on the EXEC statement.

To archive records from the history file as a separate operation, have a job (other than the CA07LOGP or CA07LOGS) set up to do so. The other job is required because the log dump jobs cannot be DEMANDED into CA7ONL. They only automatically come into the queues when a log fills or you do the /SWAP command in CA WA CA 7 Edition). The PARM operand on the EXEC statement for the SASSHIS5 program has the following format:

```
PARM='yyddd,nnnnn'
```

yyddd defines a Julian date specifying the ending date of the period to place on the log archives file. Five blanks cause no records to be archived. The value TODAY can specify the current date. This parameter has no default. The *nnnnn* defines a six-digit number, with leading zeros, representing the amount of memory available to the internal sort. If not used, it must be six blanks. The maximum available memory is the default.

When the SASSHIS5 program runs, a History Management report is written to the CNTLREPT DD. This report reflects the status of logged history or archive data which is stored as internal control data in the first record of the history file. The record provides information showing when the history data was written to the history file and when or if it was archived.

The SASSHIS6 program is used to manage the size of the history files, archive files, or both. SASSHIS6 deletes unwanted log records from the log history and log archives files while keeping the internal control data between the files synchronized. SASSHIS6 must run whenever the older log records are no longer needed for analysis. These records are permanently deleted from the system and prevent the log files from becoming too large. SASSHIS6 has a required PARM operand on the EXEC statement of PURGE= which identifies from and through dates defining the period (or range) of records to purge. You can DEMAND or schedule the execution of the SASSHIS6 program.

More Information:

For more information about the log data sets and the SASSHIS5 and SASSHIS6 processes, see the *Systems Programming Guide*.

Chapter 8: Define and Maintain the Workload Best Practices

This section contains the following topics:

[Jobs](#) (see page 57)

[Schedules](#) (see page 64)

[Calendars and RESOLV](#) (see page 69)

[Job Requirements](#) (see page 72)

[Use VRM](#) (see page 80)

[Use ARF](#) (see page 82)

Jobs

Use the SYSTEM field on the job definition panel to group your jobs for listing, forecasting, and reporting purposes. This SYSTEM name can be the application group, business unit, or other designation that you prefer. Also, using the SYSTEM field with the JOBNET field can further qualify your jobs.

Jobs are defined in CA7ONL using the job definition panels (DB.1, DB.10, and DB.11). These panels give various characteristics of the job including the following:

- Execution member name in the JCL library
- Some basic job requirements
- ARFSET to use for this job
- Acceptable return codes for the job
- Eligibility for prompting
- Some of the resources that this job uses

Before you can schedule or trigger the job, define it on this panel.

Business Value:

Following these guidelines lets you group jobs for listing and reporting purposes.

Job Load Processing

LOAD jobs when any step, DD, or data set information within a job's JCL changes.

CA WA CA 7 Edition dynamically defines some of the resources that a job uses through the LOAD process.

Note: The LOAD process does not apply to internal cross-platform jobs.

The LOAD step is inserted in a job the first time it runs under CA7ONL if RELOAD: Y is specified on the DB.1 panel or if a LOAD command is done on a job. We recommend that you load jobs when any step, DD, or data set information within a job's JCL changes. When the LOAD step executes, it builds information picked up from the operating system. The information includes the following:

- Processing characteristics of the job
- Steps within the job
- DDs referenced
- Data sets that this job uses (either as output or input)

This information is communicated from ICOM to CA7ONL and then recorded in the database. Data set requirements are automatically added in this way.

LOAD processing can be a factor in CA7ONL performance.

Business Value:

Following these guidelines helps ensure that the CA7ONL database contains up to date information about the processing characteristics of a job.

More information:

[How Input Data Sets Are Added to Jobs](#) (see page 73)

Job Definition Defaults

Change the defaults on the job definition panels to fit the needs of your installation. The product comes with certain defaults on the different job definition panels. You can change these defaults by issuing a LIST function for the following jobs:

- CPU Job - DEFAULTS (DB.1)
- XPJOB - DEFAULTX (DB.10)
- Agent Job - DEFAULTA (DB.11)

Next, update the fields for your choices.

You can optionally change the default job names from the preceding defaults. Code the job names you want to represent default values on the initialization file DBASE statement.

Business Value:

Defaults tailored to your installation improve the work flow.

Late Prompting for Jobs

Use the LATE status to determine whether processing is on time. The PROMPT field on the job definition panel defines turning prompting on or off or whether a job is eligible for a LATE status in the queues. We recommend that you use the LATE status to determine whether processing is on time.

For some jobs, the LATE status has no use. Assume that you have LATEPROMPT=LATE on the OPTIONS statement in the initialization file and PROMPT=N on the job definition panel for a job. That job never has a LATE status in the queue nor does LATE prompting occur.

Business Value:

Controlling the use of LATE prompting in your environment improves performance.

Maintenance Jobs

Set MAINT: Y for jobs that perform the file backups or copies. The DB.1 panel contains a MAINT field. This field setting defines data set relationships and this job. If some of your jobs perform file backups or copies, you do not want to treat these jobs as true *updates*.

We recommend that you set MAINT: Y for these jobs. This setting disallows the attaching of data set requirements when the job comes into the queue. No data set updates that occur from the execution of this job are picked up. A data set that a job marked MAINT: Y created cannot trigger other work in and cannot post the data set when other jobs require it.

Business Value:

This best practice ensures more efficient processing of jobs.

Condition Code Testing

Successful jobs end with a 0 return code. If a job ends with a nonzero return code, it is usually a bad completion and requires a restart. You can use the DB.1 panel to set unacceptable return codes from jobs.

```
EXECUTION: MAINID: xxxx INSERT-RMS: x COND-CODE: nnnn RO: xx  
DONT SCHEDULE -- BEFORE: yyddd hhmm AFTER: yyddd hhmm
```

COND-CODE is used with RO (relational operator) to define the job-level condition codes used to determine whether a job executes successfully. If the RO field is set to 0 (zero), no condition code test is made. For the COND-CODE field, enter 1 through 4 numeric characters from 0 to 4095.

The RO field specifies the relational operator of the condition code (COND-CODE) or whether the job's JCL uses the step level #SCC statements.

This field is not required. Both parameters test the highest condition code that a job generates.

To optionally define condition code tests at the step level, use the #SCC statements after the JOB statement and before the first execute (EXEC) statement.

Note: This test is for internal use by CA7ONL only. The test simply tells CA7ONL what action to take after the job completes. CA WA CA 7 Edition does not determine or control which steps to execute.

Business Value:

The ability to determine good job completions is critical to job management.

Additional Considerations:

The easiest way to test the COND-CODE and RO field settings is to make a sentence:

COND-CODE is RO to the job's CC

And if you can answer true to the preceding sentence, the job completed abnormally.

In other words, if a job receives CC of 4 and DB.1 panel states 0 - GT

Then the preceding sentence would read:

0 is GREATER THAN 4

Because the statement is false, the job completed typically (a good execution).

This logic is the same logic that IBM uses in condition code checking.

COND-CODE and RO are not applicable to agent jobs.

More Information:

For more information about COND-CODE, RO, or #SCC statements, see the *Database Maintenance Guide*. For more information about condition code processing in agent jobs, see the EXITCODE statement in the *CA Integrated Agent Services User Guide*.

JCL Overrides for Jobs

Use the override library for one-time JCL changes. Use alternate libraries for JCL changes that you require for more than one run.

Using CA7ONL can accomplish JCL overrides in various ways. We recommend using the override library (INDEX=254) for one time JCL changes. To make one-time changes, save the required JCL to the override library and set the USE-OVRD-LIB on the DB.1 panel to Y. This field is automatically set to Y when you save the JCL to the override library using the DB.7 (JCL Edit) panel in CA7ONL. When the job next comes into the queue, the following occur:

- The JCL is used from the override library.
- The USE-OVRD-LIB is reset to N.
- The JCL is deleted from the override library when the job completes successfully.

The special override library accommodates one-time overrides only, and the temporary member is deleted after only one execution of the job. If you want to have an override that is kept after the job completes (not use the INDEX 254 library), you can set up alternate JCL libraries. Using alternate JCL libraries enables the user to place temporary JCL in a staging-type of library for use in more than a single execution of a job.

Various # statements added to JCL can change the processing of a job or queue characteristics. You can hold the job (#HLD), set it nonexecutable (#NOX), or set it not to trigger (#NTR) and many other settings. Using the override library can be beneficial when using any of the special # statements so that the update is used only once.

You can also use #J or #X statements in the JCL for inclusion or exclusion of JCL statements or other # statements. Using the #J or #X statements for inclusion or exclusion of statements in the JCL gives the flexibility of doing the inclusion or exclusion on a date/time, time, or SCHID basis.

Business Value:

This best practice helps ensure automated JCL substitutions or changes.

Additional Considerations:

If you want a job to be nonexecutable every day that it schedules into the queue if it is between the hours of 0800 and 1400, add the following in the JCL of the job:

```
#JI,OB=1400,OA=0800,CV=CU  
#NOX  
#JEND
```

The OB and OA are verified against clock time for the beginning time and ending time to mark the job nonexecutable in the queue.

If you want the job to be nonexecutable only when it runs under SCHID 20, add the following in the JCL of the job:

```
#JI,ID=020  
#NOX  
#JEND
```

The #J statements are examined and action taken by CA7ONL at the time that a job is scheduled into the request queue. CA7ONL examines the #X statements and acts at the time a job is submitted to JES. Certain things are only set with the #J statements because these things must occur as the job is entering the queue (such as the #HLD statement). The position within the JCL of the special # statements (not including the #J or #X) is not important, but we recommend placing them right after the JOB statement. This placement makes them easier to see.

If you have JCL substitutions that are required consistently for jobs to process correctly, look into using either CA Driver or global variables (or perhaps both).

More Information:

For more information about # statements, see the *Database Maintenance Guide*.

More information:

[CA Driver or Global Variables](#) (see page 48)

Nonexecutable Jobs

Use the EXEC setting on the job definition panel to mark nonexecutable jobs. Setting the EXEC field to N on the job definition panel can make jobs nonexecutable. Nonexecutable jobs are used for many different reasons, one of which could be as a control job that triggers many others. The EXEC setting is for all runs of a job at any time.

Note: You can make a job nonexecutable only during specific times of any day it is scheduled to run or for a specific SCHID.

Business Value:

This best practice helps ensure the flexibility of your workload scheduling.

More information:

[JCL Overrides for Jobs](#) (see page 61)

Schedules

CA WA CA 7 Edition uses date and time-driven and event-driven schedules and processes on-demand work. We recommend event-driven scheduling, known as triggering. Event-driven scheduling is a far more efficient technique than date and time-oriented schedules or on-demand processing. Try to have the following balance:

- 20 percent or less of your jobs that are scheduled by date and time.
- 10 percent or less of your work on demand.

Scheduling is the heart of CA WA CA 7 Edition. The best time to schedule is always the closest to the event that causes the job to require processing. Most jobs rarely run at a specific time on a specific day because of waiting on other jobs, data sets, and so on. Usually jobs run because their data is ready for processing or data is required to be ready for processing because of file closures or updates. Use data set or job triggers and requirements to sequence the execution instead of date and time scheduling of these types of jobs.

Schedule IDs (SCHIDs) are used to denote variations in jobs' schedules, processing, or both. Requirements, networks, JCL overrides, job triggers, and so on, can all be varied by using different schedule IDs. For example, if a given job is to run on Monday through Friday but the requirements vary on Fridays. This same job could have a different schedule ID defined for its Friday run.

For flexibility, it is best not to assign a meaning to a particular SCHID.

Business Value:

Follow these guidelines for using scheduling techniques, which provide the greatest flexibility in scheduling your production workload.

Time Calculations for Jobs in the Queue

Understanding how CA WA CA 7 Edition calculates time helps in the original scheduling of jobs and requirements. CA WA CA 7 Edition has several ways of scheduling jobs. These jobs enter the request queue with several time fields either specified by the user or calculated by CA7ONL.

Business Value:

Correct scheduling and a good understanding of time speed throughput.

Additional Considerations:

The following explanations use these acronyms:

- DOTM (due-out time)
- DLTM (deadline start time)
- LDTM (lead time)
- SBTM (submit time)
- QTM (queue time)

The following are sample time calculations:

Due-Out Time calculations for DEMAND jobs:

$DOTM = \text{Current time (time of the DEMAND)} + 1 \text{ hour}$

You can specify a DOTM on the DEMAND command to change this.

Due-Out Time for schedule scanned jobs:

DOTM entered on the Schedule Job Edit (DB.2.1-E) panel

Due-Out Time for triggered jobs:

DOTM entered on the trigger panel (DB.2.x)

If QTM is entered on the trigger panel, the DOTM is calculated as follows:

$\text{Current time (time of trigger)} + QTM + \text{CLOCK-TIME (from DB.1 (CPU Job Definition) panel)} + 5 \text{ minutes}$

If a triggered job enters the queue after its DOTM, a day is added to the DOTM and DLTM. If this same job had an SBTM requirement, it would also be for the next day.

If the triggering job has a DOTM of the next day, and the triggered job comes into the queue after its DLTM, another day is added to the DOTM, DLTM, and SBTM (if used).

Deadline Start Time calculations for DEMAND jobs:

$DLTM = DOTM$

You can add a DOTM and a LEADTM keyword on the DEMAND command to change this, however, only adding a LEADTM does not change the DLTM.

Deadline Start Time calculation for schedule scanned jobs:

$DLTM = DOTM - LDTM$ from the Schedule Job Edit (DB.2.1-E) panel is what schedule scan is using to see whether it is time to scan the into the queue. The queue record dates and times (as shown in LQ type commands and what is used for LATE calculation and prompting) are calculated as follows:

$DLTM = DOTM - \text{CLOCK-TIME from the DB.1 panel} - 5 \text{ minutes}$

Deadline Start Time calculation for triggered jobs:

DLTM = DOTM - CLOCK-TIME from the DB.1 panel - 5 minutes

A job is marked LATE if it has not started by its deadline start time or completed by its due-out time. DSTM is calculated as DOTM - CLOCK-TIME (from DB.1 panel) - 5 minutes unless CLOCK-TIME on the DB.1 panel is 2359. In that case, DSTM is DOTM - LDTM - 5 minutes.

More information:

[Late Prompting for Jobs](#) (see page 59)

SCHONLY Field

Use the SCHONLY option on a CALENDAR to affect schedules. The SCHONLY field on the DB.2.8 panel can determine how a schedule day is chosen when using the DB.2.1-E panel MONTHLY RDAY field.

If you use a calendar that has SCHONLY: Y, the MONTHLY RDAY field counts available scheduling days only. If you use a calendar that has SCHONLY: N, the MONTHLY RDAY field counts days of the month. In other words, consider this schedule (from the DB.2.1-E panel):

```
__ X __ MONTHLY  JAN: X  FEB: X  MAR: X  APR: X  MAY: X  JUN: X
                   JUL: X  AUG: X  SEP: X  OCT: X  NOV: X  DEC: X
                   WEEK:                DAY-OF-WEEK:
                   RDAY: 3
```

This schedule works differently when using different calendars:

- Using a calendar with SCHONLY: N creates a schedule for the third calendar day of every month.
- Using a calendar with SCHONLY: Y creates a schedule for the third working day of every month.

Business Value:

Understanding a schedule day helps when scheduling jobs or troubleshooting issues of job processing.

DAILY and ROLL Options

Using the DAILY and ROLL options correctly enhances your scheduling processes.

When you select the DAILY option on the DB.2.1-E panel, a schedule is created for the job that matches the calendar used. The ROLL function has no bearing on a DAILY schedule because no nonscheduled days are considered.

If you want the job to run every day of the week regardless of whether it is available on this calendar, we recommend using ROLL: N and a WEEKLY schedule choosing all days of the week that apply.

Business Value:

This best practice helps ensure the correct scheduling of your workload.

Schedule a Specific Day of the Month

To schedule a day-of-week to a day-of-month requires evaluation of the day-of-month and its relationship to the day-of-week. First, determine which occurrence for the day-of-month could be for the day-of-week. Next, schedule the job monthly for that week and day-of-week, and remove all days you would not want the job to execute during that week.

Business Value:

This best practice lets you schedule a day-of-week to day-of-month.

Additional Considerations:

The following are examples of scheduling a specific day of the month.

Example: Schedule a job for Sunday only if Sunday is the 7th of the month

First, you know if Sunday is the 7th of the month, it has to be the first Sunday of the month. Then the question is: What could the first Sunday of the month be and not be the 7th? It could be the 1st, 2nd, 3rd, 4th, 5th, or 6th. You schedule that as follows:

```
__X__ MONTHLY JAN:X FEB:X MAR:X APR:X MAY:X JUN:X
      JUL:X AUG:X SEP:X OCT:X NOV:X DEC:X
WEEK:1 DAY-OF-WEEK:SUN
RDAY:/1,/2,/3,/4,/5,/6
```

The calendar for this schedule should have seven days of the week available and have the SCHONLY option of N.

Example: Schedule a job for the 15th of the month only if it is a Wednesday

```
__X__ MONTHLY JAN:X FEB:X MAR:X APR:X MAY:X JUN:X  
      JUL:X AUG:X SEP:X OCT:X NOV:X DEC:X  
      WEEK:3 DAY-OF-WEEK:WED  
      RDAY:/16,/17,/18,/19,/20,/21
```

Repeat Jobs

Use the repeat scheduling feature to run jobs repeatedly throughout the day.

You can set up a job to run continuously. Use trailer steps in the job to DEMAND itself back in during specified time frames with #J or #X statements around the SASSTRLR step. These repeat runs cannot be forecasted and required JCL changes.

We recommend using the repeat scheduling feature to run jobs repeatedly throughout the day. Jobs can be scheduled to run periodically throughout a 24 hour period on one SCHID without having to DEMAND themselves back in by using this feature. The repeat scheduling feature can be forecasted.

Business Value:

This feature lets you run repeatedly a job on one SCHID.

Additional Considerations:

On the DB.2.1-E panel, specify the interval between iterations of the job. Also specify how many times to repeat the job, when to no longer repeat the job, or both.

The repeat run only occurs when the latest iteration of the job completes successfully. The decision to repeat is made at successful completion of previous iteration. The next iteration is not added to the request queue until successful completion (or forced complete).

The DEMAND(H) command also has a repeat option for more scheduling flexibility.

Repeating jobs can trigger other jobs, but it is best that repeat jobs do not have job requirements. The reason for no job requirements is because job requirements must meet sequence testing. In other words, the required job has to run between each run of the job with the requirement.

Third Shift Schedules

Consider scheduling jobs that run around the midnight boundary on different calendars that as a rule have Tuesday through Saturday as available processing days. This setup results from day designation changes from 24:00 to 00:01 (Sunday turns to Monday, and so on).

On the DB.2.1-E panel, jobs are scheduled with a DOTM (due out time). The DOTM is for the day that the job is to complete.

- Assume that you schedule a job with a DOTM of 01:00 Monday through Friday.

The job enters the queue late Sunday through Thursday.

- Assume that you want the job to enter the queue Monday through Friday at 01:00.

Schedule that job on Tuesday through Saturday because those days are the days of the week on which the job is due to complete.

Business Value:

This best practice results in third shift work schedules correctly.

Calendars and RESOLV

Calendar scheduling is also known as date and time scheduling.

Each site requires calendars defined for the specific site. Generally, sites require calendars for Monday through Friday schedules, Saturday through Sunday schedules, nonstandard month schedules, and so on. If you do third-shift work to adjust the available processing days, consider having two calendars for each iteration.

More information:

[Third Shift Schedules](#) (see page 69)

Create Calendars

Use online or perpetual calendars. Base calendars are required for schedule resolution and can be generated through either batch or online facilities or self generated as perpetual calendars. We recommend using online or perpetual calendars. To use the online facility or perpetual calendars (DB.2.8), allocate a calendar PDS, and identify it in the initialization file with a CALENDAR statement.

The CALENDAR macro can also generate calendars in batch. Macro keywords are used for calendar definition, and the macro is assembled and link-edited using standard procedures.

Starting with r11.3, you can use perpetual calendars that are self-generating. When using perpetual calendars, you do not have to build new calendars each year manually. Perpetual calendars let you define calendars one time without regards to the year and have CA WA CA 7 Edition automatically build each new year's calendar.

Business Value:

Seamless processing with calendars is always available.

Additional Information:

To implement perpetual calendars, add PCALDSN on the CALENDAR statement in the initialization file. Next, define generic days in the PDS as a member with name PCALYYxx where the xx associates this PCAL member with a specific SCALyyxx (*yy* is the *current* year). When the calendar is next referenced (with CALMOD, RESOLV, or PRINT commands) and no SCALyyxx member exists, one is built from PCALYYxx definition. Perpetual calendars support criteria language in the PCALYYxx member to set relative days, define US holidays, and you can define unique, specific days—with any of these set to SCHDAYS or NOSCHDY.

If you are using nonperpetual calendars, the next year calendar is typically available by July 1 of the current year.

However they are created, calendars are stored in the form of bit masks that represent available and unavailable processing days for the year. Each calendar defines a single year comprising 12 months with a starting and ending year boundary that is crossed only in the first or 12th month.

Use the PRINT command to produce a month-by-month listing of a calendar. PRINT produces output that reflects the beginning and ending days of each month, holidays, or any other nonprocessing days that were defined when the calendar was created.

Resolve the schedule at the following times:

- Whenever a calendar-oriented processing schedule expires (at the end of June or the end of the year).
- The scheduling frequency is redefined for any other reason.

Global Schedule RESOLV

To help ensure seamless scheduling of the workload, we recommend that you perform a global RESOLV twice annually, first sometime between January 1 and June 30, and the second between July 1 and December 31 (after the next year's calendars are built).

Business Value:

This best practice helps ensure seamless job schedule processing.

Additional Information:

Perform global RESOLVs in batch (using BTI). The following is the syntax for the command:

```
RESOLV,OLDYR=*,TEST=NO,PRINT=YES/NO,SCAL=xx
```

The PRINT value is YES or NO. YES produces much output. Verify that the BATCHOUT data set is large enough to handle this output. If you have a BATCHOUT OVERFLOW condition when running a RESOLV command, either use keywords on the RESOLV command to cut down the number of job schedules or reallocate the BATCHOUT file larger.

Always check the RESOLV output to verify that all schedules resolved successfully.

Also, look at the RESOLV output for the following message:

SRC1-117 (W) ID MULTIPLE SCHEDULES AT LEAST ONE DAY

This message indicates two or more SCHEDIDs are scheduled to process on the same day. This message sometimes indicates a scheduling error. Determine whether the job processes multiple times on the same day. If it does, this message is an informational message. If not, add DUPDATE=YES to the RESOLV command to see what dates are duplicately scheduled so that the schedule can be corrected.

You can use the LSCHD command to determine whether you have any expired schedules or whether you have schedules expiring after June 30 or December 31. Between January 1 and June 30, you can enter the following command to see any schedules expiring July 1:

```
LSCHD,DSNBR=SJ*,ST=JUL
```

Or to see schedules that expiring December 31, enter the following:

```
LSCHD,DSNBR=SJ*,ST=JAN
```

The following is also a command that you can enter to see whether you have any currently expired schedules:

```
LSCHD,DSNBR=SJ*,ST=EXP
```

A RESOLV removes any manual modifications made using the SCHDMOD panel to schedules. Before you resolve schedules, issue the following command to see whether any have been SCHDMODEd:

```
LSCHD,DSNBR=SJ*,ST=MOD
```

More information:

[Move or Reallocate Batch Terminal Files](#) (see page 30)

Job Requirements

CA WA CA 7 Edition provides special flexibility in defining facilities that prevent jobs from being executed before all requirements are met. These requirements include input data set creation, predecessor job completion, JCL overrides, SUBMIT times, and manual tasks such as user requirements, HOLD, or VERIFY.

More information:

[Use VRM](#) (see page 80)

How Input Data Sets Are Added to Jobs

Use the DB.3.1 panel for manually adding data set requirements.

CA WA CA 7 Edition dynamically defines some of the resources that a job uses through the LOAD process. When the LOAD process runs, information is recorded in the database about data set references in the execution JCL of the job. This information is used to find out about the data sets that a job requires to run.

The job and data set definitions that are built in the LOAD process can be manually tailored as needed. These definitions are sufficient for properly sequenced job execution, but you can manually add data set requirements using the DB.3.1 panel. If you do not want a specific job ever to wait for data set requirements (or to post data set creations), set that job as MAINT: Y on the DB.1 panel.

Business Value:

Following these guidelines lets you define external data set requirements manually to meet your processing requirements.

Job Requirements

If you have a job that must wait for the completion of several different jobs, have one job trigger the execution and the other jobs set up as job requirements.

You can set up jobs to run only after the successful completion of another by using a job trigger to schedule the required job. If you have a job that must wait for the completion of several different jobs, you can have one job trigger the execution, and the other jobs set up as job requirements. In other words, if you had JOBA, JOBB, and JOBC and you want JOBC to execute after JOBA and JOBB have run, you set up JOBA to trigger JOBC. Make JOBC dependent on JOBB on the DB.3.2 panel. That way when JOBA completes successfully, JOBC comes into the queue and checks to be sure JOBB runs before it can run.

If not all runs of a job require the same other jobs to have run, set up SCHIDs to run and then connect the requirements by SCHID as needed. If JOBB does not always run when JOBA and JOBC do, connect the requirement of JOBB to JOBC only when JOBC runs under an SCHID that corresponds to when JOBB runs. In other words, if JOBA and JOBC always run on Monday through Friday and JOBB only runs on Monday, have a beginning schedule from JOBA that had a Tuesday through Friday SCHID and then another SCHID for Monday. A triggered job takes the triggering job's SCHID. When JOBA is scheduled under the Monday SCHID, JOBC triggers in with the Monday SCHID and attaches requirements for that SCHID. Define the requirement from JOBC to have JOBB run before it only under the Monday SCHID.

You can use *negative dependencies* to keep jobs from running concurrently. Define a negative job dependency by placing a / in front of the PRED-JOB on the DB.3.2 panel. This / reflects mutual exclusion of jobs and prevents them from executing at the same time. To help ensure exclusion, make a similar connection to define this job as mutually exclusive for the predecessor job. Lead time is not valid with negative job dependencies.

Negative job dependencies are checked when a job is in the ready queue preparing for submission (all other job requirements have been satisfied). If one of the job's mutually exclusive jobs has been submitted (ready queue or active queue), a positive job dependency is assigned to the job and it is moved back to the request queue. The job then waits for the successful completion of the mutually exclusive job before it is ready for submission. A job requirement is assigned even if the active job was requested with a RUN function or command. The completion of the active job from RUN does not POST the requirement; the requirement must be posted manually.

Consider using an exclusive virtual resource instead of negative dependencies in some cases. The basic difference with VRM EXC resources as opposed to negative dependencies is that you can control how the completion of a job posts (or releases) the job to run.

Business Value:

Following these guidelines helps ensure correct sequencing of jobs.

Job Requirements for Jobs in the Queue

Use job requirements to sequence correctly job executions. When jobs are scheduled into the queue for execution, the requirements for this execution of the job (by SCHID) are attached, and the master requirement count (MCNT) is incremented for each requirement. A check is then made to see whether the job requirement can be immediately (initially) satisfied.

Both of the following are satisfied:

- If a nonzero lead time (LEADTM) was specified on the DB.3.2 or DB.1 panel, has the job that is a requirement run in the specified time frame?
- Has the job that is a requirement run after the dependent job ran? (If the requirement uses a zero LEADTM, only the second test is made.)

Note: The times used in lead time checking are obtained from the database. For a job, examine the LAST-RUN DATE/TIME from the LJOB display. For a data set, use the LCTLG,DSN=*data.set.name* command to examine the last creation date and time CA WA CA 7 Edition recorded for the data set.

Business Value:

This best practice helps ensure the correct sequencing of jobs and improves job processing.

Data Set Requirements for Jobs in the Queue

Use data set requirements to sequence job executions correctly. When jobs are scheduled into the queue for execution, the requirements for this execution of the job (by SCHID) are attached. The master requirement count (MCNT) is incremented for each requirement. Data set requirements are checked to see whether they can be immediately (initially) satisfied.

One of the following is satisfied:

- If a nonzero LEADTM was specified on the DB.3.1 panel, has the data set been created within the specified time frame?
- If LEADTM is zero on the DB.3.1 panel, the data set been created after the last run of the dependent job?

Note: The times used in lead time checking are obtained from the database. The times that are compared can be seen in the LJOB command output under the LAST-RUN DATE/TIME heading. For a data set, use the LCTLG,DSN= command to see the last three creation dates and times.

Satisfaction Lead Time is also known as LEADTM and can be 0, 1-98, and 99. A 0 LEADTM signifies that the requirement must have been met after the last good run of the job that has the requirement. 1-98 is a number of hours that cannot be exceeded because the requirement was met. A value of 99 indicates that the requirement is never initially satisfied.

Business Value:

This best practice helps ensure the correct sequencing of jobs.

SIRD-11 Message

Use REQUIREMENT-LIST: Y on the DB.1 panel for a job. With this option, when that job comes into the request queue an SIRD-11 message is sent to the browse data set. If job requirements exist for the job entering the queue, dates and times are associated with those job requirements on the message.

Business Value:

This best practice helps you understand time calculations on the SIRD-11 message and helps with troubleshooting issues with job predecessors.

Additional Considerations:

The following are examples of the SIRD-11 message.

Note: If the predecessor job is added as a predecessor after the last run of a job, the results are different. That situation is not discussed in these scenarios. Take some precautions (a special watch, at least) the first time a job runs in CA WA CA 7 Edition (or has a new predecessor added).

Example: SIRD-11 message with a predecessor job that is not satisfied

Job USERABC has requirement of USERABC1, which is NOT SATISFIED when it comes into the queue:

```
SIRD-11 **** JOB=USERABC (0003) EXECUTION REQUIREMENTS **** JCLID=000
        SYSTEM= DESC=
        DUE-OUT 06.159/0932 DEAD-LINE 06.159/0932 MAINID=
        SCHID=030 PROSE#=-**NONE** ERQD=000 ENSAT=000 IRQD=001 INSAT=001
        MCNT=002 FLAGS=24/0C/00/02/80/00/02
        *** REQUIREMENTS ***
_____ INTERNAL JOB=USERABC1 DATE/TIME=05233/12:41:31.67 04125/11:15:31
```

The first date/time on the job requirement listed in the SIRD-11 message is the date/time that the requirement should have run to be initially satisfied:

```
LJOB, JOB=USERABC                                DATE=06.159    PAGE 0001
```

```

JOB      ----JCL----  SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME    ID  MEMBER  -NAME-  -ID  -ID-  DSNBR  DSNBR  STP  DDS  RUNS  DATE/TIME

USERABC 000 USERABC          010 ALL  *NONE* 000044 000 000 0000 05233/1241
```

The second date/time on the job requirement listed in the SIRD-11 message is the date/time that the job that is the requirement last ran:

```
LJOB, JOB=USERABC1
JOB   ----JCL---- SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME  ID  MEMBER  -NAME-  -ID  -ID-  DSNBR  DSNBR  STP  DDS  RUNS  DATE/TIME

USERABC1 000 USERABC1      010 ALL *NONE* 000172 001 005 0001 04125/1115
```

Example: SIRD-11 message with a predecessor job that is satisfied

Job USERABC has requirement of USERABC1 which IS SATISFIED when it comes into the queue:

```
SIRD-11 **** JOB=USERABC (0005) EXECUTION REQUIREMENTS **** JCLID=000
        SYSTEM=          DESC=
        DUE-OUT 06.159/0935 DEAD-LINE 06.159/0935 MAINID=
        SCHID=030 PROSE#=#**NONE** ERQD=000 ENSAT=000 IRQD=001 INSAT=000
        MCNT=001 FLAGS=24/0C/00/02/80/00/02
        *** REQUIREMENTS ***
_____ JOB ON HOLD
___X___ INTERNAL JOB=USERABC1  DATE/TIME=06159/08:35:09.53 05233/12:41:31
```

The first date/time on the job requirement listed in the SIRD-11 message is the date/time that the job that is the requirement last ran:

```
LJOB, JOB=USERABC1
JOB   ----JCL---- SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME  ID  MEMBER  -NAME-  -ID  -ID-  DSNBR  DSNBR  STP  DDS  RUNS  DATE/TIME

USERABC1 000 USERABC1      010 ALL *NONE* 000172 001 005 0001 06159/0835
```

The second date/time on the job requirement listed in the SIRD-11 message is the date/time that the job that has the requirement last ran:

```
LJOB, JOB=USERABC
JOB   ----JCL---- SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME  ID  MEMBER  -NAME-  -ID  -ID-  DSNBR  DSNBR  STP  DDS  RUNS  DATE/TIME

USERABC 000 USERABC      010 ALL *NONE* 000044 000 000 0000 05233/1241
```

Example: SIRD-11 message with a predecessor job that has never run

Job USERABC has requirement of USERABC2 which has never run when it comes into the queue:

```
SIRD-11 **** JOB=USERABC (0005) EXECUTION REQUIREMENTS **** JCLID=000
        SYSTEM=          DESC=
        DUE-OUT 06.159/0935 DEAD-LINE 06.159/0935 MAINID=
        SCHID=030 PROSE#=-**NONE** ERQD=000 ENSAT=000 IRQD=001 INSAT=000
        MCNT=001 FLAGS=24/0C/00/02/80/00/02
        *** REQUIREMENTS ***
----- JOB ON HOLD
----- INTERNAL JOB=USERABC2   DATE/TIME=05233/12:41:31
```

```
LJOB, JOB=USERABC2
JOB  ----JCL---- SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME ID  MEMBER  -NAME-  -ID  -ID-  DSNBR  DSNBR  STP DDS RUNS  DATE/TIME

USERABC2 000 USERABC2      010 SY1  *NONE* *NONE* 000 000 0000 00000/0000
```

Only one date/time on the job requirement is listed in the SIRD-11 message, and it is the date/time that the job that has the requirement last ran:

```
LJOB, JOB=USERABC
JOB  ----JCL---- SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME ID  MEMBER  -NAME-  -ID  -ID-  DSNBR  DSNBR  STP DDS RUNS  DATE/TIME

USERABC 000 USERABC      010 ALL  *NONE* 000044 000 000 0000 05233/1241
```

Example: SIRD-11 message with a predecessor job that has run, but the successor job has never run

Job USERABC2, which has never run, has a requirement of USERABC which has run when it comes into the queue:

```
LJOB, JOB=USERABC2
JOB   ----JCL----  SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME  ID  MEMBER  -NAME-  -ID  -ID-  DSNBR  DSNBR  STP  DDS  RUNS  DATE/TIME

USERABC2 000 USERABC2      010 SY1  *NONE* *NONE* 000 000 0000 00000/0000

SIRD-11 **** JOB=USERABC2 (0008) EXECUTION REQUIREMENTS **** (NON-EXECUTABLE)
        SYSTEM=          DESC=
        DUE-OUT 06.159/0952 DEAD-LINE 06.159/0952 MAINID=1
        SCHID=001 PROSE#=#*NONE** ERQD=000 ENSAT=000 IRQD=001 INSAT=001
        MCNT=002 FLAGS=26/0C/20/02/00/40/00
        *** REQUIREMENTS ***
----- JOB ON HOLD
----- INTERNAL JOB=USERABC DATE/TIME=06159/08:51:12.18 05233/12:41:31.67
```

The first date/time on the job requirement listed in the SIRD-11 message is the date/time that the job that has the requirement came into the queue.

The second date/time on the job requirement listed in the SIRD-11 message is the date/time that the job that is the requirement last ran:

```
LJOB, JOB=USERABC
JOB   ----JCL----  SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME  ID  MEMBER  -NAME-  -ID  -ID-  DSNBR  DSNBR  STP  DDS  RUNS  DATE/TIME

USERABC 000 USERABC      010 ALL  *NONE* 000044 000 000 0000 05233/1241
```

Example: SIRD-11 message where both predecessor job and successor job have never run

Job USERABC2, which has never run, has requirement of USERABC3 which has never run when it comes into the queue:

```
LJOB, JOB=USERABC2
JOB  ----JCL---- SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAM  ID  MEMBER  -NAME-  -ID -ID- DSNBR  DSNBR  STP DDS RUNS  DATE/TIME

USERABC2 000 USERABC2      010 SY1  *NONE* *NONE* 000 000 0000 00000/0000
```

```
LJOB, JOB=USERABC3
JOB  ----JCL---- SYSTEM  USR MAIN PROSE  SCHED  --NUMBER OF-  LAST-RUN
NAME ID  MEMBER  -NAME-  -ID -ID- DSNBR  DSNBR  STP DDS RUNS  DATE/TIME

USERABC3 000 USERABC3      010 SY1  *NONE* *NONE* 000 000 0000 00000/0000
```

```
SIRD-11 **** JOB=USERABC2 (0009) EXECUTION REQUIREMENTS **** (NON-EXECUTABLE)
        SYSTEM=          DESC=
        DUE-OUT 06.159/1040 DEAD-LINE 06.159/1040 MAINID=1
        SCHID=001 PROSE#=-**NONE** ERQD=000 ENSAT=000 IRQD=001 INSAT=001
        MCNT=002 FLAGS=26/0C/20/02/00/40/00
        *** REQUIREMENTS ***
----- JOB ON HOLD
----- INTERNAL JOB=USERABC3   DATE/TIME=06159/09:39:04.21
```

Only one date/time is listed on the job requirement listed in the SIRD-11 message, and it is the time that the job with the requirement came into the queue.

Use VRM

Use Virtual Resource Management (VRM) to control sequencing and submission of the workload.

The essential function of CA WA CA 7 Edition is to schedule work and track its progress. A job is considered scheduled by CA WA CA 7 Edition when it enters the request queue. But once work is scheduled, it must be sequenced properly. Jobs have to run in the correct order and not run until all necessary resources are available. CA WA CA 7 Edition determines resource availability in various ways. A resource can be made ready on the completion of a job or when a data set is created. A user command such as VERIFY or POST can also signal the availability of a resource.

The LOAD process discussed earlier in this document defines data set dependencies automatically. Job requirements can be added so that job sequencing is ensured. However, sometimes jobs have to wait on resources other than jobs or data sets. VRM lets you wait. To use VRM, you create a resource profile for the job (RM.1). This profile names the resources that are needed, describes how they are used to control sequencing and when they are no longer needed.

Virtual resources are not evaluated until all the standard resource requirements of the job are met.

Business Value:

Using VRM to control job submission helps ensure that CPU resources are balanced.

Additional Considerations:

The type of virtual resource influences how the job is sequenced. A resource that is defined as shared (SHR) can have any number of concurrent users. Thus, any number of jobs can run in parallel using a shared resource. But if a profile for a job specifies exclusive (EXC) use of a resource, no other job referring to that resource can run while that job runs.

- The resource count (RCT) type is like an SHR resource except that it has a numeric limit on the number of concurrent uses. In the resource profile of the job, you indicate the number of units of the resource that are used. Specify this number as part of the resource name. Then on a different panel (RM.7), declare the global limit of concurrent uses.
- A co-requisite (CRQ) resource is either active (present) or inactive (absent). You activate and deactivate a CRQ resource manually using the PRSQA and PRSQD commands. When you refer to a CRQ resource in a resource profile, you indicate whether the resource must be active (FREE=A) or inactive (FREE=I) to run.
- The address space resource (ASX) is also either active or inactive. The status of these ASX resources is determined through scans of ASCB chains maintained by z/OS on the CA7ONL host computer. When you refer to an ASX resource in a resource profile, you indicate whether the address space must be active (FREE=A) or inactive (FREE=I) to run.

Virtual resource conditions are evaluated while the job is in the ready queue. When all the resource conditions specified in the profile are satisfied, the job is eligible for submission. But if one or more conditions cannot be satisfied, the job waits in the ready queue with a status of W-RSRC. The job remains in this state until all resource conditions are satisfied.

Resource conditions can be satisfied automatically (for example, a job using a resource completes) or in response to a manual command: PRSCF, PRSQA, or PRSQD.

When you define an SHR, EXC or RCT resource, you also indicate when it is to disassociate the resource and job dynamically. The FREE value indicates this time. (This value is also used when defining CRQ and ASX resources, but there the meaning is different. When defining a CRQ or ASX resource, the FREE value indicates the state of the resource that is needed for the job to run.)

The most common disposition is FREE=Y. This disposition means to free the resource only in case of normal termination. If FREE=Y is specified and the job abends, the resource remains associated with the job until the job terminates successfully or is manually freed.

You can indicate that a job frees the resource only in the case of abnormal termination (FREE=A).

A profile can indicate always to free the resource when the job finishes without regard for the completion status of the job(FREE=F).

You can also indicate never to free a resource automatically (FREE=N). Free these resources with a command (PRSCF).

Use ARF

Use the Automated Recovery Facility (ARF) to monitor exception conditions for production jobs and to schedule recovery actions to execute at or near the point of failure.

ARF is flexible in what can be considered an exception. When using a JC (job completion) type ARFSET, ARF processing takes place once a job is requeued back to the request queue upon completion. We do not recommend using a good end of job as an exception and having ARF respond to the exception. This setting causes the job to go immediately through normal completion processing. This processing sometimes causes the ARF responses not to execute.

Business Value:

Using ARF lets you react quickly and without manual intervention for job recovery.

Activate ARF

Define an ARF terminal to use the ARF feature. You define an ARF terminal in the initialization file with GROUP, LINE, TERM, and STATIONS statements. The rules for these statements are as follows:

- The ARF GROUP statement requires a DEVICE=TRXDV.
- The ARF LINE statement requires that the BUFSIZE is 1024.
- The ARF TERM statement requires that the NLINE is 1.
- The ARF TERM statement requires that the LINLEN is 80.
- The TRXDV GROUP and LINE statements require that you include OPEN=YES.

The following example contains these statements:

```
GROUP, NAME=TRXGRP, DEVICE=TRXDV, LNAME=TRXLN, OPEN=YES
LINE, NAME=TRXLN, BUFSIZE=1024, TNAME=TRXT1, OPEN=YES
TERM, NAME=TRXT1, DEVICE=TRXDV, NLINE=1, LINLEN=80
GROUP, NAME=TRXGRP2, DEVICE=TRXDV, LNAME=TRXLN2, OPEN=YES
LINE, NAME=TRXLN2, BUFSIZE=1024, TNAME=TRXT2, OPEN=YES
TERM, NAME=TRXT2, DEVICE=TRXDV, NLINE=1, LINLEN=80
.....
STATIONS, TRMID=TRXT1, STANIDS=(TRX)
STATIONS, TRMID=TRXT2, STANIDS=(TRX2)
```

Update the initialization file RESTART statement with ARF=YES to use ARF.

Note: Consider increasing the region size that CA7ONL uses.

Verify that you have at least two TRXDV terminals defined.

Business Value:

An ARF terminal is required to issue some ARF responses.

More Information:

For more information about GROUP, LINE, TERM, and STATIONS statements, see the *Systems Programming Guide*. For more information about ARF, see the *Database Maintenance Guide*.

ARF High Return Code (HRC) Test

Use the high return code feature of ARF to improve processing.

Starting with r11.3, an enhancement addresses a new ARF high condition code intercept. Jobs using ARFSETs are tested at different points during job execution. SC, step completion, and JC, job completion, tests are done after each step completion reported by SMF.

Currently, the condition codes of previous or subsequent steps do not affect condition code tests done after each step. A new completion code test format is added to the CC/ABENDS field of HRC, meaning that a high return code test is done. HRC tests are only done at the end of a job and are met when the highest return code from *any* of the jobs steps fits the test.

In previous releases, assume that you coded an ARFSET with responses to issue for return code 0. If *any* step within the job gets a 0, the ARFSET would *hit* and responses done. This enhancement means that the customer can code an ARFSET to *hit* only if the highest return code from any step meets the criteria. The test is made similar to the DB.1 CC and RO tests.

Business Value:

This best practice lets you test for and react to a high return code from any step in a job.

Define an ARFSET to Catch Any Bad Job Completion

Use the following input for the CC/ABENDS fields when defining an ARFSET to catch any of the following bad job completions:

- System abend
- User abend
- Condition code other than 0
- JCL error

```
CC/ABENDS : SYS GE 000 / USR GE 000 / CC GT 000 /  
           JCL EQ 000 / FL EQ 000  __ ??? GE 000
```

Business Value:

This best practice lets you react to all bad job completions.

Define an ARFSET to Limit Submission Time

Use the following to set up an ARFSET to prevent job submission after a certain time of day. Use the type-specific tests fields to define condition tests for EC, EE, IS, LB, LE, and LS condition types. The type on the ARFSET is IS (for interrupt submission). When this condition *hits* (in this example, after 12:00 on the job's scheduled days, first the RESPONSES (if any) are done and then the FINAL DISP.

```
EC, EE, IS, LB, LE, LS TST:
      R0: GE  DATE: DOD      TIME: 1200      AO: ?  INT/ADJ:
```

Business Value:

This best practice lets you control the processing of jobs past a certain time of day.

Define an ARFSET for an LA Condition

Define an ARSET for time-sensitive jobs for late conditions. If you have jobs that are time-sensitive, we recommend that you define an ARFSET to issue messages (or CA WA CA 7 Edition commands) when the job becomes LATE. If you have an ARFSET for an LA condition, the ARF response is executed only if the initial prompt (SCNP-11) occurs when the job becomes LATE in the request queue.

Business Value:

This best practice lets you react quickly to a LATE status on jobs.

Use ARF Variables

A set of ARF variables is available to use in the ARF actions. The actions are taken when a condition has occurred. The variables include items such as job name, system name, schedule ID, and more. Variables permit the flexibility of defining a single ARF definition that multiple jobs can use. An example of using variables in an ARF action is the following, which sends a highlighted message (WTO) to the console with the job name, both the JES and CA WA CA 7 Edition job number:

```
AM,CM=H,M=** LOOK!  &JOBNAME(&JES# &JOB#) ABENDED!  **
```


Chapter 9: Security Best Practices

As a best practice, we strongly urge customers to use all available external security options with CA WA CA 7 Edition. Use of external security to control access to CA WA CA 7 Edition resources lets you centralize the maintenance of security. Existing USERID definitions and data set access rules are used in the CA7ONL environment by using the full external security interface.

No special considerations are required when you are using CA ACF2 or CA Top Secret for the external security interface.

Business Value:

Following these guidelines optimizes access to product functions to help meet the requirements of your environment.

This section contains the following topics:

[Make RACF Changes](#) (see page 87)

[Use UID Security](#) (see page 88)

[Multiple Job Security](#) (see page 90)

Make RACF Changes

Make RACF changes available to CA7ONL without a restart. CA WA CA 7 Edition calls made to RACF through CA Common Services do a FRACHECK. For this reason, restart CA7ONL to pick up changes to RACF security definitions.

We recommend that you change this processing so that a bounce of CA7ONL is not required after making any RACF changes. Change the FRACHECK to a RACHECK, which does not require a restart to refresh the security definitions. Change the RESTABLE (Resource Class Table) for PA@EL and SU@MIT in the CAS9 module CAS9SAFC from the following:

```
DC C'PANEL ',C'PA@EL ',X'00',X'80',X'00',X'00'  
DC C'SUBMIT',C'SU@MIT ',X'00',X'80',X'00',X'00'
```

To the following:

```
DC C'PANEL ',C'PA@EL ',X'00',X'00',X'00',X'00'  
DC C'SUBMIT',C'SU@MIT ',X'00',X'00',X'00',X'00'
```

After you update CAS9SAFC, reassemble and link-edit it using the CAS9 SAMPJCL member CAS9CSSF. Next, reinitialize the CAS9 module CAISSF with the following command in CAIRIM:

```
CAIRIM REFRESH(SSF)
```

Business Value:

This best practice enables continuous processing.

Additional Considerations:

Sites can decide to use other resource classes (CALENDAR, RESOURCE, and others) instead of the defaults. Examine your initialization file SECURITY statement, and change RESTABLE appropriately.

Use UID Security

Use UID security to control who can access which jobs.

Business Value:

Using external security for UIDs lets you change UID security without having to restart CA7ONL.

Additional Considerations:

Use the CA WA CA 7 Edition UID, user security identification, to control which users have access to which jobs. You can find the UID on the DB.1 panel.

To use external security to implement UID security controls

1. Decide whether you want to use the default UID table, SASSRTBL, or create a site-specific UID table. If you want to build your own table, the source for SASSRTBL is in the CAL2SRC library and serves as a model. Update the CA WA CA 7 Edition SMP/E environment using USERMOD AL2UM09 in the CAL2OPTN library.
2. Add the keyword `UID=aaaaaaa` on the SECURITY statement in the initialization file, where `aaaaaaa` is SASSRTBL for the default module or the name of the module that you created.
3. Authorize, through your external security package, user IDs to have access to resource names in the SASSRTBL (or site-specific module) that correlate to the UID value. This authorization occurs under the PANEL resource class, or another class as defined by the RCLASS keyword on the SECURITY statement in the initialization file.
4. Place the resource name that a user is permitted to use in the user's profile using the `/PROFS` command.
5. Verify that a UID value is associated with each job through the job definition panel. This value is used to determine which users have access to the job.

How it works:

When a user logs into CA7ONL, the user's profile is examined for the UID resource name (added through the `/PROFS` command). A security call is made to verify that the user has access to that UID resource. The UID resource is then resolved into a UID number from the SASSRTBL (or site-specific module).

For example, assume that you give USER1 the profile resource of CA70001. When USER1 logs on to CA WA CA 7 Edition, a call is made to external security to see whether they have access to the resource of CA70001. If they do, then their UID value is set as 001, meaning that they can only access jobs with a UID value on the job definition panel of 1 or 0.

Without a `UID=` on the SECURITY statement, your internal security module is checked to see whether the USERID is present. If so and if it has a UID value, that value is used. If you have assigned COIDs, you continue to use your `USER=` module for this purpose.

Multiple Job Security

Use the SECURITY statement MULTIJOB parameter to prevent multiple jobs being submitted from one JCL member.

The MULTIJOB parameter prevents CA WA CA 7 Edition from submitting multiple jobs when it submits a production job. These secondary jobs can inherit a user ID that is more powerful than the one under which the job usually executes. The options include FLUSH (non-Parallel Submission Processing (PSP) systems) and REQUEUE (PSP systems) to prevent the submission of the job. With PSP systems, the job is queued to the request queue with a status of MJOB.

Note: CA WA CA 7 Edition does not know or track these secondary jobs.

Business Value:

This best practice prevents excess security authority for jobs that CA WA CA 7 Edition does not directly control.

Chapter 10: Reporting Best Practices

CA WA CA 7 Edition has robust reporting capabilities. You can report on what is current or from historical data in various ways. Many different commands and report engines area are available. Following these recommendations maximizes the reporting capabilities of the product.

This section contains the following topics:

[Use BTI or CAICCI](#) (see page 91)

[Use Program SASSHIS8](#) (see page 91)

[Use CA Earl or CA Easytrieve](#) (see page 92)

[Use APA Graphs](#) (see page 93)

[Use SQL Reports](#) (see page 93)

Use BTI or CAICCI

Use BTI (Batch Terminal Input) or CAICCI to issue commands and receive the output. Use the commands in the LQ family of commands (LQ, LRDY, LACT, and so on) to watch the current processing. Use the LPRRN and LRLOG commands for some historical run information about your jobs. Use forecasting commands to create a listing of jobs that are scheduled to run during a specified time frame.

Business Value:

BTI and CAICCI let you access CA7ONL from batch JCL or a user program. BTI also handles large amounts of output.

Use Program SASSHIS8

Use program SASSHIS8 for history reporting. CA WA CA 7 Edition creates log data set records that the history reporting facility uses. SASSHIS8 is a report engine that extracts CA WA CA 7 Edition log records from combinations of the log, log history, and log archives files based on control statements. You can obtain a variety of reports.

Business Value:

This best practice provides reports about processing at your site.

More Information:

For more information about available history reports, see the *Report Reference Guide*.

Use CA Earl or CA Easytrieve

Use CA Earl or CA Easytrieve for reporting. CA WA CA 7 Edition provides the source statements to create a standard set of reports using either the CA Earl or CA Easytrieve report languages. The CA Common Services include abbreviated versions of CA Earl and CA Easytrieve at no extra charge.

CA WA CA 7 Edition provides the report definition statements and JCL that both products require to create an identical set of standard reports. The reports serve a number of purposes including monitoring system performance, workload performance, manual workload intervention activities, and so forth.

Business Value:

This best practice lets you report about processing at your site.

More Information:

For more information about the available CA Earl or CA Easytrieve reports, see the *Report Reference Guide*.

Use APA Graphs

Use Automated Performance Analysis (APA) graphs for statistical reporting. CA WA CA 7 Edition maintains various counters for many events, such as how many jobs were added. These counters are by each day and start at 0 at midnight. You can request any 24-hour day occurring up to two years prior if the statistics file has not been reinitialized.

Business Value:

APA graphs present daily activity information in a numbered format.

More Information:

For more information about the APA Graphs, see the *CA WA CA 7 Edition Reports Reference Guide*. For more information about the APA command, see the *CA WA CA 7 Edition Command Reference Guide*.

Use SQL Reports

Use the SQL reports for your database reporting. The definition and active workload (queue) data is now found in a relational database. SQL, a query language, can retrieve data from the database that is based on desired selections. SQL is an easy language to learn. Tutorials are available on the Internet. You can create your own reports instead of depending on the canned reports that CA WA CA 7 Edition supplies.

Business Value:

SQL permits users to query the database for specific data in which they are interested.

Chapter 11: Disaster Recovery Best Practices

Obviously, Information Technology (IT) is the backbone of most organizations. Seamless, continuous processing is the goal of most data centers. Business computing has become sophisticated and at the same time complex. When all goes according to plan, IT is invisible except to the ones *in the trenches*.

However, when a data center must go into disaster mode, a thought out plan of preparedness lets an organization continue seamlessly. The basic strategy for building a disaster recovery plan involves establishing a clear and comprehensive policy that helps ensure business continuity. Understanding and including the key components of the business is the primary goal. The complex nature of disaster planning including evaluating the associated risk and potential impact with testing and auditing requirements makes it an absolute necessity that each component participates and offers a well-defined approach to disaster recovery.

This section contains the following topics:

[DR Mode](#) (see page 95)

[Disaster Recovery for a Scheduled Outage](#) (see page 97)

[Failover and Recovery for an Unscheduled Outage](#) (see page 101)

[Quiesce Job Submission](#) (see page 103)

DR Mode

Running CA WA CA 7 Edition for job scheduling facilitates restoring batch processing after a *system down* disaster. The nature of the outage and the level of preplanning is an important factor in determining disaster recovery procedures.

The initialization file has two statements, DRMODE and DRCLASS, which are used to define the options for Disaster Recovery and to activate DR Mode.

The DRCLASS statement can specify one or more disaster recovery classes that are active.

Business Value:

In CA WA CA 7 Edition, the Disaster Recovery Mode option permits the inclusion of only specified workloads in processing. Identifying the appropriate components for a given disaster is obviously one of the major challenges. The DRMODE option allows for a global default disaster class and the ability to define a specific job to a particular class. When running in Disaster Mode, the Schedule Scan function makes the appropriate determination of what work to submit based on the active disaster recovery classes.

Additional Considerations:

The following are Disaster Recovery Mode examples.

Example: Define a class to include jobs without DRMODEs

A default class can specify to include all jobs without a DRMODE defined. You can also use a special reserved word, @SYSTEM to use the job's SYSTEM name as the disaster class.

```
DRCLASS,DEFCLASS=@SYSTEM
```

Example: Define default disaster recovery class DR1

This example states that the default disaster recovery class is DR1. This default applies to all jobs without a disaster class defined.

```
DRMODE,DEFCLASS=DR1
```

Some special considerations apply when running in Disaster Recovery Mode. Do you want to consider requirements for jobs during recovery? Do the requirements apply to triggers also? Should a job not in an active DR class that a job in an active DR class triggers run nonexecutable? Two keywords control the handling of requirements and triggers.

For requirements, set the RQMTS= keyword to DR or ALL. If the predecessor job's disaster recovery class is not active, the DR value indicates that when running in DR MODE, job requirements are automatically satisfied. In some cases, predecessor jobs never run in this particular disaster mode, and you want the requirement satisfied. The RQMTS=ALL option states that normal job requirement processing occurs.

Example: Default class and satisfied requirements

This example states that the default disaster recovery class is DR1 and that requirements are satisfied for jobs whose disaster class is not active.

```
DRMODE,DEFCLASS=DR1,RQMTS=DR
```

Example: Default class and normal processing

This example states that the default disaster recovery class for jobs without a DRCLASS designation is DR1 and that normal job requirement processing occurs.

```
DRMODE,DEFCLASS=DR1,RQMTS=ALL
```

The TRIGGERS= keyword is used to specify how trigger processing occurs during disaster recovery processing. The default TRIGGERS=DR mode indicates that only jobs with an active disaster recovery class are triggered. The TRIGGERS=ALL mode indicates that normal trigger processing occurs.

Example: Default class, normal processing, and some triggering

This example states that the default disaster recovery class for jobs without a DRCLASS designation is DR1. Normal job requirement processing occurs. Only jobs with an active disaster recovery class are triggered.

```
DRMODE,DEFCLASS=DR1,RQMTS=ALL,TRIGGERS=DR
```

Starting with r11.3, you have another option for triggering: TRIGGERS=NONEXEC. This setting lets triggered jobs process through the CA WA CA 7 Edition queues as nonexecutable if a job is in an inactive DR class. This setting permits the full job stream to enter the queue but only executes the active DRCLASS jobs.

The VERIFY keyword on the DRMODE statement lets operators confirm a disaster recovery mode startup. If CA WA CA 7 Edition is started with the DRMODE=YES EXEC parameter and the DRMODE initialization file statement includes the VERIFY=YES keyword, a WTOR is issued to allow for the confirmation of the disaster mode startup. The default is VERIFY=NO.

Disaster Recovery for a Scheduled Outage

Use this best practice to schedule downtimes on CA7ONL.

In a scheduled outage, you can use CA WA CA 7 Edition easily for effective recovery. You can stop job/task scheduling and submission to quiesce the batch processing before system shutdown. You can use CA WA CA 7 Edition data to ensure that processing continues with less disruption of service.

In the initial preparation for the outage, determine whether key CA7ONL files can handle a possibly *backed-up* workload. Start with an estimate of when and how long the system is down, and forecast the workload for that time frame. In other words, if the system outage is estimated to be 4-6 hours, forecast that date/spanned time with the FJOB command. If you have set up your jobs with the DRCLASS designations, you can forecast only those jobs with an activated DRCLASS. Get a count of jobs forecasted adding the number of jobs that are historically already in the queue then.

CA WA CA 7 Edition makes it easy to stop and restart the batch processing in an orderly manner. Use the `STOP,Q=ALL` command to halt the automatic submission of jobs. Any jobs that are already submitted continue to process on the operating system and go through completion processing when finished, but no new work is submitted to JES. You can stop the schedule scan with an `SSCAN,TIME=0` command.

Before shut down, issue an `SSCAN` command and note the `NEXT SCAN PERIOD START TIME`. This time is the time frame that would be used to bring jobs into the queue with a start time that would fall between that time and the calculated scan end time (discussed later) on its next job scan. This information is important when readying the system to resume normal processing.

Shut down CA7ONL. After a successful shutdown, run the log dump jobs against both log files (LOGP and LOGS) as 'insurance' for a seamless recovery.

When it is time to restart CA7ONL, the time noted on the `SSCAN` output (`NEXT SCAN PERIOD START TIME`) is important. If the date/time of the restart is before that time, start with the schedule scan and the queues active. If the date/time of the restart is after that time, steps are required to get back to the initial point of stoppage.

If the restart is past the noted `NEXT SCAN PERIOD START TIME`, start CA7ONL with the schedule scan deactivated and the queues stopped. (To deactivate the schedule scan at startup, use `RUNOPT=NSTA` on the `INIT` statement in the initialization file. And, to bring up CA WA CA 7 Edition with the queues stopped, use `STOPQ=YES` on the `SCHEDULE` statement in the initialization file.)

To resume job submission, enter a `START,Q=ALL` command. To resume the schedule scan automated facilities requires that you enter a series of commands. When CA7ONL is started with a cold type of start, schedule scan is set to current time. To restart schedule scan, set the `NEXT SCAN PERIOD START TIME` with an `SSCAN,DATE=yyddd,PERSTART=hhmm` command to set the `NEXT SCAN PERIOD START TIME` to the date/time noted in the `SSCAN` output from before the shutdown.

After the `NEXT SCAN PERIOD START TIME` has been set, issuing `SSCAN,SCAN=SCH` wakes schedule scan immediately and looks for jobs with a start time that falls between the `NEXT SCAN PERIOD START TIME` and a calculated scan end time. (The calculation for scan end time is current time (time schedule scan started) plus `SPAN` plus `QDWELL`.) If this time frame would bring in too many jobs, you can set the schedule scan to work with *intervals* by adding a `PEREND` keyword to the `SSCAN` command as follows:

```
SSCAN,DATE=yyddd,PERSTART=hhmm,PEREND=hhmm
```

The scan end time is then set to the value entered on `PEREND`.

After the begin and end time for the schedule scan is set, issue an SSCAN, SCAN=SCH command. This command wakes up schedule scan for one scan only (if set with the PEREND). Automatic wake-up is turned off until a scan is done without the PEREND parameter. When all goes well, resumes normal processing.

Note: Verify that the PERFORM option of 5 on the INIT statement in the initialization file is not used in these situations:

- You are using schedule scan to bring in intervals of work.
- You are rescanning an already scanned time frame after the first wake-up.

In other words, when rescanning an already scanned time frame, duplicate checking is done during the first scan that occurs. However, if rescanning occurs on any subsequent schedule scans, you can bring in duplicate work when using PERFORM=5 on the INIT statement in the initialization file.

Business Value:

Following these guidelines ensures that CA7ONL downtime is minimized.

Additional Considerations:

To recap:

- Make sure that queue/support files are large enough to handle backed up workload
 - FJOB
 - /DISPLAY,Q=ALL
- Quiesce job submission
 - STOP,Q=ALL
 - SSCAN,TIME=0

Note the NEXT SCAN PERIOD START TIME from SSCAN output
- Shut down CA7ONL
 - /SHUTDOWN,Zn
- Dump the CA WA CA 7 Edition primary and secondary log files
 - UCC7LOGP and UCC7LOGS jobs
- If the restart time is before the noted NEXT SCAN PERIOD START TIME, restart CA7ONL
 - S CA7ONL
- If the restart time is after the noted NEXT SCAN PERIOD START TIME, change the initialization file settings and then restart CA7ONL
 - RUNOPT=NSTA on the INIT statement
 - STOPQ=YES on the SCHEDULE statement
- When the normal processing is to occur
 - Issue START,Q=ALL command to start the job submission
 - Use the SSCAN commands to set date/time and turn on the schedule scan

More information:

[Quiesce Job Submission](#) (see page 103)

Failover and Recovery for an Unscheduled Outage

Use this best practice to react to unscheduled downtimes on CA7ONL.

A system failure or an unscheduled outage is the dread of all IT personnel. Though many other processes must be taken into account, getting CA WA CA 7 Edition back up and submitting batch work is a priority. If the outage is not DASD-related, consider having a dormant copy of CA7ONL (TYPE=DORM) automatically start on another LPAR in the sysplex. You can also set up CA7ONL and ICOM to be eligible for ARM restarts.

We recommend that you use the CA Datacom/AD Shadow MUF facility so that the database is available when an LPAR fails. Have the Shadow MUF execute on the system where the dormant copy of CA7ONL executes. If IBM ARM is used for restart CA7ONL, have it start CA7ONL on the system where the Shadow MUF executes.

To minimize the failover time, or the amount of time CA WA CA 7 Edition is not present in the sysplex, a dormant copy of CA WA CA 7 Edition is started on the same or a different LPAR in the sysplex. The dormant copy initializes to a point where it looks for the CA WA CA 7 Edition enqueue presence. The enqueue major/minor name is UCC7SVT/CA7n, where *n* is the instance name. If the enqueue is present in the sysplex, the dormant copy of CA WA CA 7 Edition waits on a timed interval. When it wakes up, it looks for the presence of the enqueue name again. If the enqueue from the primary copy of CA7ONL is not present, the dormant copy issues a WTOR. The dormant copy starts taking over the functions of the normal CA WA CA 7 Edition, such as submitting jobs and processing job completions.

CA WA CA 7 Edition includes an interface to the IBM Automatic Restart Manager (ARM). You can use this interface instead of the dormant copy. You would define an ARM policy for CA WA CA 7 Edition in the sysplex. If CA7ONL or ICOM abend in the system, the active ARM policy indicates to start CA7ONL or ICOM again (either on the same LPAR or on a different LPAR).

Note: The business use for using a dormant copy or ARM is that maintaining an active CA7ONL in the system permits almost continuous operations of job submission and tracking.

In a worst case scenario, the outage is that you do not have queue files. To recover from this type of outage, the log files are key to recovery under most circumstances. If possible, before you restart CA7ONL, run the log dump job against both primary and secondary log files to ensure that the historical data is as up-to-date as possible.

When it is time for processing to resume, start CA7ONL with a TYPE=COLD and the initialization file setting disabling schedule scan and stopping the queues. This start is a COLD type start, which also formats the queue data (all queue data is lost).

With the history file created from the log file dump job, data is available for input to the Recovery Aid program. To run the Recovery Aid program, execute SASSHIS8 with a 50 control card. This program produces reports and a batch file. The reports include the output from an LQ command from the point of failure. The batch file contains DEMAND(H) commands for the jobs that were in the request, ready, and active queues.

After you restart CA7ONL, run a batch terminal interface job with the commands produced by the Recovery Aid program. Jobs that were in the queues at the time of failure are DEMANDED into the queue, ready for processing. Jobs that were in the active queue or ready queue and had already been submitted have DEMAND commands created with a TYPE=RES enabling these jobs to be restarted or force completed.

From this point, the course of action is determined by whether to resume normal processing or only a subset of the workload runs.

To resume normal processing, follow the directions for setting and turning on schedule scan from the previous information about a disaster recovery scheduled outage. You do not have a NEXT SCAN PERIOD START TIME to model the date and time input from, but you can use the time of the outage.

To run only a subset of the normal workload, start the queues, forecast the workload, and then demand the workload into the queues. As an alternative, issue a HOLD,Q=ALL command to place a HOLD on all work in the queues, an SSCAN,SCAN=HLD so that all work that enters the queue after has a hold requirement, set/turn on schedule scan and then release or cancel jobs in the queues as needed.

Business Value:

Following these guidelines ensures that CA7ONL downtime is minimized.

Additional Considerations:

To recap:

- Dump the log files (if possible)
 - UCC7LOGP and UCC7LOGS jobs
- Run the Recovery Aid program
- Start CA7ONL with TYPE=COLD
 - RUNOPT=NSTA on the INIT statement
 - STOPQ=YES on the SCHEDULE statement

- If your normal processing is to occur from this point, ensure that queue/support files are large enough to handle possibly backed-up workload
 - FJOB
 - /DISPLAY,Q=ALL
- Execute the BTI job with an output file from the Recovery Aid program
- Issue START,Q=ALL command to start the job submission
- Use the SSCAN commands to set date/time and turn on the schedule scan
- If only a subset of the processing is to occur
 - HOLD,Q=ALL
 - SSCAN,SCAN=HLD
 - DEMAND new work as required
 - Release queue records when ready to process them

More Information:

For more information about the Recovery Aid programs, see the *Systems Programming Guide*. For more information about the SASSHIS8 program, see the *Report Reference Guide*.

More information:

[Quiesce Job Submission](#) (see page 103)

Quiesce Job Submission

During scheduled outages, it is sometimes best to quiesce job submission. This topic discusses the best practices for stopping job submission when preparing for a system IPL or other outage.

Enter the following top line commands:

```
HOLD,Q=REQ  
HOLD,Q=RDY  
SSCAN,SCAN=HLD
```

These commands result in the following:

- Adding a HOLD requirement to all jobs currently in the queues.
- Placing a HOLD requirement on all jobs that later come into the request queue.

You can stop schedule scan from waking up and bringing in work (turn off schedule scan) by issuing the following command:

```
SSCAN, TIME=0
```

If any jobs are active, you can wait for their completion if you prefer.

If you have not shut down CA7ONL, issue the following commands when you are ready to start regular processing:

```
RELEASE, Q=REQ  
RELEASE, Q=RDY  
SSCAN, SCAN=REL
```

These commands post the HOLD requirements added to jobs by the earlier HOLD commands. These commands stop the adding of a HOLD requirement on all jobs that later come into the request queue (negates the SSCAN,SCAN=HLD command entered earlier). You can RELEASE the HOLD requirement on a job by job basis when you do not want to post all jobs that are held.

If you shut down CA7ONL after quiescing, consider starting CA7ONL with schedule scan turned off. To do so, change the initialization file INIT statement to RUNOPT=NSTA. You can also stop job queue movement (going from the request queue to the ready queue for submit) by changing the SCHEDULE statement in the initialization file, adding STOPQ=YES.

If you start CA7ONL with schedule scan turned off, you can turn it on with the following top line command:

```
SSCAN, SCAN=SCH
```

And if you chose to stop the queues (STOPQ=YES), you can start them with the following command:

```
START, Q=ALL
```

Next, you can issue the RELEASE commands described previously to post the HOLD requirements.

Business Value:

Quiescing job submission helps ensure that nothing is submitted without you first performing checks on your system.

Chapter 12: Troubleshooting Best Practices

This section contains the following topics:

[Negative Dependencies](#) (see page 105)

[Tracking](#) (see page 106)

[Locked in LOAD Status](#) (see page 107)

[Not Tracking](#) (see page 108)

[SASSXTRK Program](#) (see page 109)

[Dumps of CA7ONL](#) (see page 111)

Negative Dependencies

Why do jobs with negative dependencies sometimes run when a job they are negatively dependent on abends?

Jobs with negative requirements cannot run together. However, if a job that is a negative requirement is in the request queue in restart status when a job that is negatively dependent on it submits, it does not hold.

When a job with negative dependencies enters the queue, the negative dependencies are *initially posted* requirements. When a job with negative dependencies is ready to submit, a check is made to see if any of the jobs it is negatively dependent on are submitted or active. If so, the job is requeued back to the request queue and the negative requirement is *unposted* to make it a job requirement. This result means that the "unposted" job must complete successfully to post this requirement and let the job move to the ready queue for submission. If the negative dependencies are not active at the time the job is to submit, no dependency is set. Consider using Virtual Resource Management (VRM) instead of negative dependencies because with negative dependencies, one of the jobs can abend and others can run. In other words, if you do not want the jobs to execute at the same time and you do not want them to execute when any others have failed, use VRM with an EXC (exclusive) resource. With this process, you can set whether a job must end successfully to free the resource.

Business Value:

Understanding how negative dependencies work helps when troubleshooting an issue with negative dependencies.

Tracking

Why do some jobs not track in CA WA CA 7 Edition (remain in the ready queue) while others are tracking?

In a JES2 shared pool environment, initialize CA WA CA 7 Edition with the CAIRIM procedure on every LPAR in the sysplex even if no CA WA CA 7 Edition submitted jobs run on that LPAR. If no CA WA CA 7 Edition submitted jobs run on that LPAR you do not have to start ICOM, but perform the CAIRIM initialization of the CA WA CA 7 Edition environment.

When CA WA CA 7 Edition submits a job, an indicator is put in the first JOB statement. When the job goes through the converter interpreter and the IEFUJV exit, it is recognized as CA WA CA 7 Edition submitted. An indicator is placed in the SMF common area. When subsequent SMF records are cut for the job (job start, step end, job end, and so forth), our exits can collect this data and feed it back to CA WA CA 7 Edition through ICOM. In a shared JES environment, JES decides where the job goes through the converter interpreter. Usually a job is converted on the LPAR where it is submitted, but during heavy processing JES can pass this process to another LPAR.

If a job converts on an LPAR that does not have the CA WA CA 7 Edition environment initialized, CA7ONL does not track this execution. To see whether the job was converted on an initialized LPAR, look in the JESLOG of the job for the following IBM message:

```
IEFC025I  INSTALLATION MODIFIED JCL
```

The presence of this message means that when the job converted, the CA WA CA 7 Edition environment was present on that LPAR. The absence of this message means that it was not, and if it is not present, the job does not track in CA7ONL.

Business Value:

Understanding how work is tracked helps when troubleshooting an issue with task tracking.

Locked in LOAD Status

Why are jobs LOCKED IN LOAD STATUS in CA WA CA 7 Edition?

If a job must have entered the queue but did not, look at the browse data set for messages produced during the time it would have entered the queue. If you see the message SP07-03, SATO-35, or SCNJ-15 saying that the job is in LOCKED STATUS, issue an LLOCK,JOB=* top line command. If the job is still in LOCKED status, a stated reason is present.

If the reason is LOCKED IN LOAD, the job needs a successful load before it schedules into the queue. Do a LOAD,JOB=*jobname* command. Let the load execute and the job flush from the request queue. Next, look at the browse file for a message of LOAD SUCCESSFUL or LOAD UNSUCCESSFUL. Usually, you can find the name of this file by looking for the ddname of BROWSE in the CA7ONL procedure.

If the load is successful, nothing else is required.

If the load is unsuccessful, the message gives insight as to what in the load data was not able to be added or updated in the database.

If you must get a successful load of a job and cannot readily fix the issue (perhaps the issue is with database size and you cannot REORG the database right now), you can do the following to get the job loaded. Do a LOADH on the job. Edit the QJCL, changing it to execute only an IEFBR14. Next, release the hold and let the LOAD continue with the updated JCL. When you can LOAD the *real* JCL for this job, fix the reason for unsuccessful LOAD.

Business Value:

Understanding how the load process works helps when troubleshooting an issue with load processing.

Not Tracking

What do I check if no jobs are tracking?

CA WA CA 7 Edition tracks work that it submits using SMF feedback.

The failure to track work in CA7ONL is reported in many ways. Jobs that complete do not trigger their follow-up work, jobs hang in the ready queue, and so forth, but these are symptoms of CA WA CA 7 Edition not tracking the work it is submitting.

If some jobs track and others do not, verify that security is not causing the problem. Look for a JOB FLUSHED message on OS console. The message probably has no job name.

Jobs in the ready queue are either ready to submit or have already been submitted and moved to the active queue when a job initiation record is received. To verify whether a job is already submitted, look at the output from an LQ command (LQ, LRDY, and so on). The output has a SUB/START DATE/TIME heading. For a job in the ready queue, a value in the SUB/START DATE/TIME field of *NONE* means that the job is not submitted. If a date and time are in this field, that time is the time that the JCL was written to internal reader or submit data set. LRDYP,JOB=,LIST=STATUS show a nonsubmittal reason if *NONE* is in this field.

If using submit data sets and a date/time is in the SUB/START DATE/TIME field, check to see whether the submit data sets are used correctly between CA7ONL and ICOM.

If a date/time is in the SUB/START DATE/TIME field on the LRDY output, and the job has run, check to be sure that ICOM is active on the LPAR where the job is running. Check the communications data set name in the CA7ONL and ICOM procedures to verify that they are the same file. When both programs start, a CA-7.INCD message states the volume where the communications data set resides and whether it is shared.

Check the SVCNO statement in the initialization file for SASSVC=NO. For job tracking, set this keyword as SASSVC=YES.

If the CA WA CA 7 Edition environment was reinitialized with CAIRIM, consider recycling CA7ONL and ICOM.

You can issue CAISMFU as a command from ISPF option 6 or TSO.

The output should show SASSUJV, SASSU83 and SASSU84 active:

```

CAS9001I  Interface Summary:
CAS9002I
CAS9003I  Name      Vers  Description      Status
CAS9004I  CAS9U83  S910  CAIRIM INTERCEPT  Active
CAS9004I  CAS9U84  S910  CAIRIM INTERCEPT  Active
CAS9004I  CAS9ATRT S910  CAIRIM INTERCEPT  Active
CAS9004I  SMFISDMP SF15  JARS/SMF INITIATOR  Active
CAS9004I  $IXRUJSI L141  IXR JOB INIT EXIT   Active
CAS9004I  TLMSSMFE TL55  TLMS SMF EXIT       Active
CAS9004I  SASSUJV  L2B0  CA-7 JOB VAL EXIT   Active
CAS9004I  SASSU83  L2B0  CA-7 SVC 83  EXIT   Active
CAS9004I  SASSU84  L2B0  CA-7 SVC 84  EXIT   Active
CAS9004I  JR51SMFE 7.1  JARS ONLINE CHARGES Active
CAS9004I  CASEJOB  W110  CA-ENF Job Init     Active
CAS9004I  CASESM83 W110  CA-ENF Intercept    Active
CAS9004I  CASESM84 W110  CA-ENF Intercept    Active
CAS9002I
CAS9005I  Number of interfaces 13  Number of calls processed 684533

```

Run the U7TESTSV program. Issue a DEMAND,JOB=CA07SVCT command, and check the JES output of this job (you may have to do a DEMANDH and then edit the QJCL to route the job to various LPARS if you have more than one). The messages produced by this job are documented in the *Message Reference Guide*.

Run the CAL2OPTN member AL2ENVR (if you use release 11.1, the member name is CAL2ENVR in SAMPJCL) to get an environmental report.

Business Value:

Understanding how work is tracked helps when troubleshooting an issue with task tracking.

SASSXTRK Program

You are sometimes asked to run the program SASSXTRK so that CA Support can view your CA WA CA 7 Edition log file for problem solving. SASSXTRK is a program that is used to extract records from the CA WA CA 7 Edition log file. The SASSXTRK program sample JCL is provided in the sample JCL CAL2JCL as member AL2ZXTRK.

This program produces a control statement edit report and an output file that contains the extracted CA WA CA 7 Edition log records. Other reporting jobs can use this output file to produce specific reports. Depending on the SYSIN, the program can extract records based on a date and time, a job name, or both.

Use the following sample JCL to execute the SASSXTRK program:

```
//jobname JOB ...
//EXTRACT EXEC PGM=SASSXTRK
//STEPLIB DD DISP=SHR,DSN=ca7.ca12load
//LOGIN DD DISP=SHR,DSN=ca7.logtape
//LOGOUT DD DISP=(NEW,CATLG,DELETE),DSN=extracted.file,
// UNIT=SYSDA,SPACE=(CYL,(10,5),RLSE),
// DCB=(RECFM=VB,LRECL=2100,BLKSIZE=21004)
//SYSOUT DD SYSOUT=*
//SYSIN DD *
* yydddhmm yydddhmm
/*
```

The LOGIN DD is required and can be either the log or history files. The LRECL must be 2100.

The LOGOUT DD is required and can be DASD or tape. The LRECL is copied from the LOGIN DD statement. This file contains a copy of the log records that were selected.

The SYSOUT DD is required and contains information from the control statement edit routine and possibly some error messages.

The SYSIN DD is required and is the control statement for selecting records to write to LOGOUT. The control statement has the following format:

```
* yydddhmm yydddhmm
```

You can specify a job name on the control statement starting in column 1. If you do, only records with the matching job name are extracted. If you are running this utility to provide log data for CA Support, use an asterisk (*) in column 1 to denote a generic request. The generic requests extract *all* records for the requested time frame.

Business Value:

Understanding how to obtain log records for a specified time frame helps when troubleshooting many issues.

Dumps of CA7ONL

To ensure that adequate areas of MVS storage are included in any dump from an abend that occurs in any CA WA CA 7 Edition component (or any address space dump that CA Support requests), include the following in the appropriate IEADM α 00 members in SYS1.PARMLIB:

```
SDATA=(CSA,LPA,LSQA,PSA,RGN,SQA,SUM,TRT,GRSQ)
```

With Version 12.0, 64-bit storage is allocated for many prime areas, such as active workload queues. As a result, the SYSUDMP is no longer sufficient for the problem resolution. The JCL procedure for CA7ONL now uses the ddname SYSMDUMP and routes the data to a generation data group. If CA7ONL is shut down cleanly (with RC = 0), the GDG is deleted. If CA7ONL issues an abend, the SYSMDUMP data set is retained for further examination.

Business Value:

Using the correct dump values can help solve issues quicker.