ISR Application

ISR – Image Services Replication

Installation and Configuration Guide

July 10, 2006
Version 2.1.7

Notices

This document contains information proprietary to FILENET Corporation (FILENET). You may not disclose or use any proprietary information or reproduce any part of this document without written permission from FILENET.

Even though FILENET has tested the hardware and software and reviewed the documentation, FILENET makes no warranty or representation, either express or implied, with respect to the hardware, software, or documentation, their quality, performance, merchantability, or fitness for a particular purpose. FILENET has made every effort to keep the information in this manual current and accurate as of the date of publication or revision. However, FILENET does not guarantee or imply that this document is error free or accurate with regard to any particular specification. As a result, this product is sold as is, and you the purchaser are assuming the entire risk as to its quality and performance.

In no event will FILENET be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect in the hardware, software, or documentation, even if advised of the possibility of such damages. In particular, FILENET shall have no liability for any programs or data stored in or used with FILENET products, including the costs of recovering such programs or data.

Some states do not allow the exclusion or limitations of liability for incidental or consequential damages, so the above limitation or exclusion may not apply to your installation. You may also have other rights that vary from state to state.

No FILENET agent, dealer, or employee is authorized to make any modification, extension, or addition to the above statements.

Table of Contents

1. INTRODUCTION & OVERVIEW 1

1.1 SINGLE DIRECTION DUPLICATION DIAGRAM 2

1.2 BI-DIRECTIONAL DUPLICATION DIAGRAM 4

2. DEFINITIONS 7

3. HARDWARE & SOFTWARE REQUIREMENTS 9

3.1 MULTIPLE PS UTILITIES INSTALLED (I.E. ISR & HPII, ISR & NLS) 9

4. PREREQUISITES 10

4.1 OPERATING SYSTEM DATE / TIME PROPERTIES 10

4.2 P8-IM SOFTWARE PATCH (MINIMUM) REQUIREMENTS 10

4.3 P8-IMSTK SOFTWARE PATCH (MINIMUM) REQUIREMENTS 10

4.4 P8-IM SERVER CONFIGURATION REQUIREMENTS 10

4.4.1 Add P8-IM services to “hosts” file 10

4.4.2 Edit PATH for user environment 11

4.4.3 Create “peer” system environment in IS 11

4.4.4 SysAdmin logins 12

4.4.5 NFS Mount point REQUIRED 12

4.5 INSTALL IMAGE MANAGER TOOLKIT (WAL) 12

4.6 INSTALL JAVA RUNTIME 12

4.7 OBTAIN THE LICENSE KEY (PSLSF) 13

5. SOFTWARE INSTALLATION 16

5.1 UNIX PLATFORM 16

5.1.1 Using the local installer (PSL_install) UNIX only 18

5.2 WINDOWS SERVER PLATFORM 20

6. CONFIGURING ISR 23

6.1 ISR.CFG – CREATION AND CONFIGURABLE ATTRIBUTES 23

6.2 CONFIGURING ENCRYPTED PASSWORD (OPTIONAL) 29

6.3 WQS QUEUE LIST (OPTIONAL) 31

6.4 ISR_CONFIG JAVA APPLICATION OVERVIEW 32

6.4.1 Using ISR_Config on Source P8-IM systems 34

6.4.2 Definition of P8-IM Activity Log file entries 41

6.4.3 Using ISR_Config on Destination and Source P8-IM systems .42

6.5 ACTIVITY LOGGING SHARE OR MOUNTED FILE SYSTEM 46

6.6 RUN CHECKSSN 47

6.7 USERS / ENVIRONMENT 48

6.8 CACHE PARAMETERS 49

6.9 CACHE ONLY SYSTEMS 49

7. P8-IM ACTIVITY LOGS 50

8. ISR PROGRAMS – FUNCTIONALITY 52

8.1 RUNTIME 52

8.2 8.2 PROGRAMS 53

8.3 REQUIRED SCRIPT ISR_FLUSHLOGS CUSTOMIZED FOR SITE 61

8.3.1 ISR_FlushLogs Customized for bi-directional filtering 62

8.4 OPTIONAL SCRIPT ISR_FINALPROC CUSTOMIZED FOR SITE 64

8.5 SECPASSWD - OPTIONAL SECURITY OBJECT UPDATE TOOL 64

8.6 BI-DIRECTIONAL CONTROLLER 66

9. STARTING AND STOPPING ISR 68

9.1 USING THE ISR_START AND ISR_STOP PROGRAMS 68

9.2 EXECUTING ISR_START AND ISR_STOP ON THE COMMAND LINE.....68

9.3 RUNNING ISR AS A WINDOWS SERVICE.....68

10. CONFIGURING ISR TO RUN 3RD PARTY (SINGLE-DIRECTION).....71

10.1 PURPOSE OF A 3RD PARTY ENVIRONMENT.....71

10.2 REQUIREMENTS IN A 3RD PARTY ENVIRONMENT.....71

10.3 CONFIGURATION IN A 3RD PARTY ENVIRONMENT.....71

10.3.1 Configuring 3rd Party System.....71

10.3.2 Configuring Source System(s) in a 3rd party environment.....72

10.3.3 Configuring Destination System in a 3rd party environment.....72

10.4 RUNTIME IN A 3RD PARTY ENVIRONMENT.....73

10.4.1 Startup in 3rd party environment.....73

10.4.2 Shutdown in 3rd party environment.....73

11. ISR REPORTING, VALIDATION AND MAINTENANCE.....75

11.1 ISR_REPORTS.....75

11.2 ISR_VIDAMIN.....75

11.2.1 Command line options.....76

11.2.2 Input File Contents.....77

11.2.3 Formulating a command line.....81

11.2.4 Output files and logs.....81

11.2.5 Processing Vidamin Activity Logs.....82

11.3 PENDING WORK.....82

11.4 ERROR DETECTION, RECOVERY AND CLEAN-UP.....83

11.4.1 WKRpak.....85

11.5 CLEANING ISR DATA AND WORKING FILES.....85

11.6 SYSTEM SYNC VALIDATION 87

12. FILTERING 90

12.1 DOCUMENT CREATES 90

13. TROUBLESHOOTING 91

13.1 PROCEDURE FOR REPORTING PROBLEMS 91

13.2 UTILITY / OPERATING SYSTEM AND WAL VERSION RELATIONSHIP 92

13.3 ERROR LOGGING FILES 93

13.4 MONITOR FAILS TO RUN 94

13.5 ISR_ENGINE IMPROPER SHUTDOWN 95

13.6 PROBLEMS RUNNING ISR_CONFIG 96

13.7 SOURCE LOG FILES NOT PICKED UP IN AN INTERVAL 98

13.8 FULL FILE SYSTEM LIMITATION 98

13.9 WHEN TO USE WAL_PURGE – UNIX ONLY 98

13.10 JOURNAL LOG ERROR ENTRIES 99

13.10.1 Folder Security Attributes 99

13.10.2 Document / Folder Filing 99

13.10.3 Folder Renaming *(bi-directional only)* 100

13.10.4 Missing SCRs for Annotation create support 100

13.11 PROCEDURE FOR CREATING/DELETING SECURITY OBJECTS 100

13.11.1 ISR Security Object mode set to DUP (single-direction) 101

13.11.2 ISR Security Object mode set to DUP (bi-directional) 101

13.12 MULTI SYSTEM COMMITTAL (MSC) 103

13.12.1 MSC Functional Limitation 103

13.12.2 MSC Document Commit problems on the Destination 103

13.13 CONTENT FEDERATION SERVICES & NON-CATALOGED DOCUMENTS 104

13.14 ERROR CODES 104

13.15 COMMON P8-IM ERROR CODES 107

14. APPENDIX A–EPHEMERAL PORT SETTINGS 108

14.1 DEFINITION 108

14.2 WINDOWS PLATFORM 109

14.3 *MAXUSERPORT* 109

14.4 *TCPMAXCONNECTTRANSMISSIONS* 110

14.5 *TCPMAXCONNECTRETRANSMISSIONS* 110

14.6 *TCPTIMEDWAITDELAY* 110

14.7 *FN_COR_QLEN* 111

14.8 UNIX PLATFORM 112

14.9 IBM AIX 112

14.10 HP-UX 113

14.11 SUN SOLARIS 114

1. INTRODUCTION & OVERVIEW

MSS is a Professional Services software solution that duplicates FILENET Image Manager activities between two FILENET domains. While MSS is the name of the complete implementation package, its core consists of a set of ISR executables, the subject of this documentation.

ISR software is FILENET installable ONLY. FILENET personnel will contact your site to schedule your implementation. If you have any questions regarding implementation, please direct them to your FILENET Sales Team. Please see the most recent product announcement for MSS (ISR) should you have any questions regarding requirements or specific software versions not answered in this document.

This document describes the installation, configuration and use of Single-direction and Bi-directional Duplication Architecture for Image Manager developed by FILENET Professional Services. This document is intended to be read by people who are familiar with the UNIX, Windows or the Windows 2000 operating systems as well as FILENET's Image Manager software. The ISR solution is dependant upon Image Manager Toolkit (WAL) and a license key to function. The license key should be obtained by your Professional Services Systems Consultant prior to arrival at your site.

Please read all sections prior to the FILENET Consultant's arrival. A FILENET Professional Services System Consultant will perform the synchronization procedures on all systems within the ISR NET, verify prerequisites are present, and assist the System Administrator with required space calculations before installing, configuring and running ISR.

ISR functionality is identical when running on a UNIX platform vs. a Windows platform. However, the method for installing ISR does differ on the two platforms. Please see the section on installation that is specific to your Operating System for details.

MSS is designed is to process user transaction loads in a specified interval. The amount of transactional data that can be processed in an interval is constrained by the systems available memory. Automated/programmatic data transactions may provide more data for processing than available memory will allow. These transactions may need to be tailored to submit a smaller amount of data during an interval.

1.1 Single Direction Duplication Diagram

Your FileNet Image Manager Domain will begin Activity logging on the Source System(s) as designated in the IM act_log_conf file. Please see section on IM Activity Logging for more details on this portion of the architecture.

1. Source Monitor starts RPC Server

2. Destination Monitor starts RPC Server

   2-a. RCP communication at initialization

3. Destination Monitor starts ISR_Engine

4. ISR_Engine calls Source ISR_Server to execute ISR_FlushLogs

   4-a. ISR_Engine will move Activity Logs from Source to Destination

5. ISR_Engine calls ISR_Builder to convert logs to (APP) packets

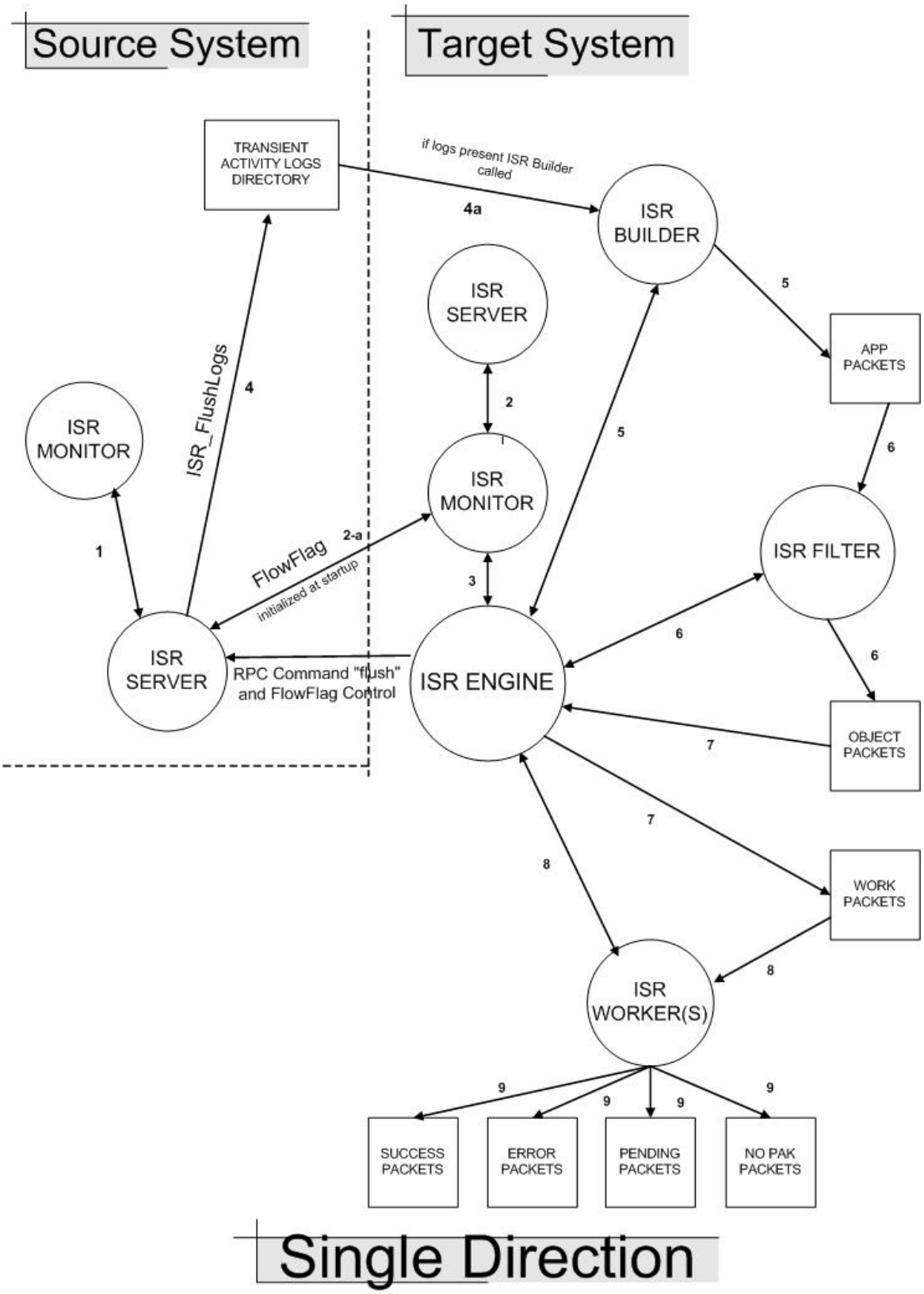
6. ISR_Engine calls ISR_Filter to filter APP packets and create OBJ packets

7. ISR_Engine collects OBJ packets and creates WKR packets

8. ISR_Engine calls ISR_Worker(s) to execute WKR packets

9. ISR_Worker processes binary WKR packets and distributes them to Success, Error or Pending depending on the ability to process the Activity in the currently configured environment. Additionally, a NOpak packet is created as a result of WKRpak (correctly) processing an IGNORE. (See sections on ERRpak and Pending for more details on specific behavior.)

See diagram next page


Source System
Target System
TRANSIENT ACTIVITY LOGS DIRECTORY
if logs present ISR Builder called
4a
ISR BUILDER
ISR SERVER
5
APP PACKETS
ISR_FlushLogs
4
2
ISR MONITOR
5
6
ISR MONITOR
ISR FILTER
1
FlowFlag
2-a
initialized at startup
3
6
6
ISR SERVER
RPC Command "flush" and FlowFlag Control
ISR ENGINE
OBJECT PACKETS
7
7
8
WORK PACKETS
8
ISR WORKER(S)
9
9
9
9
SUCCESS PACKETS
ERROR PACKETS
PENDING PACKETS
NO PAK PACKETS
Single Direction

1.2 Bi-Directional Duplication Diagram

Your FileNet Image Manager Domain will begin Activity logging on the Physical Source and Physical Destination System(s) as designated in the IM act_log_conf file. Please see section on IM Activity Logging for more details on this portion of the architecture.

1. Physical Source ISR_Monitor starts Physical Source RPC ISR_Server

2. Physical Destination ISR_Monitor starts Physical Destination RPC ISR_Server

   2-a. RPC communication at initialization
   2-b. RPC communication at initialization

3. Physical Destination ISR_Monitor starts Physical Destination ISR_Engine

4. Physical Source ISR_Monitor starts Physical Source ISR_Engine

5. Physical Destination ISR_Engine blocks Physical Source ISR_Engine

6. Physical Destination ISR_Engine calls Physical Source ISR_Server to execute ISR_FlushLogs

   6-a. ISR_Engine will move Activity Logs from Physical Source to Physical Destination

7. Physical Destination ISR_Engine calls Physical Destination ISR_Builder to convert logs to (APP) packets

8. Physical Destination ISR_Engine calls Physical Destination ISR_Filter to filter APP packets and create OBJ packets

9. Physical Destination ISR_Engine collects OBJ packets and creates WKR packets

10. Physical Destination ISR_Engine calls Physical Destination ISR_Worker(s) to execute WKR packets

11. ISR_Worker processes binary WKR packets and distributes them to Success, Error or Pending depending on the ability to process the Activity in the currently configured environment. Additionally, a NOpak packet is created as a result of WKRpak (correctly) processing an IGNORE.

12. Physical Destination ISR_Engine unblocks Physical Source ISR_Engine

13. Physical Source ISR_Engine blocks Physical Destination ISR_Engine

14. Physical Source ISR_Engine calls Physical Destination ISR_Server to execute ISR_FlushLogs

    14-a. ISR_Engine will move Activity Logs from Physical Source to Physical Destination

15. Physical Source ISR_Engine calls Physical Source ISR_Builder to convert logs to (APP) packets

16. Physical Source ISR_Engine calls Physical Source ISR_Filter to filter APP packets and create OBJ packets

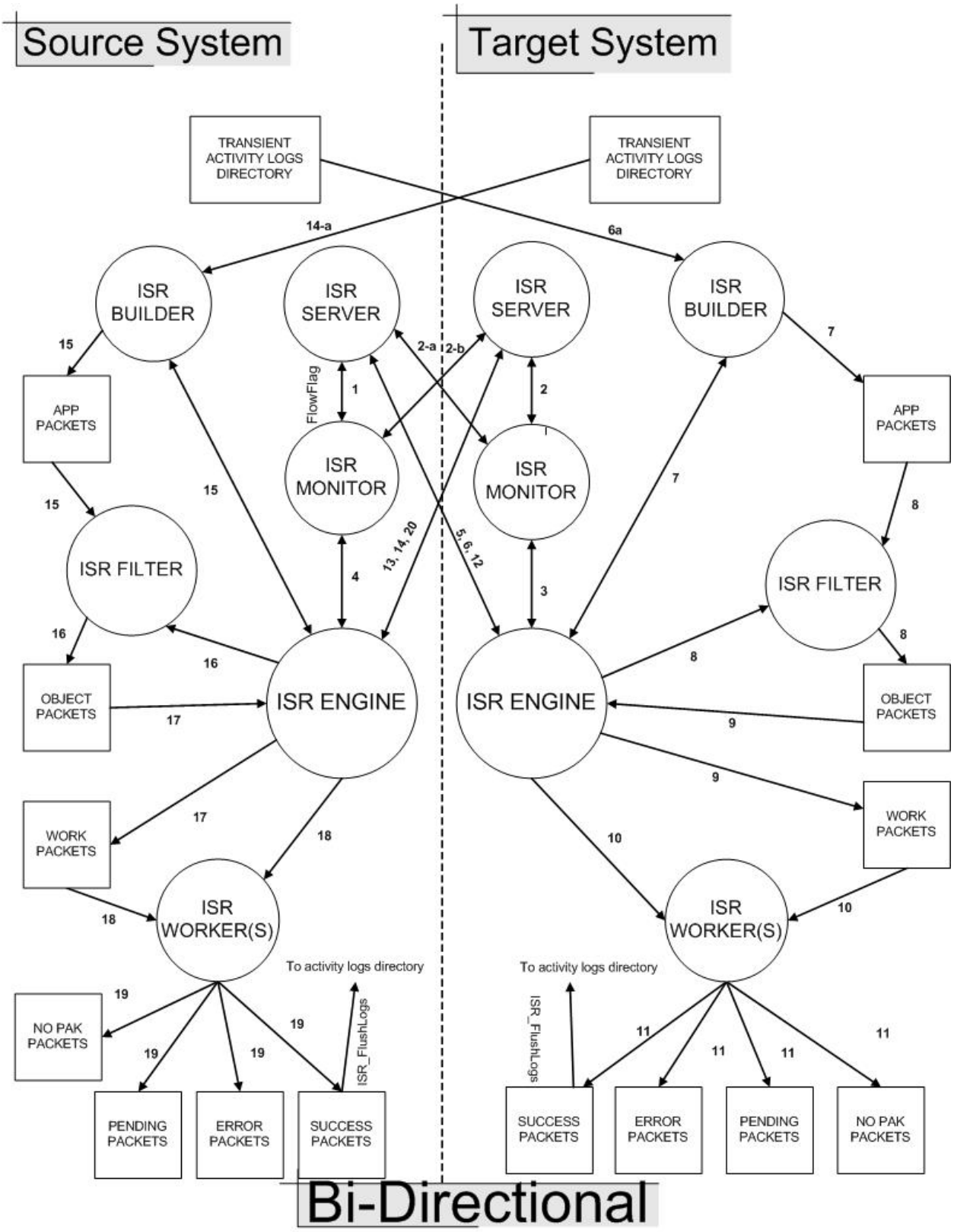
17. Physical Source ISR_Engine collects OBJ packets and creates WKR packets

18. Physical Source ISR_Engine calls Physical Source ISR_Worker(s) to execute WKR packets

19. ISR_Worker processes binary WKR packets and distributes them to Success, Error or Pending depending on the ability to process the Activity in the currently configured environment. Additionally, a NOpak packet is created as a result of WKRpak (correctly) processing an IGNORE. (See sections on ERRpak and Pending for more details on specific behavior.)

20. Physical Source ISR_Engine unblocks Physical Destination ISR_Engine

21. Return to step #5 for start of next interval (See diagram next page)


Source System
Target System
TRANSIENT ACTIVITY LOGS DIRECTORY
TRANSIENT ACTIVITY LOGS DIRECTORY
14-a
6a
ISR BUILDER
ISR SERVER
ISR SERVER
ISR BUILDER
15
2-a
2-b
7
FlowFlag
1
2
APP PACKETS
APP PACKETS
ISR MONITOR
ISR MONITOR
15
15
7
8
13, 14, 20
5, 6, 12
ISR FILTER
ISR FILTER
4
3
16
16
8
8
OBJECT PACKETS
ISR ENGINE
ISR ENGINE
OBJECT PACKETS
17
9
9
WORK PACKETS
WORK PACKETS
17
18
10
18
ISR WORKER(S)
ISR WORKER(S)
10
To activity logs directory
To activity logs directory
19
NO PAK PACKETS
19
19
19
ISR_FlushLogs
ISR_FlushLogs
11
11
11
11
PENDING PACKETS
ERROR PACKETS
SUCCESS PACKETS
SUCCESS PACKETS
ERROR PACKETS
PENDING PACKETS
NO PAK PACKETS
Bi-Directional

2. DEFINITIONS

ISR – Multiple System Synchronization is the FILENET Professional Services offering that combines the procedures, processes and ISR software implementation for a Disaster Recovery dynamic backup IM system.

ISR NET – The collection of all domains associated within the ISR Duplication environment.

MSC – Multi System Committal: See IM documentation for information on MSC.

Source System – All domains within the ISR NET that have IM Activity Logging turned on in an effort to duplicate or replicate all configured Activity Logging to a Destination domain. In Bi-directional mode there is a Physical Source and a Virtual Source. See definitions for each.

Destination System – The domain within the ISR NET that will pull all Logging files from the Source System(s) in an effort to duplicated or replicate configured Activity Logging. In Bi-directional mode there is a Physical Destination and a Virtual Destination. See definitions for each.

Physical Source (bi-directional mode) – The system where the first entry in the ISR_Config user interface contains this domain name. This system's Event logs will be the first to be duplicated when ISR is started. This processing would be considered the first half of an interval.

Physical Destination (bi-directional mode) – The system where the second entry in the ISR_Config user interface contains this domain name. This system's Event logs will be duplicated second. This processing would be considered the second half of an interval.

Virtual Source (bi-directional mode) – The system whose Activity logs are actively being processed during half of an interval. This system's Engine is not in control of the FlowFlag directional controller.

Virtual Destination (bi-directional mode) – The system that is processing Activity logs acquired from the Virtual Source during half of an interval. This system's Engine is in control of the FlowFlag directional controller.

FlowFlag – The bi-directional controller that locks duplication for one domain over another inside an interval.

Domain – The name of the FILENET Image Service.

IM – FILENET Image Manager

Activity Logging – Transactions (events) logged by IM(s) where the IM is a Source domain in the ISR NET when running in single direction mode. When in Bi-directional mode both IM servers in the ISR NET will have Activity Logging turned on.

WAL – Image Manager Toolkit (WorkFlo Application Library)

Single-direction Duplication Mode – Running ISR between one or more Source machines in a single directional path to only one Destination machine which results in configured Source machine(s) actions being duplicated on the Destination system.

Bi-directional Duplication Mode – Running ISR between 2 IM domains in a bi-directional mode, which results in configured IM logged actions being replicated on both systems. (ISR processing will occur on both systems during one interval.)

Interval – This setting is the time it takes for ISR to process all available Activity Logs. In a Single-Direction environment the interval consists of the Destination pulling the available Activity Logs from the Source to it's working area, processing the logs and performing the WAL calls to duplicate the Events. In a Bi-Directional environment the interval consists, first, of the Physical Destination pulling and processing the available Activity Logs from the Physical Destination. Then the Physical Source (which becomes the Virtual Destination) will pull and process the available Activity Logs from the Physical Destination (which becomes the Virtual Source).

3. HARDWARE & SOFTWARE REQUIREMENTS

ISR 2.1.x provides version releases compiled for WAL 3.6 and WAL 4.0 along with all the Operating Systems currently support by IM 3.6 and IM 4.0. When installing on a UNIX Operating System the ISR installation program will determine the WAL version release and install the ISR version that coincides with the WAL version found on the server. Windows Server installs will require the user to know the WAL version level and select the correct version during the installation process. (See the section on Installation for more information on how to detect your WAL version level.)

Any other requirements are dependent upon the WAL and IM supported hardware and software requirements. ISR 2.1.x is not supported on any FILENET Image Manager Release before version 3.6.10/SP2 (plus required *SS SCRs). Specific required software versions and fixes are outlined in the Prerequisites section of this document. IM and WAL must already be installed, configured and running before installing ISR. Please see IM and WAL documentation for those requirements.

The ISR software installation will require a minimum of 10 MB of hard disk space. Runtime or Working Directory space usage will be dependent upon size of user files, volume, and other factors such as file retention schedules.

3.1 Multiple PS Utilities Installed (i.e. ISR & HPII, ISR & NLS)

Guidelines for installing/upgrading if multiple instances of an Image Manager Utility are installed on a system:

The main issue is the PS Shared Library versions need to be compatible.

If you are installing more than one product (HPII, MRII, DART, NLS, Cache Maint) install the newest version product last. This will ensure you have the latest version of the shared library installed.

If the PS library release for the utilities differs by the third digit (i.e. 2.4.1 & 2.4.2), then they are compatible as long as the highest level version of the library is the one being used.

If the PS Library release differs by the second digit (i.e. 2.3.0 & 2.4.0), the utilities will not be compatible. If this is the case you are required to install the latest release for all products.

4. PREREQUISITES

4.1 Operating System Date / Time Properties

All ISR domain Servers within the ISR NET must have their Date / Time Clocks synchronized. This is important as ISR uses the Date/Time stamp on Events handed to it by the IM Activity Logging to process Action items on the Destination system.

Should the Destination system's clock move ahead of the Source system(s) and an interval is activated exactly on the hour you may see logs, produced during that interval, remain on the Source system(s) and not collected for processing until the next interval. It is recommended that the time clocks be within the same minute or not more than 59 seconds apart.

Different Time Zones are supported in the ISR functionality and there are no specific configuration procedures necessary for this environment.

4.2 P8-IM software patch (minimum) requirements

All Operating System, Database and Service Pack levels must follow the P8-IM version guidelines before installing ISR. These requirements must be installed before running ISR 2.1.x. (Minimum requirements should always be checked on the CSS Website for whatever is current at the time of the installation.)

- P8-IM 3.6.10 SP2
- P8-IM 3.6.30 SP2 + June 2003 Tech Info CD
- P8-IM 4.0.0

4.3 P8-IMSTK software patch (minimum) requirements

The latest Image Manager Tool Kit (WAL) Service Pack is recommended.

4.4 P8-IM server configuration requirements

4.4.1 Add P8-IM services to "hosts" file

Verify your "hosts" file contains the correct P8-IM services values for all peer ISR domains in the ISR Net:

- *Tcpip systemname domainname domainname-filenet-nch-server*

Example:

- 192.168.10.2 psverve verve verve-filenet-nch-server

IMPORTANT NOTE: The *domainname* of your P8-IM Service is case sensitive in relationship to ISR runtime files. Specifically, the domain=*domainname* listing in the ISR.cfg must match with the *domainname* entry in the hosts file. To ensure your hosts file keeps in sync with your ISR configuration files, keep your hosts nch server *domainname* entry consistent with the character case entered as your P8-IM configured domain name and your ISR.cfg domain name.

4.4.2 Edit PATH for user environment

The PATH in the user environment must have the WAL shared library directory, but not the P8-IM shared library directory. See the WAL install readme file. It is covered in this document to emphasize the importance of setting the PATH correctly. The UNIX PATH differs from its the Windows Server counterpart. It is important not to confuse the Windows configuration with the UNIX environment configuration. The PATH should not be limited to the examples below, as there are other software and OS requirements found in each environment PATH that must be retained. The examples below are to only show the WAL vs P8-IM PATH hierarchy.

- To configure the PATH in UNIX:

  cd /home/fnsw
  vi .profile (Bourne or Korn shell)
  vi .login (C Shell)

- Change the PATH segment statement placing /fnsw/client/bin:/fnsw/local/bin in front of /fnsw/bin

  /fnsw/client/bin:/fnsw/local/bin:/fnsw/bin

- The PATH segment in Windows when appended should look like the following:

  <drive>:\fnsw\client\bin;<drive>:\fnsw\client\shobj;<drive>:\fnsw\bin;<drive>:\fnsw\lib\shobj;

4.4.3 Create “peer” system environment in IS

Make the systems in ISR NET “Peer” systems through the P8-IM fn_edit utility.

- Run fn_edit and Select the “Add a peer system” procedure on the main screen. Once completed, go to the Peer Systems tab and verify the peer system is present and all its properties are correct.

- Setup unique Document ID ranges and Surface ID ranges for each system in the ISR Net, just as one would normally do for peer systems.

4.4.4 SysAdmin logins

Add a minimum of 10 more logins to the SysAdmin Image Manager user.

- ISR needs the SysAdmin user to login to P8-IM domains in the ISR NET to process work. It will be necessary to add 10 to the current amount of "concurrent logins" your site currently has configured for the SysAdmin user. (Set this minimum a bit higher than 10 if 8 workers are configured.)

4.4.5 NFS Mount point REQUIRED

A UNIX NFS Mount or Windows Share is required for Activity logs accessibility.

- The P8-IM Activity Logging runs in such a way that an P8-IM "flush" function must be performed at the beginning of an interval and upon completion of this "file close" all logs will be pulled to the Destination system for processing. This process is "time" critical thus mandating the mounted or shared file system access.

4.5 Install Image Manager Toolkit (WAL)

Install Image Manager Toolkit (WAL) CD for your Operating System.

- Follow the instructions on the installation CD. Be sure to follow all configuration directions.
- Apply any WAL patches necessary for your site's environment.

IMPORTANT NOTE: WAL and P8-IM patches must be kept in sync. Whenever an P8-IM Service Pack is installed the accompanying WAL Service Pack must be installed.

4.6 Install Java Runtime

If not already present on your system, Install Java Runtime on all root servers in domains that are within the ISR NET. The ISR_Config Java application uses the "jre," a java executable, that is included in the Java Runtime software.

- The ISR installation CD contains the installation executables and instructions provided by the vendor for all supported environments. Paths to the install as well as the download web sites for your environment can be found below. (When installing in a UNIX environment the default install locations can be /usr/local/bin or /opt/java/bin/jre.)
    - AIX: /JRE/AIX/{jre130.tar.gz, License.txt, README.JRE130}

http://www-106.ibm.com/developerworks/java/jdk/index.html

- HP11: /JRE/HP-11/{jre11805_os11.depot}

http://www.hp.com/products1/unix/java/java1/jdk_jre/downloads/index.html

- HP10.20: /JRE/HP-UX/{jre11805_os10.depot}

http://www.hp.com/products1/unix/java/java1/jdk_jre/downloads/index.html

- Solaris: /JRE/SunOS/{j2re-1_1_8_14-solsparc.bin, sun_jre_license.txt}

http://java.sun.com/products/jdk/1.1/jre/index.html

- WINDOWS: /JRE/i386/{j2re-1_3_1-win.exe, win_jre13_license.txt, win_jre13_readme.txt}

http://java.sun.com/products/jdk/1.1/jre/index.html

4.7 Obtain the License Key (PSLSF)

Obtain the License key file from Professional Services (PSLSF)

- The ISR solution will require a license key containing the specific FILENET Image Server Domain SSN numbers for each system in the ISR NET. This file will be named "PSLSF" and should be obtained by your Professional Services Systems Consultant prior to his arrival at your site. The LicenseForm.doc file can be found on the installation CD or on the CSS download site. This form can be submitted to pslicense@filenet.com and is normally processed within a 24-hour period.
- Once ISR is installed this license key (PSLSF) will be placed in all the install directories using a binary format.
- A sample of the License key file request form is included below.

Important!

Read before Installation

FileNET®
Unleash the Power of Information™

The software product you are about to install requires a License Key File in order to execute. The License Key File is a unique file generated from your System Serial Number and is NOT included on the product CD.

- If you are performing a software upgrade only, you already have a License Key File and do not need to obtain a new one. You may disregard this notice.
- If you are adding a new Professional Services product and have a License Key File for other Professional Services products, you must still obtain a new License Key File.
- If you are requesting a temporary use License Key File (FileNET only), it will expire in 45 days and a permanent License Key File must be ordered.

To obtain a License Key File, email ALL of the following information to the mailbox listed below or fill out the electronic form on your product CD called LicenseForm.doc and email it. An electronic License Key File will be emailed back to the reply address with installation instructions.

Email: pslicense@filenet.com

Please make sure that all of the information is supplied to obtain a License Key File.

Company Name:

Address:

Contact Name:

Phone Number:

System Serial Number:

(10 digit number located on enclosed Software License document or fill in 'Temporary Use' and the requestor's name)

Model Number:

*(6 digit number with * located on enclosed Software License document)*

Sample PSLSF License request form

5. SOFTWARE INSTALLATION

5.1 UNIX Platform

This section describes the installation procedure for UNIX systems **only**. (Refer to the next section "Windows Server Platform" for installation instructions when installing ISR on a Windows Server Operating System.)

The following procedure installs ISR into the directory **/fnsw/local/bin**.

1. On the server that you will be installing ISR on, log in as the **root** user.
2. Insert the CD into the CD-ROM drive.
3. Mount the CD if your environment does not auto mount the CD.
4. Change directory to the CD-ROM mount point. This should place the user at the root of the CD where the "ps_install" executable is located.

NOTE: HP systems require a specific flag in the mount command to read the ISO9660 format of the ISR CD. Please use the following flags with the mount command.

mount –o cdcase <device> <mountpoint>

5. Run the installation program for ISR using the following command.

NOTE: The "csh" shell specification in the command is needed if the user is not using the csh shell. If the user is already using the csh shell, including "csh" in the command will not cause any problems for the install.

>csh ./ps_install

NOTE: A period is needed before and sometimes after the ps_install command.

IMPORTANT NOTE: There can be more than one csh (c shell) available on a UNIX system. The use of the csh (c shell) when executing the ps_install program can fail depending upon the default csh. Should there be a failure during installation it may become necessary to issue the ps_install command without specifying a shell. Another alternative would be to issue the command specifying the path to an alternate csh on the system.

6. Verify that the following files were written to the install directory (/fnsw/local/bin):

AIXOS	HP	Sun
CheckSSN	CheckSSN	CheckSSN
ISR_2_1_7	ISR_2_1_7	ISR_2_1_7
ISR_Builder	ISR_Builder	ISR_Builder
ISR_cfg.help	ISR_cfg.help	ISR_cfg.help

ISR_cfg.sample	ISR_cfg.sample	ISR_cfg.sample
ISR_cfg.txt	ISR_cfg.txt	ISR_cfg.txt
ISR_Client	ISR_Client	ISR_Client
ISR_Config	ISR_Config	ISR_Config
ISR_Engine	ISR_Engine	ISR_Engine
ISR_Filter	ISR_Filter	ISR_Filter
ISR_FinalProc.sample	ISR_FinalProc.sample	ISR_FinalProc.sample
ISR_FlushLogs.sample	ISR_FlushLogs.sample	ISR_FlushLogs.sample
ISR_java.jar	ISR_java.jar	ISR_java.jar
ISR_Monitor	ISR_Monitor	ISR_Monitor
ISR_net_sample.csv	ISR_net_sample.csv	ISR_net_sample.csv
ISR_Report	ISR_Report	ISR_Report
ISR_rpt.help	ISR_rpt.help	ISR_rpt.help
ISR_rpt.txt	ISR_rpt.txt	ISR_rpt.txt
ISR_Server	ISR_Server	ISR_Server
ISR_Sleep	ISR_Sleep	ISR_Sleep
ISR_start	ISR_start	ISR_start
ISR_stop	ISR_stop	ISR_stop
ISR_sums	ISR_sums	ISR_sums
ISR_TServer	ISR_TServer	ISR_TServer
ISR_Vidamin	ISR_Vidamin	ISR_Vidamin
ISR_Worker	ISR_Worker	ISR_Worker
NLSc	libNLSc.sl	libNLSc.so
PSs	libPSs.sl	libPSs.so
PSs_2_5_1	PSs_2_5_0	PSs_2_5_0
PS_Password	PS_Password	PS_Password
SecPasswd	SecPasswd	SecPasswd
timer.awk	timer.awk	timer.awk

DO NOT DELETE THE FOLLOWING FILES FROM A PREVIOUSLY INSTALLED SYSTEM IF YOU ARE UPGRADING ISR AS THESE FILES ARE WORKING CONFIGURATION FILES.

PSLSF
ISR.cfg
ISR_net.csv
ISR_FlushLogs

ONLY if implemented in your environment:
.password (if you are using encrypted password functionality)
ISR_FinalProc

IMPORTANT NOTE: **ISR_cfg.sample** is a sample configuration file. After it is modified to run in your environment, it must be copied to or renamed **ISR.cfg** to be used as the runtime configuration file. **ISR_net_sample.csv** is a sample configuration file. The **ISR_net.csv** file is made using **the ISR_Config** program. It is not necessary to hand edit the **ISR_net_sample.csv** file for this configuration item. Please see instructions under the section entitled "Configuring ISR."

NOTE: Following installation, ISR can be moved to another directory, if desired, by simply moving all of the above files from /fnsw/local/bin and changing the path environment variable to the new location.

5.1.1 Using the local installer (PSL_install) UNIX only

Updated software versions can be downloaded from the CSS Download web site. There are detailed instructions in the directory entitled "Documents", which is found in the Professional Services Utilities area.

Installing the Utilities on a UNIX operating system when the *.tar.Z file is downloaded off this CSS download Web Site.

To ensure that you are installing the correct build for your environment please stamp the following modules:
WAL module -> /fnsw/client/shobj/*SysV*
The "system" stamp for WAL must match the first two major numbers (4.0.xx.xx).
The example below is for WAL 4.0* builds (system 4.0.20.73)

The "Rel_type" must match. The example is for Solaris (Rel_type: rel_solaris)

```
hqpsleppard(fnsw)/fnsw/lib/shobj> stamp libSysV.so
   system 4.0.20.73(0) (userid 8, Thu Nov 18 20:08:47 2004)
   developer 4.0.20.0.5 (userid 8, Thu Nov 18 20:08:45 2004)
   SubSys: mv, Rel_type: rel_solaris, SCR#: 212145, mode: 100755, size: 1385940
```

The following steps must all be done by a **root** user.

FIRST:
Install the Professional Services utility from the CSS download site using the local installer:

1. Download the *.tar.Z file for the OS and WAL compile that matches the configuration of the system you will be installing onto.
2. Download the PSL_install program
3. Place both files in a local directory on the system to be installed (/tmp)

4. Set the permissions on the installer:
   chmod 777 PSL_install
5. Make sure the permissions and character case of the *_tar.Z file are correct:
   chmod 666 *_tar.Z
   make sure the tar file case is as follows:
   "HPII, DART, or MRII" is in uppercase
   "tar" is in lower case
   "Z" is in uppercase
6. Enter the install command with or without "./" infront of the program. Some environments require the "./" preceding the install program.
   ./PSL_install HPII
   or
   PSL_install HPII

Using the PSL_install program will ensure the utility files ownership and file permissions are set correctly. This installer will also append the utility file ownership and permissions to the P8-IM set permissions file. Should the P8-IM be
reinstalled or updated via a Service Pack the set permissions file will be replaced with a new file and all utility permission settings in the file will be lost. Reinstalling the utility will restore the utility information to the set permissions file.

IMPORTANT NOTE: During the install the PSL_install executable will display to the screen verbose information of each process completion. The copy process of each file will be displayed. When installed as an upgrade, over an existing version the installer will report an error should the copy of a new file over an existing file fail. However, the installer will continue and complete the install. The only way you will know that the install was incomplete would be to review this display output. Should this happen correct the problem and rerun the install.

SECOND:
All WAL libraries should be fnsw:fnusr. To fix this and keep the correct user on wal_purge and wal_daemon you can do the following steps.
1. login as root
2. cd to /fnsw
3. chown -R fnsw:fnusr client
4. cd to /fnsw/client/bin
5. chown root wal_purge wal_daemon
6. chmod 4755 wal_purge wal_daemon

THIRD:
The Path must be set correctly for the user that is running HPII/MRII/DART. If it is the fnsw user then do the following:
1. cd /home/fnsw
2. vi .profile
   Fix the first part of the path to look like this:

/fnsw/client/bin:/fnsw/bin
The client path must come before the IMS path.

3. Check to see if the /fnsw/lib/shobj path is exposed by entering the command:
   set > junk.txt
   Search through the junk.txt file to see if the string /fnsw/lib/shobj is listed.

Find out why they have the shared library path exposed.
The only reason it might be is because eProcess is on the box.
If so PS utility personnel would like talk with you immediately as this is an intermittent problem we have found at some sites where the eProcess requirement is conflicting with a PS Utility because of a bad WAL interaction.

FOURTH:
1. Check to see if the /fnsw/client/bin/wal_gti is present.
2. The IMS MUST be recycled using an:
   initfnsw terminate
   initfnsw start

5.2 Windows Server Platform

This section describes the installation procedure for Windows Server systems **ONLY**.

1. On the server that you will be installing ISR, log in as a user with Administrator privileges.
2. Insert the CD into the CD-ROM drive and installation will begin via the Autorun program found in the root of the CD. If Autorun is turned off in your Windows Operating System environment use Explorer to access the root of the CD and then start the install by Double clicking on PS_Install.exe or by selecting Start, Run on the Start Menu and input the following command:

   <CD ROM drive letter>: PS_Install.exe
3. At the Application Package Installer dialog select the ***Continue*** button
4. At the software installation screen select the WAL version installed on your system from the ***Available Releases*** frame and then select the ***Install*** button.

   - To check your WAL version level on a Windows Server you can stamp the following module. The stamp information will look similar to the output below.

Windows Server DOS prompt:

```
stamp <drive>:\fnsw\client\shobj\*SysV*
```

Example of a WAL 4.0.5 module stamp:

\fnsw\client\shobj\wal_sysv.dll (WINDOWS bin):
system 4.0.5.7(0) (lib, Wed Oct 15 08:38:16 2003)
developer 4.0.0.0.11 (lib, Wed Oct 15 08:38:14 2003)

SubSys: mv, Rel_type: wal_nt, SCR#: 184822, mode: 100666, size: 447160

5. Verify the Home directory

IMPORTANT NOTE: The initial Home directory for ISR is: **C:\ISR.** This Home directory is usually changed to a folder path in the same general location as the P8-IM software and WAL software are normally installed. To specify this preferred area for the install location, select the ***No*** button.

6. If you selected the ***No*** button in the above step the ***Destination Path Selection*** dialog will open and you will be asked to specify/select a new install location. **Select create** the new install location and then select the ***Accept*** button

   A standard installation point for ISR is:

   < drive letter>: \fnsw_loc\bin

DO NOT SELECT AN INSTALLATION PATH LIKE "\Program Files". The space in the path name will not work with ISR.

7. Verify the new Home directory by selecting the ***Yes*** button, or to return to the ***Destination Path Selection*** dialog select the ***No*** button

8. The ***Copying Files*** dialog will display the installation progress and when installation has completed select the ***OK*** button

IMPORTANT NOTE: If this is an upgrade the Copying Files might experience an invalid error about failure to copy or delete existing files. To ensure the copy files completes correctly you may delete the following files from the previous install before performing the upgrade.

CheckSSN.exe
ISR_2_1_7
ISR_Builder.exe
ISR_cfg.help
ISR_cfg.sample
ISR_cfg.txt
ISR_Config.exe
ISR_Engine.exe
ISR_Filter.exe
ISR_FinalProc.sampl
ISR_FlushLogs.sampl
ISR_java.jar
ISR_Monitor.exe
ISR_net_sample.csv
ISR_net_sample.xls
ISR_Report.exe
ISR_rpt.help
ISR_rpt.txt

ISR_Sleep.exe
ISR_start.exe
ISR_stop.exe
ISR_TWinServer.exe
ISR_Vidamin.exe
ISR_WinClient.exe
ISR_WinServer.exe
ISR_Worker.exe
NLSc.dll
PSs.dll
PSs_2_5_0
PS_Password.exe
PS_util.exe
SecPasswd.exe
timer.awk

DO NOT DELETE THE FOLLOWING FILES FROM A RUNNING SYSTEM IF YOU ARE UPGRADING ISR AS THESE FILES ARE WORKING CONFIGURATION FILES.

PSLSF
ISR.cfg
ISR_net.csv
ISR_FlushLogs.bat (if you have customized it for you environment)
.password (if you are using encrypted password functionality)
ISR_FinalProc.bat

9. Close the Installation Program interface by selecting the ***Quit*** button

IMPORTANT NOTE: **ISR_cfg.sample** is a sample configuration file. After it is modified to run in your environment, it must be copied to or renamed **ISR.cfg** to be used as the runtime configuration file. **ISR_net_sample.csv** is a sample configuration file. The **ISR_net.csv** file is made using **the ISR_Config** program. It is not necessary to hand edit the **ISR_net_sample.csv** file for this configuration item. Please see instructions under the section entitled Configuring ISR.

NOTE: Following installation, ISR can be moved to another directory, if desired, by simply moving all of the above files from <drive>:\fnsw_loc\bin and changing the path environment variable to the new location.

6. CONFIGURING ISR

Once the software is installed, the following steps will configure each system within the **ISR NET**. The completed ISR configuration will utilize the **ISR.cfg** and **ISR_FlushLogs** ascii files that are hand-edited. The **ISR_net.csv** file is the runtime version of the **ISR_net_sample.csv** created by the **ISR_Config** Java application. The Source system(s) P8-IM Activity Logging will also use the **ISR_Config** Java application to setup the required **act_log_conf** file necessary for P8-IM Activity Logging.

There are two optional configuration files that can be created. One is an encrypted password file that supercedes the readable password in ISR.cfg, and the other is a **WQS list file** determines the duplication of WQS queue objects. Instructions for both these optional features can be found in sections below.

Each section of this chapter will first direct configuration in a single-direction mode. Where required, instructions stipulating bi-directional mode will follow.

6.1 ISR.cfg – Creation and Configurable Attributes

The **ISR_cfg.sample** is supplied as a template for the required **ISR.cfg** configuration file. This file, along with its configurable attributes will be used to provide runtime information to the ISR programs. Copy the **ISR_cfg.sample** to the filename **ISR.cfg.** This filename is case sensitive. **ISR.cfg** is a text file and can be modified using a text editor. Basic steps to edit this file are the same for both single-direction and bi-directional mode. However, in bi-directional mode, it will be important to edit a few extra lines to support bi-directional functionality. These lines are designated by comments stipulating the need for their use in bi-directional mode.

There are two types of configurable attributes in the **ISR.cfg** file. Mandatory attributes must be input during configuration time or ISR will not run. The balance of configurable attributes is optional and contains default values that can be individually set for value, rate or performance adjustments.

Create the ISR.cfg file on each system within the ISR NET. Edit this file to setup mandatory and optional settings necessary for ISR during runtime.

<u>Mandatory attributes</u>

LogonAttribute:

- **UserName** – This is the P8-IM Security User Name. Enter the **UserName** to be used to login to the local P8-IM domain. The user must be the P8-IM SysAdmin user. (Must be set or ISR will not start.)

- **PassWord** – This is the P8-IM Security Password. Enter the **PassWord** to be used to login to the local P8-IM domain. (Must be set or ISR will not start.)

If Password Security is a high priority you may choose to use the encrypted password functionality. To utilize this feature set the **UserName** to "**ISR**" and remove or comment out the **PassWord** parameter keyword. See the section on "Configuring Encrypted Password" for instructions on activating this functionality.

IMPORTANT NOTE: The ISR user MUST have at least 10 (ten) concurrent logins configured/available through the P8-IM Security Administration. (Set this minimum a bit higher than 10 if 8 workers are configured.)

- **Domain** – This is the P8-IM Domain Name. Set the Domain to the local system domain – The **ISR.cfg** file on the Source system will have the Source domain name entered for the Domain value. The **ISR.cfg** file on the Destination system will have the Destination domain name entered as the Domain value. (Must be set or ISR will not start.)

  Bi-directional mode uses the same domain name entries as single direction. The Physical Source (which is the first entry in the ISR_Config display) will have the Domain entry set to its own domain name. The Physical Destination will have the Domain entry set to its own domain name.

- **Organization** – This is the P8-IM Organization Name. It is the second part of the 2 part naming convention where the first part is the Domain name. ONLY enter the second part (organization) of the two-part name in the Organization field. (Must be set or ISR will not start.)

  Domain:Organization

PgmAttribute:

- **WorkingDirectory** – This is the directory under which all ISR directory structures are created. All working binary packet and ASCII log files will be located under this directory structure. The value for this parameter must be a fully qualified path. In a Windows environment this path may consist of a full path using the <drive>: letter or full UNC path. This Working Directory is the path the Destination system will use when it moves the P8-IM Activity logs from the Source system(s). This is also the area where all processing of the P8-IM Activity log files occurs. On a Destination system this will be the selected location containing enough space to house all ISR packet and log files used during the processing of the P8-IM Activity log files from the Source system(s). On a Source system this **WorkingDirectory** is not an active directory and must simply be set to a valid folder or path anywhere on the system. (Must be set or ISR will not start.)

  Bi-directional mode – The Physical Source as well as the Physical Destination must have a WorkingDirectory entered in this attribute. The path value entered can be the same as ISR will append each peer system's SSN to the WorkingDirectory path before creating the balance of the WorkingDirectory structure. This will create two separate structures for ISR work to be processed.

Optional attributes: The balance of configurable items found under **PgmAttribute:** in the ISR.cfg file are optional. These optional attributes can all run using the default values and do not need to be edited unless your environment requires specific tuning as designated by your FILENET System Consultant.

- **DefPass** – allows for the user to setup a default password for all new users duplicated or replicated through ISR. The default for this field is to set no password on the Destination system. Uncomment out the field and enter a string inside the double quotes will cause the User to be assigned the value of the DefPass field. The P8-IM Security Object model does not allow Source passwords to be accessed by the Destination system through a WAL call so this functionality will only be used when a new user is created. (Default=None)

- **MaxErrorsAllowed** – sets the number of errors that are allowed within the period specified by **ErrorDelay**. If this number is exceeded, the program will stop. (Min=1, Max/Default=65535)

- **ErrorDelay** – sets the window of time (in seconds) for which **MaxErrorsAllowed** measures its maximum errors. The minimum is 5 seconds. A setting of zero will default to 5 seconds. (Default=5, Max=10)

- **MaxFetchChildren** – sets the maximum number of Indexes, Documents and Annotation binary packets Worker(s) can process asynchronously. The default is 3 but the value can be changed to a minimum of 1 and a maximum of 8 in an effort to adjust the load. (Default=3, Max=8)

IMPORTANT NOTE: When the **MaxFetchChildren** keyword is set to equal one (1) there will be no wkr* journal log created as all runtime output will be directed to the eng* journal log. All other journal logs will be present.

- **LoadLimit** – allows the user to designate the maximum number of logs to process during an interval as set sizes. Currently there are only two options: 1 hour or 24 hours. The entire group of logs found in the Activity Logs directory will be pulled to the Destination. However, the amount of logs built into binary packets, filtered and processed at one time will depend upon whether the **LoadLimit** is set to 1 hour or 24 hours. Any numeric value entered between 1 and 24 will result in a 1 hour set size. Any numeric value entered above 24 will result in a 24 hour set size. (Minimum=1, Max/Default=24)

  Example:
  When **LoadLimit=1** and logs present represent 3 days of work, the entire contents of the Source Activity Logs area will be pulled to the Destination. The first hour (oldest hour/day) of logs will be processed, then the next and so on until all the "sets" of logs are processed. These sets will be done one at a time during one interval. If the interval is set for 5 minutes but each set takes 5 minutes and you have 30 sets it will take 150 minutes to complete the interval. During the 150-minute interval ISR will continue to calculate the configured 5-minute

intervals. Once the 150-minute interval is completed the next interval's start point will be the balance of minutes left over in the current time span of minutes.

- **TimeOut** – The TimeOut keyword has two purposes however the settings for those purposes are dynamic and will vary dependant upon conditions.

  The RPC portion of the TimeOut affects the ISR_Monitor executable. The ISR_Monitor is responsible for the RPC calls between the systems within the ISR Net. It is the retry of RPC calls that can be configured by this portion of the TimeOut setting. An ISR_Monitor when first started will retry its call to its peer system for up to an hour or until the ISR_Engines start. Once the ISR_Engines start the retry timeout will be set to the interval configured for processing which is found in the ISR_net.csv and consequently set in the ISR_Engine when it starts. To reduce the default retry of an hour the TimeOut keyword can be set to a minimum of zero (0) seconds or a maximum of 30 seconds. If the TimeOut keyword is set for a setting, which is higher than 30 seconds, the RPC retry will default to 30 seconds.

  The second purpose of the TimeOut keyword determines the length of time that the ISR_Engine will wait between retry attempts when setting the FlowFlag. During startup the ISR_Engine(s) will attempt to set the FlowFlag when running in bi-directional mode. The state of the FlowFlag is initially dependent upon the Destination and cannot be set by the Source until the startup state is correctly set for access Source access by the Destination. While the Destination Engine is doing its work in the first part of the interval the Source Engine will sleep and wake based on this timeout value until it is able to grab the FlowFlag and start its phase of the interval. Once both engines have completed both phases of their interval the ISR_net.csv interval setting will be the figure the time counter will use to control this sleeping retry logic. This purpose carries a minimum setting of zero (0) and a maximum setting of 5 minutes. The value is entered in seconds where 300 seconds would be 5 minutes. (Default = 300 seconds)

- **PendErrHrs** – is the time variable in hours attributed to how long items in the pending.log will be retried before being sent to an ERRpak. Pending items will be housed in APPpak during and after retries until the PendErrHrs value is reached. Specific items will be sent to pending to allow P8-IM processes to complete before sending associated events to error should there be any delays. A retry of an event will take place at the beginning of each interval until the PendErrHrs setting has been reached. (Max/Default=24)

  Items dependent upon document committals include:
  - Folder Filing of a document
  - Document Index Record updates
  - Annotations on a document

  Document duplication via ISR can also be sent to pending when:

- Cache space is unavailable thus causing the duplication of a document to stay in pending through intervals until either cache space has been freed or the PendErrHrs time limit has reached.

- **CleanUpHrs** – is the time variable in hours attributed to how long the following list of successful log and pkt files will remain on the system. Successful criteria are met when OKpak files are present but ERRpak files for that same interval are not. When the time variable is set, ISR will check for OKpak files as well as ERRpak files. Should criteria for deletion exist the working files that fall within the time frame will be removed from the WorkingDirectory. (Minimum=1 <hour>, Default=0 <items will not be deleted>)
  - **APPpak/old**
  - **OBJpak/old**
  - **WKRpak/old**

- **Timing** – provides for log output used in performance analysis. The default value is set to FALSE. Setting the value of this variable to TRUE will cause additional information to be logged. The timer.awk program can used along with this added log information for performance analysis.

  Usage for UNIX Operating Systems and Windows Operating Systems with UNIX Tools for DOS loaded: (Some newer UNIX systems may require the use of nawk instead of awk.)

  awk –f timer.awk journals/imp20020328

  nawk –f timer.awk journals/imp20040129

IMPORTANT NOTE: Timing=False will be overridden and internally set to True by debug when turned on using the –D option.

IMPORTANT NOTE: Solaris 5.8 requires the use of nawk, as awk will give poor results. The nawk program can also be used instead of awk on other UNIX Operating Systems when present.

IMPORTANT NOTE: For this functionality to work within a Windows Operating System environment the "awk" executable must be available. If you are unfamiliar with this executable please contact your FILENET System Consultant for information.

- **YearFmt="%Y"** – This is the display format for the year portion of the date which is appended to the ISR journal files. This parameter can change the length of a file name. The default is '%Y' which displays in the 4-digit format "yyyy". A 2-digit "yy" format can be displayed by changing the default to '%y'.

The sample configuration file, **ISR_cfg.sample**, is shown below:

```
;*********************************************************************
;       LOGON INFORMATION
```

```
;*************************************************************************
;ISR.cfg.sample
LogonAttribute {
        UserName="SysAdmin"
        PassWord="SysAdmin"
; For a source system the Domain should be the source domain
; For any other system (target or 3rd party) the Domain
; should be the target domain
        Domain="profserv" (MUST change to correct domain)
        Organization="FileNet"
}
; General Information
PgmAttribute {
; HOME indicates the directory where this config file is found. The
; config file is usually found in the utility installation directory.
;
; Directory used for ISR Working files and their directory structures
;    WorkingDirectory=d:\icer (Windows)
;    WorkingDirectory=/icer (for UNIX>
;      JournalDirectory={Default=<HOME>}
; Default Password on a Create of a Security Object is “None”
;    DefPass=”None”
;
; Maximum errors which can take place within a specified time period
; before ISR will halt.
;    MaxErrorsAllowed=(Minimum=1, Maximum/Default=65535)
;
; Specified time period used by MaxErrorsAllowed
;    ErrorDelay=(Default=5 seconds, MAX=10)
;
; Maximum number of Index/Document/Annotation objects Worker can process
; asynchronously
;    MaxFetchChildren=(Default=3, Max=8)
;
; TimeOut is the time variable of how long the Destination ISR will
; wait to perform a retry when a Document or Annotation is being
; duplicated or replicated from a Source system.
;    TimeOut=<Default 1000 secs>
;
; Time period pending actions will be retried before sending to ERRpak
;    PendErrHrs=<Default 24>
;
; Time period specific working files will be retained before they will
; be automatically deleted.  The default is NOT to delete any files.
;    CleanUpHrs=<Default 0>
;
; Timing provides for log output used in performance analysis
;    Timing=(Default is FALSE)
;
; Display format for the year %Y <default> is ‘xxxx’, %y is ‘xx’.
; This effects displaying of a day only and extends the length
```

```
; of the file name.
;    YearFmt="%Y"
;
; Creates hourly or daily (default) sets of Activity logs to be
; processed within an interval.
;    LoadLimit=(1 for an hour or default of 24 for a day)
}
;******* END OF CONFIGURATION FILE ****************************************
```

IMPORTANT NOTE: A line beginning with a semi-colon (;) is a comment line and is ignored by the program. Blank lines are **not** valid in the **ISR.cfg** file. If a blank line is introduced into this configuration file, the utility will not run.

IMPORTANT NOTE: The above example shows directory paths in the UNIX format. For a Windows system, the directory path entry in the **ISR.cfg** file should follow the correct Windows path conventions.

6.2 Configuring Encrypted Password (optional)

The mechanism to provide values for an P8-IM login through the Professional Services Utilities is to set the values of the UserName and PassWord in the LogonAttributes of the ISR.cfg file. There are two ways to enter this information. The System Administrator can enter the values for the UserName and PassWord via ascii text form. This means that the password for the user will be visible to anyone with read access to the ISR.cfg file. The second way to configure these login attributes is to use the Encrypted Password functionality. Follow the instructions below to use this functionality.

The System Administrator will use the PS_Password program to create the ".ps_passwd" file which will contain the encrypted records. The first time the PS_Password program is run the System Administrator will need to provide a password for the creation of the ".ps_passwd" file as well as the password(s) for any Professional Services Application and User Name. The .ps_passwd file password must be remembered or the System Administrator will not be able to run the PS_Password program a second time to make changes or additions to the .ps_passwd file.

In bi-directional mode each system will need to have its own PS_Password configuration run.

Login as **fnsw** and run the **PS_Password** program:

UNIX:

cd /fnsw/local/bin
>PS_Password

Windows Server:

Open a DOS window
cd <drive>:\fnsw_loc\bin

PS_Password

The first time the **PS_Password** program is run, you will be notified that it is running in "*Initialization Mode.*" You will then be asked to Enter and Reenter (validate) the password for the *Password Application Key*. (The character limit for this password is 8 characters and it is not changeable once it has been set.) The *Password Application Key* is the password that will be used to run the **PS_Password** program a second time. Subsequent runs of the program will require the System Administrator to enter this password and the *Initialization Mode* message will not be displayed.

The **PS_Password** program will then ask for the *PS Application Name*, a *Password* for the login being used and the *User Name* of that login. You MUST provide the P8-IM SysAdmin user ID.

Once the application name, password and user name have been entered the program will display the application and user name with validation that the password has been set. Another application password can be set at this time. Should you not wish to provide any other application passwords you may simply select the **Enter key** at the next prompt**,** the **PS_Password** program will discontinue and display a list of all applications and user names entered.

Example on a UNIX system:

```
/fnsw/local/bin> PS_Password
Creating PS Password File - Initialization Mode
  Enter Password Application Key      :
  ReEnter Password Application Key      :
        Enter PS Application Name       : ISR
        Application (39)'ISR'
        Enter New Password for Application ISR        :
        ReEnter Password for Application ISR        :
          Enter User Name       : SysAdmin
          PS Program 'ISR': User=SysAdmin, Password Set
  Enter PS Application Name       :
**** Current Applications Set ****
**  Application (39)'ISR' [User='SysAdmin']
*********************************
```

Now that the **PS_Password** program has been run and the *.ps_passwd* file has been created the System Administrator can configure the **UserName** and **PassWord** values in the **ISR.cfg** file.

The **UserName** value in the configuration file should be set to the application name entered into the .ps_passwd file. In this case that would be **UserName="ISR".**

The **PassWord** value in the **LoginAttribute** section of the **ISR.cfg** file should be removed or commented out. If the ISR_Monitor sees the **PassWord** parameter in the **ISR.cfg** file during start-up it will NOT use the encrypted password that was entered when the **PS_Password** program was run and ISR will not run.

Example of the **ISR.cfg LogonAttribute** section when using the encrypted password functionality:

```
;*************************************************************
;       LOGON INFORMATION
;*************************************************************
;ISR.cfg.sample
LogonAttribute {
        UserName="ISR"
;       PassWord="" <this parameter has been commented out>
; For a source system the Domain should be the source domain
; For any other system (target or 3rd party) the Domain
; should be the target domain
        Domain="profserv" (MUST change to correct domain)
        Organization="FileNet"
}
```

IMPORTANT NOTE: Should the password, which allows access to the **.ps_passwd** through the **PS_Password** program, be lost or forgotten, the **.ps_passwd** file will have to be deleted and recreated via the **PS_Password** program Initialization Mode.

6.3 WQS Queue list (optional)

The instructions in this section should only be followed if you are implementing Queue (WQS) content duplication. This ISR functionality provides for a mechanism to handle the duplication of items contained inside specified queues listed in this queue list.

The creation/updates of Queue Definitions and Workspaces is handled in the main ISR interval. Workspace and Queue structures and attributes are a separate area of duplication than the actual queue contents. This is because the P8-IM Activity Logging provides event logs for queue “Definition” and “Workspace” changes only. It does not provide for field value changes that make up the contents of a queue. The separation of this functionality allows for the physical queues to remain in sync per your sites requirements without the overhead of copying queue contents at every Object interval. The user has the option of copying queue contents when the Source system(s) are quiet thus creating less overhead during heavy production hours.

The configuration file provides an option of queue contents to be copied at the end of every Object interval should site requirements dictate this level of synchronization. It must be noted that this option will create extreme amounts of data transfers that could impact P8-IM and network traffic.

Queue contents are updated by a block copy of the entire contents of a queue specified in the “WQS List File.” The configuration factors that allow this functionality to take place are setup when ISR_Config is run and the System Administrator sets the WQS List File path along with the interval for WQS duplication. This interval is a separate configuration item, different than the interval setting for processing available P8-IM Activity Logs.

In bi-directional mode each system will need to have its own Workspace / Queue list. It is important to note that one queue cannot contain active work on both systems. A queue must only be accessed from its home system and the peer system will simply be a backup of that queue's contents.

To create the WQS List File use the following format:

Format:

```
<workspace>.<queuename>
```

Example:

```
Workspace1.Queue1
```

NOTE: The WQS List File is checked for correctness. An error is logged if the <workspace>.<queuename> is invalid.

NOTE: To configure the Queue contents to copy at the end of every Object interval "Int (min)" simply set the "WQS Int" setting to 0 (zero). When the "WQS List File" entry contains a "file list name" and is not set to "NONE" and the "WQS Int" setting is 0 (zero), it will perform the queue contents copy at the same interval setting as the Object copy. If the "WQS int" is less than the Object Interval, the value is ignored and the interval set to Object interval. The "WQS Int" cannot be less than the Object interval.

NOTE: When NO Queue contents copy is desired the value must be set to NONE. The ISR_Config application will enter this setting automatically upon opening the application, however, deleting the path/filename can leave the value blank. A blank in this value will cause a failure when ISR_start is executed.

6.4 ISR_Config Java application overview

The **ISR_Config** Java application is used for further required configuration setup on both the Source and Destination systems. The below information contains setup options for both single-direction and bi-directional mode. Please refer to the section which pertains to your environment.

IMPORTANT NOTE: In a single-direction environment the ISR_net.csv files created by ISR_Config for the Source and Destination should have matching entries. The ISR Net Configuration Screen should match in the following entry areas: Source, Method, Destination, Int(min), Doc, Index, Ann, Folder, Sec and WQS.

In a single-direction environment: the application must be used to create configuration files as follows:

System A – create the ISR_net.csv and the act_log_conf files

System B – create the ISR_net.csv

In a single-direction environment: the first window in the ISR_Config UI will contain entries:

Source = SystemA (same on both systems)

Method = "==>" (same on both systems)

Destination = SystemB (same on both systems)

Int(min) = numeric (same on both systems)

Activity Log Path = absolute path (unique on Destination, not used on Source but must be a valid folder/dir structure on local system for ISR_start validation)

WQS List File = absolute path including filename (unique on Destination, not used on Source so can be set to NONE on the Source)

WQS Int = numeric (set on Destination system, not used on Source so can be set to zero "0" on the Source.)

Doc = KEEP, CVT, MSC, or NONE (same on both systems)

Index = KEEP, CVT or NONE (same on both systems)

Ann = KEEP, CVT or NONE (same on both systems)

Folder = KEEP, CVT or NONE (same on both systems)

Sec = KEEP, CVT, DUP or NONE (same on both systems)

WQS = KEEP, CVT or NONE (same on both systems)

Comments = Text input (same or unique on both systems)

IMPORTANT NOTE: In a **single-direction** mode the SEC setting has 4 options. Your FILENET Implementation Consultant can advise on the proper setting for correct object access in your environment.

In bi-directional mode the ISR_Config Java application MUST be run in its entirety on BOTH systems. Each system in a pair will be treated as though it were a Source system. The entries in the first four windows of the UI containing Source, Destination and interval values MUST match. However, the remainder of the available entries can differ in that each system can have its own working directory, queue list, queue interval and comments value that differ from its peer system.

In a bi-directional environment: the application should be used to create configuration files as follows:

System A – create the ISR_net.csv and the act_log_conf files

System B – create the ISR_net.csv and the act_log_conf files

In a bi-directional environment: the first window in the ISR_Config UI will contain entries:

Source = SystemA (same on both systems)

Method = "<==>" (same on both systems)

Destination = SystemB (same on both systems)

Int(min) = numeric (same on both systems)

Activity Log Path = absolute path (unique on both systems)

WQS List File = absolute path including filename (unique to each system)

WQS Int = numeric (same or unique on both systems)

Doc = KEEP, CVT or NONE (same on both systems)

Index = KEEP, CVT or NONE (same on both systems)

Ann = KEEP, CVT or NONE (same on both systems)

Folder = KEEP, CVT or NONE (same on both systems)

Sec = DUP (which is the default in bi-directional mode)

WQS = KEEP, CVT or NONE (same on both systems)

Comments = Text input (same or unique on both systems)

IMPORTANT NOTE: In a **bi-directional** mode the SEC setting will always be set to **DUP** as this is a requirement for bi-directional object access.

To start the **ISR_Config** Java application:

UNIX – enter ISR_Config at the command prompt

Windows Server – enter ISR_Config in a DOS prompt or double click on the executable from the Home directory folder

6.4.1 Using ISR_Config on Source P8-IM systems

The ISR_Config executable will be used to setup 2 necessary configuration files on all your Source systems. ISR NET information is housed in the first window displayed and the P8-IM transaction logging can be setup through a display available when selecting the “Act Log” button from the first display. This first section of instructions below are the configuration items found when the user selects the “Act Log” button from the first display.

Start the ISR_Config Java application on your Physical Source system

On UNIX, set your DISPLAY variable and run:
> **cd /fnsw/local/bin**
> **ISR_Config &** (run in background mode)

In a Windows environment run:
select the **<drive>:\fnsw_loc\bin** folder
double click on **ISR_Config.exe**

or

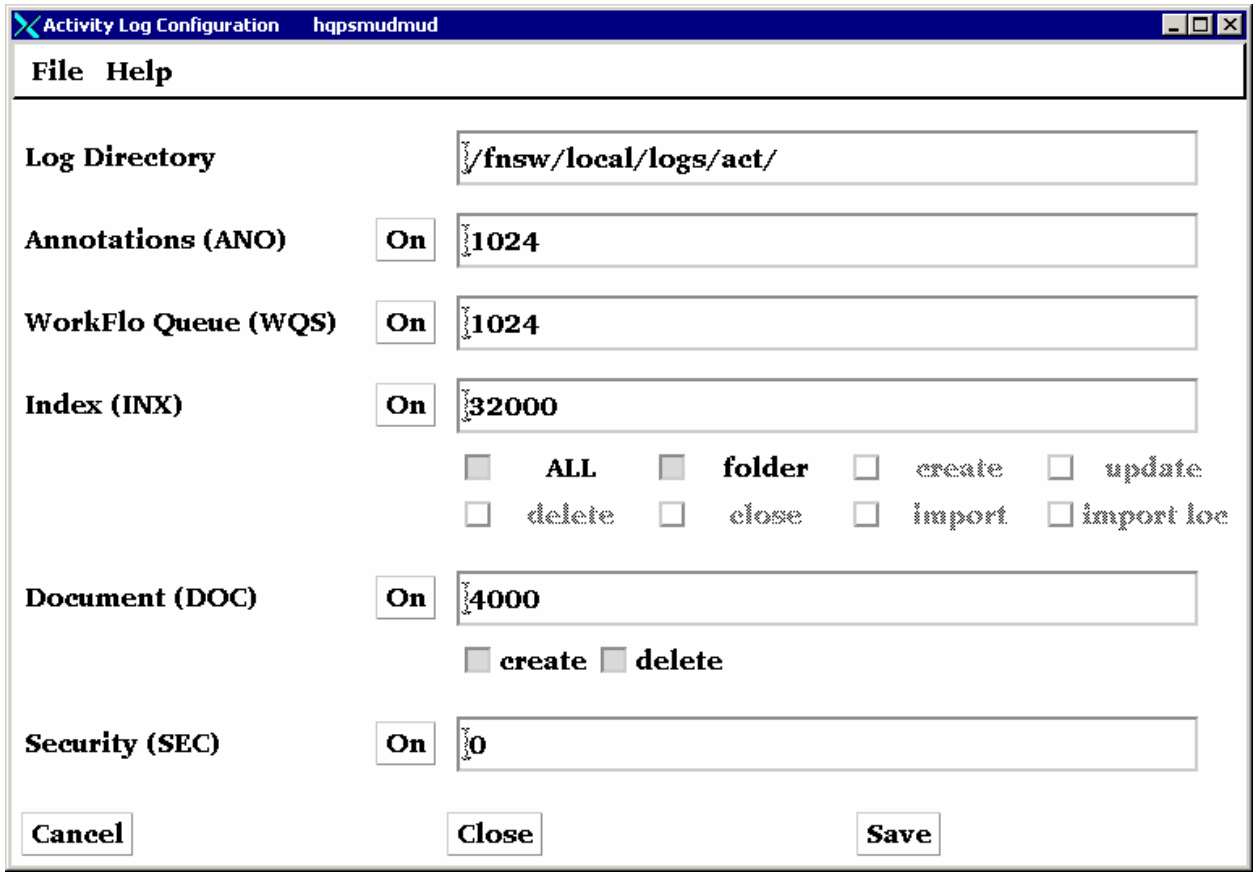
open a DOS prompt window

cd <drive>:\fnsw_loc\bin

enter **ISR_Config** on the command line

The first window displayed will be used for entering information that specifies the relationship between the Source system you are running ISR_Config on and its Destination system. Valid data must be entered into these items, however, in a single-direction environment, the data on the Physical Source system is checked for path validity and file presence but this information is not used. Descriptions and Instructions of these items are contained in the section immediately following this section.

Once the Java configuration application has been started on all Source systems, select the **Act Log** button on the first window displayed to display the Activity Log Configuration window shown below.



The Log Directory path should be entered making sure to include the drive letter when configuring in a Windows environment. The path MUST end with the correct forward or backward slash for your environment.

Select which events you would like the P8-IM Activity Logging to log. This is done by selecting the button adjacent to the text listing and toggle the button to the On or Off position. The Activity Log Configuration window will display default buffer sizes for these logging areas. You may use these predefined kb sizes or adjust the sizes to fit your site requirements.

Descriptions of each event follow. After making the selections to support your environment select the **Save** button and then the **Close** button to return to the main **ISR NET Configuration** window.

Description of P8-IM Activity Logging Events

The P8-IM Activity Logging has been implemented to log specific events that occur on an P8-IM Domain Server. The System Administrator must activate this functionality or logging will not occur. The 5 areas which events can be logged are: Document, Index (includes Folders), Annotation, Security and WorkFlo Queue (WQS). These areas are explained in more detail below.

The general functionality of P8-IM Activity logging will not be able to log events any more granularly than described below. In the event your site may require a further breakdown of event, a site can create a tool that can filter the Activity logging files before ISR has access to them. In this way only site specified events will be made available to ISR for duplication.

Log Directory – This is the path that the P8-IM Activity Logging will create the event log files in. **The "Log Directory" path must be followed by a back-slash for Windows Server environments or a forward-slash for UNIX environments.**

Annotations (ANO) – Selecting "On" will allow the P8-IM to log all events generated by the creation, updating and deletion of annotations on the system. The text box contains the buffer size to house these events until an ISR_FlushLogs is performed or the buffer becomes full and writes out these events to a log file. Selecting "Off" will stop all annotation creates, updates and deletes from being captured by P8-IM Activity logging. This area can be turned ON or OFF with no other variables allowed.

WorkFlo Queue (WQS) – Selecting "On" will allow the P8-IM to log all events generated by the creation, updating or deletion of Queue Definitions or Workspaces. The text box contains the buffer size to house these events until an ISR_FlushLogs is performed or the buffer becomes full and writes out these events to a log file. Selecting "Off" will stop all WorkFlo Queue events from being captured by P8-IM Activity logging.

The entire contents of a Queue can also be duplicated or replicated through ISR but this is not a function that results from any P8-IM Activity logging event. The ISR Destination system will contain a parameter in the ISR_net.csv file that allows for a Queue list and interval, which is independent of the ISR interval that processes the P8-IM Activity logging files. Each time the WQS interval setting is reach ISR will perform a read of the Queue file list and copy the entire contents of each WQS queue from the Source system to the Destination system.

IMPORTANT NOTE: Currently only WorkFlo Queues are supported queue structures for P8-IM Activity Logging. These queues should not be confused with the Workflow Queues found in the Visual Workflow, eProcess, or P8 products. Those structures are not supported at this time.

Index (INX) – Selecting "On" will allow the P8-IM to log specified events associated with Index logging. Index events include events in both the index and folder areas. The index specific, Index events that can be logged through the P8-IM logging include "document index record" or "dir" create, update, delete, close, import, and import_loc. The folder specific, Index events that can be logged include "folder" create, update, delete, close, move, copy, rename, file, and unfile. The text box contains the buffer size to house these events until an ISR_FlushLogs is performed or the buffer becomes full and writes out these events to a log file. Selecting "Off" will stop all Index and Folder events from being captured by P8-IM Activity logging. These events cannot be independently set which allows for Index events to be either entirely ON or OFF.

The buttons available under Index (INX) allow for further granularity of logged events. It is **IMPORTANT TO NOTE** that the button "ALL" only relates to Index events and will not effect a "folder" event. Please read the following bullets for specifics on how the Index (INX) section in the display can be set.

- ALL = all Index related events will be logged. These include "document index record" create, update, delete, close, import and import_loc. **The "ALL" button <u>DOES NOT INCLUDE</u> "folder" events.**

- folder = all Folder events will be logged. These include Folder create, update, delete, close, move, copy, rename, file, unfile, file_after, and reorder.

- create = When "ALL" is not selected the user has the option to select "create" to allow for "document index record" creates to be logged. This selection is not necessary, and **<u>should not be turned on</u>**, as any document create will automatically do the "dir create" associated with the document. When both document create and dir create have been selected there will be double the Event logs. Any setting of this type should be discussed with the FILENET Consultant.

  NOTE: In a Multi System committal (MSC) environment, this INX setting, "dir create," should be turned off. This is exemplified in the document create (DOC) section.

- update = When "ALL" is not selected the user can select "update" to allow for "document index record" updates to be logged. Any site doing post-committal updated should have this button selected to log all "dir update" events.

- delete = When “ALL” is not selected the user can select “document index record” deletes to be logged. This selection is normally not necessary as any document delete will automatically do the “dir delete” associated with the document. When both document delete and dir delete have been selected there will be double the Event logs. Any setting of this type should be discussed with the FILENET Consultant.

- close = When “ALL” is not selected the user can select “document index record” close to be logged. This event is a special Index record change and should be logged if your site sets document states to “close.”

- import = When “ALL” is not selected the user can select “document index record” imports to be logged. This selection is normally not necessary as any document create that comes through a Fast Batch will automatically do the “dir import” associated with the document. When both document create and dir import have been selected there will be double the Event logs. Any setting of this type should be discussed with the FILENET Consultant.

  Note: In a Multi System committal (MSC) environment, this INX setting, “dir import,” should be turned off. This is exemplified in the document create (DOC) section.

- import_loc = When “ALL” is not selected the user can select “document index record” import_loc events to be logged. This selection is necessary to log Report Manager Cross Cataloged Events. A Report Manager Cross Cataloged Event does not have any document create associated with it and requires a specifically logged event in order for the item to be duplicated. Any setting of this type should be discussed with the FILENET Consultant.

Document Events:

P8-IM Activity Logging allows the user to separately configure document Creates and document Deletes. The reason for this added configuration versatility is that sites already using Multi System Committal (MSC), or sites wishing to use MSC, will not want to log document or index/import Creates but only document Deletes.

When ISR is used for Doc creates it will be necessary to activate the P8-IM Activity Logging to capture events on the Source system pertaining to both document creates and document deletes.

Document Create (DOC) – Selecting “On” will allow the P8-IM to log all events generated by the creation of Documents. The text box contains the buffer size to house these events until an ISR_FlushLogs is performed or the

buffer becomes full and writes out these events to a log file. Selecting "Off" will stop all Document create events from being captured by P8-IM Activity logging.

The creation of Documents can be done by ISR or through Multi System Committal. Both processes use Fast Batch Committal to supply documents from the Source system to the Destination system. There is an advantage to using Multi System Committal as the Fast Batches are created at the time the document is introduced into the Source system. This process immediately duplicates the document on the Destination. When ISR is used the event must be logged and then, at the designated interval ISR will request a Fast Batch to be created and duplicate the document on the Destination system. Using Fast Batches will require that there be adequate page_cache space to handle the load of images.

Multi System Committal (MSC) - If Multi System Committal is enabled, turn off doc create (DOC), dir import (INX), and dir create (INX) in activity logging. Here are the steps:

1. Run /fnsw/client/bin/ISR_Config (set DISPLAY variable if necessary)
2. Click on Act Log (on the bottom of the screen)
3. Uncheck "create" from the Document (DOC) row.
4. Confirm "delete" is checked
5. Uncheck "ALL" from the Index (INX) row.
6. Check INX "update, delete, close and import loc." (INX "create and import" should remain unchecked)
7. Confirm that "folder" is checked.
8. Click Save then Close
9. Click the pull down box for DOC and select MSC
10. Click Save then Close

You must recycle your P8-IM system to make the activity logging changes effective.

Document Delete (DOC) – Selecting "On" will allow the P8-IM to log all events generated by the deletion of Documents. The text box contains the buffer size to house these events until an ISR_FlushLogs is performed or the

buffer becomes full and writes out these events to a log file. Selecting "Off" will stop all Document delete events from being captured by P8-IM Activity logging.

Security (SEC) – Selecting "On" will allow the P8-IM to log all events generated by the creation or updating of any Users or Groups. The text box contains the buffer size to house these events until an ISR_FlushLogs is performed or the buffer becomes full and writes out these events to a log file. Selecting "Off" will stop all Security events from being captured by P8-IM Activity logging.

P8-IM Activity logging will log Security creates, imports, renames, updates and deletes for all User and Group changes. This logging allows for a setting of ON or OFF. It is also suggested that the buffering be set to zero (0), which will ensure all Security events are included at every interval.

IMPORTANT NOTE: Any changes through P8-IM Security Administration to Security "Devices", "Functions" or User passwords will NOT be logged by the P8-IM Activity logging and will therefore not be Duplicated/Replicated by ISR.

IMPORTANT NOTE: Synchronization of the Source(s) and Destination Security Databases must include the 4 default groups and 3 default users from the Source systems being created onto the Destination system.

IMPORTANT NOTE: Professional Services personnel will ensure the synchronization of the Source and Destination Security Databases.

Destination Group Security must have the following:

AuditG:source:FileNet
FieldServiceG:source:FileNet
OperatorG:source:FileNet
SysAdminG:source:FileNet
AuditG:destination:FileNet
FieldServiceG:destination:FileNet
OperatorG:destination:FileNet
SysAdminG:source:FileNet

SysAdmin:source:FileNet
SysAdmin:destination:FileNet
Operator:source:FileNet
Operator:destination:FileNet
FieldService:source:FileNet
FieldService:destination:FileNet

Bi-directional mode – Both Source and Destination Security Databases must each have the alternate system's 4 default groups and 3 default users.

6.4.2 Definition of P8-IM Activity Log file entries

The P8-IM Activity logs contain a cryptic entry line to describe an event. ISR_Builder converts the line to an binary packet file with instructions to perform the duplication. Below is a list of event definitions as contained in the P8-IM Activity log files. See the following section entitled " P8-IM Activity Logs" for actual logging examples of these events.

Any direct insertion into the database that does not include standard P8-IM functionality will NOT log an event in the P8-IM Activity Logs. If any site is concerned as to whether a custom implementation will create a logged event they should test these actions in conjunction with P8-IM Activity logging to ensure functionality is present before implementation of ISR.

Please consult the Image Manager Activity Logging Manual for the most complete reference of all event descriptions.

####### 0 0 = performs a "document" create, delete or update where the ###### represents the document ID number

Document create actions are logged by any document added to an IS. These can be initiated via IDM Desktop, IDM Web, any WAL application or Fast Batch process, Import, etc.

dir = performs a "document index record" create, delete, update, close, import or import_loc

Document Index Record create events are logged when a Document create is logged. DIR update events are logged when Index values are changed as well as document attributes. DIR close events are logged when a documents state is set to "close." DIR import events are logged when documents are imported into an P8-IM via COLD and Fast Batch (HPII). DIR delete events are logged when a document is deleted from the FILENET system. DIR import_loc events are logged when Report Manager Cross Cataloging takes place.

folder = performs a "folder" create, delete, close, move, copy, rename, file, unfile, file_after, reorder, or update

Any folders that are created or have their attributes changed will cause a folder create or folder update action to be logged. A deleted folder will cause a folder delete action to be logged.

user = performs a "user" create, delete, rename, import, or update

The creation or deletion of a "user" will cause an event to be logged. Changing the attributes of an existing user will cause a "user update" to be logged. When the "add user" or "edit user" functionality is used to add a user

to a group the P8-IM Activity Logging will log a “user create” or “user update” as well as a “group update.”

group = perform a “group” create, delete, rename, import or update

The creation or deletion of a “group” will cause an event to be logged. Changing the attributes of an existing group will cause a “group update” to be logged. When the “add group” or “edit group” functionality is used to add a user to a group the P8-IM Activity Logging will log a “group create” as well as a “group update.”

workspace = perform a “workspace” create, delete, rename or update

The creation of a workspace will cause a “create workspace” to be logged. Any other updates on an existing workspace or deletes will create “update workspace” or “delete workspace.” However, any changes attempted on an existing workspace with queues are bound by P8-IM functionality restrictions and will not be logged should errors result in the attempted operation.

queue = perform a “queue” create, delete, empty, rename, move or update

The creation of a queue will cause a “create queue” to be logged. Any updates on an existing queue will cause a “update queue” to be logged. When a queue is deleted the P8-IM Activity Logging will log a “delete queue.” It is important to remember that queue changes are bound by the same P8-IM functionality restrictions as workspace changes. Please refer to the Image Manager documentation for more information on these restrictions. Should errors result in the attempted operations the action will not be logged.

Queue contents bulk copy = all contents of a designated queue will be copied

The ISR functionality, which provides for queue contents to be copied from the Source system to the Destination, is driven by a separate interval specifically designated for this process. The Queue List file will tell ISR which queue(s) to copy while the Queue Interval will designate when.

The queue item bulk copy must get a lock on the Destination queue to perform the copy. Should the Document Class on the Destination have inserts into that queue turned on the bulk copy will fail because of an inability to get a lock when the bulk copy interval is initiated.

6.4.3 Using ISR_Config on Destination and Source P8-IM systems

The Destination system will need to have all systems that will be seen as Source systems within the ISR NET configured by using the ISR_Config executable. The ascii file created during this procedure is named **ISR_net.csv**.

The Source systems must also have this step performed as there are configuration items necessary for program runtime that will be required. When run on the Source it is important to place only the one (itself) Source system's domain name in the "Source" text box and the Destination system's domain name in the "Destination" text box. The "Activity Log Path" will not be used but must be a valid path on the system.

On UNIX run:

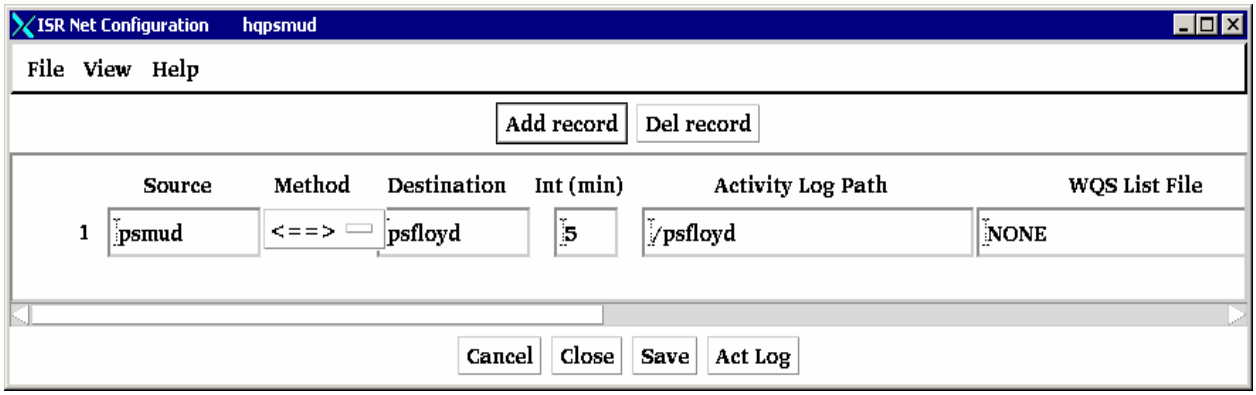
> **/fnsw/local/bin/ISR_Config &**

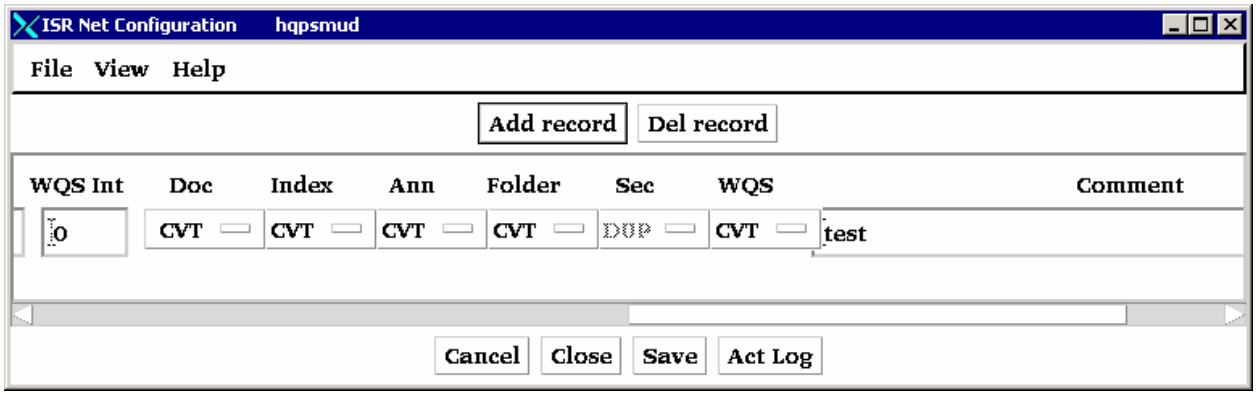
On a Windows environment run:

<drive>:\fnsw_loc\bin\ISR_Config.exe

Once the Java configuration application **(ISR_Config)** has been started select all configurable items and set properties for your **ISR NET** environment. An example of the application interface follows the descriptions below.

The below example shows the Destination system (psfloyd) configured to run with one Source system (psmud). The systems are configured to run in single-direction mode. The screen shot provided has been cut into two sections, as the width of the viewable display is dependent upon scrolling within the application window.





Add record button – will provide a second, third, etc. line for adding multiple Source systems to your **ISR NET**.

Del record button – will remove a line from your **ISR NET** configuration.

Select the delete button and a window will be displayed requesting the Administrator to select which numeric line is to be deleted. The Administrator can select a line number and then delete or cancel the process.

Source text box – enter a Source domain name to be included in your **ISR NET**.

Method selection box – allows for the ability to select between Single-direction "==>" and Bi-directional Duplication mode "<==>".

Destination text box – enter the name of your Destination domain, enter the same Destination domain name in all lines if you have multiple Source systems. This Destination domain will receive log files from one or many Source system(s).

Int (min) text box – enter time in minutes for how long ISR should wait between time intervals for processing of P8-IM Activity Log files provided by one or many Source system(s). Your Professional Services Implementation Consultant, using your site's load and requirements, will tune the time interval.

Activity Log Path text box – enter the path to the Source log files

UNIX – mount point exported from Source Activity Log area and mounted on Destination system.

Windows Server – UNC path to a shared Source Activity Log area.

Example:

\\sourcesystemname\sharedfoldername

WQS List File text file – enter the path to the WQS ascii file which is usually created in the ISR install directory. See WQS Queue list section for more details on the creation of this file.

WQS Int text box – enter the time in minutes for how long ISR should wait between the intervals that will bulk copy selected queue contents from the Source system(s). See WQS Queue list section for more details on this functionality. The time interval will be tuned by your Professional Services Implementation Consultant for your site's load and requirements.

Doc selection box – select the method of Document duplication for your site.

CVT = convert the domain property references on the document from the Source domain reference to the Destination domain

NO = do NOT process log events regarding documents

KEEP = keep the Source domain property references on the document

MSC = Multi-System Committal mode – ignore doc creates, dir creates and dir imports

IMPORTANT NOTE: If DOC = MSC, "KEEP" will be set for logging events and will override all other activity settings except SEC.

IMPORTANT NOTE: If DOC ≠ MSC, what ever value DOC is set to will be set for Folder (FLD), Index (INX), and Annotation (ANN) and will override any other values those events are set to.

Index selection box – select the method of Index item duplication for your site.

CVT = convert the domain property references on the Index items from the Source domain reference to the Destination domain

NO = do NOT process log events regarding Index items

KEEP = keep the Source domain property reference on the Index items

Ann selection box – select the method of Annotation duplication for your site.

CVT = convert the domain property references on the Annotations from the Source domain reference to the Destination domain

NO = do NOT process log events regarding Annotations

KEEP = keep the Source domain property reference on the Annotations

Folder selection box – select the method of Folder duplication for your site.

CVT = convert the domain property references on the Folders from the Source domain reference to the Destination domain

NO = do NOT process log events regarding Folders

KEEP = keep the Source domain property reference on the Folders

Sec selection box – select the method of Security duplication for your site.

CVT = convert the domain property references on the User and Group Security items from the Source domain reference to the Destination domain

This will also convert the domain of the group that is authorized to administrate the User account it has been assigned to.

NO = do NOT process log events regarding User and Group Security items

KEEP = keep the Source domain property reference on the User and Group Security items

This will also keep the domain of the group that is authorized to administrate the User account it has been assigned to.

DUP = duplicated the User and Group Security item for both Source and Destination domain names.

The DUP mode will use the domain of the group that is authorized to administrate the User account it has been assigned to that is the same as the domain of the User. An example of this model is:

User1:domain1 -> SysAdminG:domain1

User1:domain2 -> SysAdminG:domain2

IMPORTANT NOTE: Security changes should only be made from the Physical Source system. In a fail-over situation the Destination becomes the Physical Source and can accommodate Security changes. However, all SEC changes must be processed in full before switching back to the original Physical Source system when the disaster that caused the fail-over has been resolved.

IMPORTANT NOTE: In a Bi-directional mode the requirement is that Security Object duplication will be set automatically to DUP. This setting cannot be changed by the user.

WQS selection box – select the method of Document duplication for your site.

CVT = convert the domain property references on the WQS Workspaces and Queues from the Source domain reference to the Destination domain

NO = do NOT process log events regarding WQS Workspaces and Queues

KEEP = keep the Source domain property reference on the WQS Workspaces and Queues

Comment text box – enter any comments for this Source and Domain configuration. This entry is not used by ISR but is provided for System Administration information.

IMPORTANT NOTE: When upgrading from 1.x to version 2.x you must open the ISR_Config application and resave the existing values.

6.5 Activity Logging Share or Mounted File System

It is mandatory to create a share or mount point for the Source Activity Log folder so that the Destination system can access the available logs for processing. The UNIX

environment will require the directory structure to be exported and appropriate permissions set, so that the Destination system can create a mount point to access the contents of the directory. The Windows environment requires the shared folder has the appropriate permissions to allow access to the Destination system via a UNC path.

In a UNIX environment, set the directory permissions and export the P8-IM Activity Logging file system(s) from the Source system(s). Mount the exported file system on the Destination system to make the Activity Logging on the Source(s) available to the Destination system. This mount point is entered via the ISR_Config application on the Destination system in the Activity Log Path text box. Test to ensure your permission level allows for Activity logging and ISR functionality to occur in the share / mount point. The permissions may need to be 774 or 775 depending upon the overall Security model at your site. The ability for the P8-IM to write the logs, followed by the ability of ISR to copy and delete the logs is necessary.

In a Windows server, the folder must first be shared with appropriate permissions. This share is then entered as a UNC path via the ISR_Config application on the Destination system. The P8-IM must be able to write the Activity Logs, and ISR must be able to copy and delete the logs as necessary.

6.6 Run CheckSSN

Running the CheckSSN program ensures the following:

1. WAL libraries have been installed correctly and can be accessed.
2. A WAL call can be made to the P8-IM to attain an P8-IM login.
3. The ISR.cfg file is present, can be read and validation of its configurable items can be achieved.
4. The PSLSF file is present and correct for the domain it has been created for.

Example:

On UNIX default location installs:

> cd /fnsw/local/bin

On WINDOWS default location installs:

<drive letter>: cd \fnsw_loc\bin

Run the following executable:

> **CheckSSN**

Sample input values:

```
Enter IMS Domain:       <your IMS domain>
Enter IMS Organization: FileNet <or your Organization name>
```

```
Enter User name:        SysAdmin
Enter Password:
Enter Application name: ISR
```

Sample output of successful connection and communication with WAL and IMS:

```
Logging onto 'profserv:FileNet' as 'SysAdmin/SysAdmin'
@'SSNCHECK'
Getting SSN of DocServices
SSN for IMS System 'profserv:FileNet' is 1100202180
   Primary Security File='./PSLSF
  SSN=1100202180,YY=100,DDD=5,NumSSN=126,FirstK=5, Pgm='ISR'
License for 'ISR' - Registered Copy
>
```

If you receive an error running CheckSSN, display the error tuple message to determine why the failure occurred. Correct the problem by installing missing software or configuring access to the IS, and then re-run CheckSSN. Further information on errors can be found in the Troubleshooting section at the end of this document.

6.7 Users / Environment

UNIX Operating System User only
The most common user environment used to run ISR is the ***fnsw*** user. Should a different user be required to run ISR that user MUST be a member of all the same groups that the FILENET ***fnsw*** is assigned to. This user must have read, write and execute privileges for all directories, files and programs used by ISR. The UNIX user must be a member of all the same groups as the ***fnsw*** user (***fnusr, fnop, fnadmin, dba)***.

WINDOWS / UNIX Image Manager User
An Image Manager user is required for logon to the IM. In a **Windows** environment, the **SysAdmin** user is the **only user** that can be used.

In a **Unix** environment, the most common user for this configurable item is the **SysAdmin** user. This user must have read, write, and execute privileges for documents on the FILENET Image Service library. The name and password of this Image Manager user must be entered in the **ISR.cfg** file. This user must also have been configured with at least 10 concurrent logins. (Set this minimum a bit higher than 10 if 8 workers are configured.)

The following criteria should be considered if not running ISR with the **SysAdmin** user:

- The logon attributes should match the Image Manager user, and standard logon information for the FILENET Image Manager system.
- When running any PS application from a user that is not SysAdmin, you **must** set the attributes on the user so that it has the ability to delete security objects.

- The user must be set to 'supervisor'. Being a member of the SysAdminG group is *not* adequate.
- If the customer does not run ISR as SysAdmin, much of the security synchronization cannot be accomplished. SysAdmin has some special privileges that cannot be duplicated or assigned to any other user. Security errors reported in this situation are not ISR errors but environmental (user id) errors.

6.8 Cache Parameters

Cache Parameters are configured through **fn_edit** and will not take effect until an **fn_build** is executed and the FILENET software has been restarted. Since cache usage varies by system, it is not possible to specify default settings for ISR.

When "Document creates" has been turned on through the ISR_Config Java application, ISR will use fast batch committal to copy documents from the Source system to the Destination. This requires that the Destination P8-IM must have access to a contiguous amount of Retrieval Cache (Page Cache) to operate.

If there is insufficient cache available, ISR_Worker will wait and then retry the batch a limited number of times, while cache services attempts to clean the cache. The variables on the Destination system that effect cache usage are the speed of the jukebox to write to an optical platter, the amount of document fast batches being received, and the available Retrieval Cache to house the fast batches while processing takes place. Should Retrieval Cache become limited and thus cause a document write slowdown, it will be necessary to add additional Cache (page_cache), if system resources allow it.

Should the system resources be insufficient, such as limited cache space, ISR can be tuned to slow down its processes. Please see the section on "Configuring ISR."

6.9 Cache Only Systems

Systems that do not utilize OSAR/MSAR, but use CSAR or SSAR no longer require the use of Cache Maintenance (CSMT). ISR can unlock cache objects on destination servers with the following configuration change in ISR.cfg:

For API based SDS devices (using ClipId)

CleanCache=TRUE (False if the Default)

For Non-Api based SDS devices (no ClipIds)

CleanCache=TRUE (False if the Default)

CheckIndexDB=FALSE (True is the default)

7. P8-IM ACTIVITY LOGS

The P8-IM Activity Logs are formatted per FILENET'S RCI 2429 requirements. The following information is to assist the user when the programmatic generation of customized logged events is a site requirement.

The following are Activity Logging samples as generated from an active IS. Newer releases may show the user name for annotation related activities.

Document logs:

```
2002040811.DOC.NON_ACTIVE.0
2002/04/08 11:09:48.078 create 3000120452 0 0
2002/04/08 11:10:24.546 create 3000120453 0 0
2002/04/08 11:10:54.359 create 3000120454 0 0
```

Index logs:

```
2002040811.INX.NON_ACTIVE.0
2002/04/08 11:08:46.578 folder create /NewFolder
2002/04/08 11:09:07.328 folder rename /NewFolder /Thick_10
2002/04/08 11:09:48.281 dir create 3000120452
2002/04/08 11:09:50.484 folder file /Thick_10 3000120452
2002/04/08 11:10:24.718 dir create 3000120453
2002/04/08 11:10:26.781 folder file /Thick_10 3000120453
2002/04/08 11:10:54.515 dir create 3000120454
2002/04/08 11:10:56.578 folder file /Thick_10 3000120454
2002/04/08 11:11:55.656 dir update 3000120454
```

Annotation logs:

```
2002040811.ANO.NON_ACTIVE.0
2002/04/08 11:12:44.062 create 3000120452 1 1
2002/04/08 11:12:44.109 create 3000120452 1 2
2002/04/08 11:12:44.156 create 3000120452 1 3
2002/04/08 11:12:44.171 create 3000120452 1 4
2002/04/08 11:12:44.203 create 3000120452 1 5
```

Workflow Queue logs:

```
2002040811.WQS.NON_ACTIVE.0
2002/03/26 17:34:17.515 queue delete Queue02 WSpace02
WflServer:psverve:FileNet
2002/03/26 17:34:46.843 workspace rename WSpace02 WorkSpace_02
WflServer:psverve:FileNet
2002/03/27 10:42:41.906 workspace rename WorkSpace_02 WSpace02
WflServer:psverve:FileNet
2002/03/27 10:43:17.390 queue create Queue02 WSpace02
WflServer:psverve:FileNet
```

Security logs:

```
2002040811.SEC.NON_ACTIVE.0
2002/06/27 13:33:11.377 create group Group_03:pearl:FileNet
2002/06/27 13:33:11.431 update group Group_01:pearl:FileNet
2002/06/27 13:33:11.467 update group Group_03:pearl:FileNet
2002/06/27 13:33:36.193 create user User_04:pearl:FileNet
```

```
2002/06/27 13:33:36.258 update group Group_03:pearl:FileNet
2002/06/27 13:42:20.467 update group SysAdminG:pearl:FileNet
2002/06/27 13:42:20.530 update group Group_01:pearl:FileNet
```

8. ISR PROGRAMS – FUNCTIONALITY

8.1 Runtime

IMPORTANT NOTE: Do not touch files in the WorkingDirectory when ISR is running. Runtime files may be in the process of being written from or written to. Manual access of these files can corrupt them. Any data collection or viewing should be done when ISR modules are not running.

ISR can be invoked with the following parameters by including the specific flags listed below when issuing the **ISR_start** command. The default mode is no parameters (Verbose Mode). This mode causes **ISR_Monitor** to send log entries to the *nohup* files found in /tmp on a UNIX system. The status during the initialization time is recorded in these log files. Once the balance of the programs are started, the **ISR_Monitor** will cease to write to the *nohup* files while continuing to write any necessary output to the **ISR_Monitor** journal logs.

IMPORTANT NOTE: When starting a PS application using it's 'start' program, by default it will create a file in the current directory called '<app>.out'. This file will be updated with information throughout the start process. The applications 'stop' program will also append information to this file when you use it to shutdown the application.

The other thing that the 'start' program does is establish the location of where the 'nohup' (programs stdout data) files go.

This is determined by examining the following environment variables, in the following order:

"NOHUP_DIR", "WAL_LOG_DIR", "TEMP", TMP", "TMPDIR",

If none of these are found, then the default location is **'/tmp'**.

This is done becasue on some systems (mostly Solaris) the /tmp folder is also the 'dump' location, or is cleaned out at boot time. By examining the environment values in this manner, we allow the customer to 'relocate' the nohup files.

```
ISR_start -h
   Setting to /fnsw/local/bin
   Setting Default 'mon' log file
   Starting 'mon' Journal Log ./journals/mon20050224
Usage:  ISR_Monitor -h [homedir] -q -v [domain]
       ---------------- ISR_Monitor Options ----------------
       -dom <domain>   Required - Source Domain
                       Target Domain is defined in cfg
```

```
                two-part NCH domain, e.g. domain:org
-tz <mins>      Mins +/- from GMT, (Default=Local)
-int <interval> interval in mins, (Default=60)
----------- PS Application General Options -----------
-h              This help message
-h <homedir>    Home Directory
                Location of PSLSF and ISR.cfg
-q              Quiet mode
-v [domain]     Display Version - Option domain
                Provide domain to include IDMIS version
----------- Special - use only as directed -----------
-S              Spy mode
-D              Debug mode
-M              Debug Memory mode
-C <mins>       Clock Timeout
```

8.2 Programs

The following section gives a more detailed description of the operation of the various **ISR** programs.

- **ISR_start (Single-direction)** – is the highest-level program. Its purpose is to start all required programs for the system it is issued on. Different programs may start depending upon whether the system is configured as a Source or Destination within the ISR NET. ISR_start is also used to configure and de-configure the Windows Service functionality. On Windows systems, whether ISR_start is running as a Service or from the command line it will log important information and error messages to the System Application Event Log. See the balance of the program descriptions below for more information.

  "ISR_start –D –S –M" is used for debug output when a debug run is requested from FILENET personnel. The use of these flags should never be used for standard runtime.

IMPORTANT NOTE: On Windows systems, if ISR_start is run in a DOS Command window with a fully qualified pathname including the drive letter, it may appear to hang because the full pathname signifies it is running as a service. To run ISR_start from a Command Line the procedure is to cd to the folder where ISR_start is located and invoke it simply as "ISR_start."

- **ISR_stop (Single-direction)** – signals each program to shutdown and stop processing. When executed, a file called ISR_FLAGS is created in the HOME directory. The ISR_Engine(s) and ISR_Monitor look at this file at non-critical processing points. After ISR_Engine(s) has stopped, ISR_Monitor and ISR_Server will follow. No new Activity Logging files will be processed, but current Activity Logging files will be completed before shutdown.

The Stop command must be issued using the –h *homedir* flag if **ISR_start** was started with the –h *homedir* flag.

- **ISR_start / ISR_stop (Bi-directional)**

ISR_start / Bi-directional – Start Physical Source first using ISR_start. This will start the ISR_Server/ISR_WinServer and the ISR_Monitor on the Physical Source. Then execute ISR_start on the Physical Destination. No delay is necessary. This will start the ISR_Server/ISR_WinServer, ISR_Monitor and ISR_Engine on the Physical Destination. As the processes start on the Physical Destination the Physical Source will be checking for Physical Destination processes to start the ISR_Engine on the Physical Source. When both start sequences have finish, each system will have the ISR_Server/ISR_WinServer, ISR_Monitor and ISR_Engine executables running.

ISR_stop / Bi-directional – First, stop on the Physical Destination first using ISR_stop. This step can be followed an ISR_stop on the Physical Source. Processes will NOT stop until all interval processing has finished. It is important not to manually stop processes that continue to run, as this interrupts event processing, causing the systems to become out-of-sync. It is also important that Image Manager **IS NOT** stopped until all ISR processes shutdown gracefully. Stopping Image Manager during ISR processing/shutdown can cause systems to become out-of-sync.

Should the Source be stopped before the Destination, the Destination's ISR_Engine will detect the lack of resources on the Source side and initiate the stop command. This process may take between 15 and 30 minutes to take place.

- **ISR_Server / ISR_WinServer** – The ISR_Server will be found running in the UNIX environments on both the Source and Destination systems. ISR_WinServer is the Windows form of the ISR_Server. ISR_WinServer will be found running on both the Source and Destination systems. ISR_Server performs and answers all RPC calls within the ISR NET and must be running for ISR_Engine(s) to start on a Destination system.

  The ISR_Server and ISR_WinServer can be run from a command line when FILENET personnel request specific module debugging.

  > ISR_Server &

  > double click on ISR_WinServer.exe

- **ISR_Monitor** – The Monitor will always be found running on both the Source and Destination systems. The Monitor will start and monitor the Engine(s) running on the Destination system. It is also responsible for starting and stopping the ISR_Server/ISR_WinServer.

  Standard out and standard error for ISR_Monitor is redirected to the *NOHUP* file.

IMPORTANT NOTE: Monitor cannot start an ISR_Engine correctly when the P8-IM domain name is equal to the string "Target."

- **ISR_Engine** – In single-direction mode, the Engine(s) will run on the Destination system only. There will be one Engine running for each Source System configured in the **ISR NET**. The Engine is the application's main program. It will make the interval calls and send binary packets through the ISR_Builder, ISR_Filter and ISR_Worker.

  The ISR_Engine can be run from a command line, if FILENET personnel request specific module debugging. The syntax is:

  > ISR_Engine –dom *sourcedomain* –int *numeric_time_set_in_minutes* –D –S

  The "-int" flag is used to set the interval as the interval set in the configuration files will not be read when running ISR_Engine in this mode.

  Bi-directional – The **ISR_Engine** will start on both Physical Source and Physical Destination. Please see details on **ISR_start** for startup sequence.

- **ISR_Builder** – In single-direction mode, the Builder will run on the Destination system only. The Builder is started by Engine when there are Activity Logs available for processing. Builder will create binary packet files from the Activity Logs it accesses from the Source system during an interval.

  To view the contents of a binary pkt file the System Administrator can run ISR_Builder, from the Home directory, with the input below. Note that ISR_Builder must be run from a command prompt on a Windows platform.

  In UNIX:
  cd /fnsw/local/bin
  ISR_Builder –dom domainname –ssn <SSN> –in /*path*/*to*/WKRpak/*yyyymmddtttt*.pkt.

  In Windows :
  cd <drive>:\fnsw_loc\bin
  ISR_Builder –dom domainname –ssn <SSN> –in <drive>:*path**to**WKR*pak*yyyymmddtttt*.pkt

  No wild cards may be used in the above command. When running in a Windows environment be sure to include the drive letter (or UNC path) as well as the correct "forward slash" in the path. See ISR_Builder –h for additional options.

  Bi-directional – ISR_Builder will run on both the Physical Source and the Physical Destination, however, the process will start and stop as called by ISR_Engine.

- **ISR_Filter** – This program accesses the binary packets created by Builder and organizes the work to be done into a logical order and eliminates duplicate work. The Engine appends the packets created with the following naming conventions:

IDAxx – contains annotation, index and document creates, updates, closes, imports, import_locs and deletes
FLD – contains all folder creates, updates, closes, moves, copies, renames, and deletes
FDR – contains all document file to folder actions including "folder" file, unfile, file_after, reorder
SEC – contains all security creates, deletes, renames, imports, and updates
WQS – contains all WorkFlo Queue structure changes found in the workspaces or the queues (including creates, updates, deletes, moves, copies, renames)

Bi-directional – ISR_Filter will run on both the Physical Source and the Physical Destination, however, the process will start and stop as called by ISR_Engine.

- **ISR_Worker** – In single-direction mode, the ISR_Worker(s) will run on the Destination system only. The ISR_Worker will be started by Engine during an interval if there is work to be processed. More than one ISR_Worker can be started depending upon the workload. This work comes in the form of binary packet files created by ISR_Builder. When the available work during the interval is completed, ISR_Worker will terminate.

  When the ISR_Worker cannot perform an action that is considered to be a valid event to be processed it will log that event into the **pending.log** file and retry the event at the beginning of the next interval. Should the event still not be processed the log file will continue to hold the event and attempt to reprocess it again at the start of any subsequent intervals until the PendErrHrs keyword value has been reached. Please review the section on ISR Maintenance for more information on the pending.log functionality.

  Worker WRKpkt binary files will contain the following processing entry keywords at the beginning of each Event line:

  PROCESS – filter has determined that this line needs to be executed
  IGNORE – filter has determined that this line does not need to be executed
  PASS – worker was able to process this line successfully
  FAIL – worker was not able to process this line successfully
  PENDING – worker could not do the work at this time and has set it to be done in a subsequent interval or until the PendErrHrs is reached which will send the Event to ERRpak

  In a bi-directional mode ISR_Worker will generate an ISR Activity binary pkt and an ISR Activity ASCII log. These two items are generated from work done. The location of these files will be:

  <WorkingDirectory>/SSN/OKpak/{***.isr**,***.isr.log**}

  - The ***.isr** (example: 2003031407.SEC.isr) is a binary pkt used by the peer system to filter and stop duplicated Events from entering into an infinite loop.

- The ***.isr.log** (example: 2003031407.SEC.isr.log) is an ASCII text file created for site usage that can be used as an implementation feature for customer driven automated auditing or visual checking of work. This file will represent what ISR did to accomplish the work required in an interval on ONE peer system. It will be necessary to check the opposite peer for an ***.isr.log** listing its interval work.

In an isr.log file you will always see the string "UNSET = " at the beginning of each line. This is the preliminary entry in a binary pkt file that show that no determination has been made to what work will be done. The file has not gone through filter.

Bi-directional – ISR_Worker(s) will run on both the Physical Source and the Physical Destination, however, the process(es) will start and stop as called by ISR_Engine.

- **ISR_Vidamin** – Utility for spot checking system synchronization. For details on how to use ISR_Vidamin please see the section entitled "ISR Reporting, Validation and Maintenance."

- **ISR_Sleep** – Written for Windows as a command line function for batch files to provide for a delay or "sleep." Should not be used in a UNIX environment. Use the UNIX ***sleep*** command.

  For consistency or clarity, it can be used in UNIX, also, as it does the same sleep as provided by UNIX shells.

- **ISR_cfg.{help,txt,zip)** – These three files are JAVA files used during the running of **ISR_Config**.

- **ISR_rpt.{help,txt,zip}** – These three files are JAVA files used during the running of **ISR_Report**.

- **ISR_Client and ISR_WinClient** – These utilities function the same from the user perspective but are compiled specifically for the UNIX and the Windows Operating Systems. Please see the below output and OS designation for each.

  The **ISR_Client** or **ISR_WinClient** programs allow the user, in their designated Operating System, to perform commands, requests and set the **ISR_Server** or **ISR_WinServer** to debug mode.

  The 12 options are pre-configured commands that can be executed by inputting a number at the "***** Command :** " line. Selecting "0" zero or the return key will cause the utility to exit.

IMPORTANT NOTE: *The Windows command,* ***ISR_WinClient****, must be executed in a DOS window*.

```
D:\FNSW_LOC\bin>ISR_Winclient
```

```
*** Enter Domain : psweezer
    - Domain: psweezer = psweezer
    - 10.10.8.20        psweezer
    Remote RPC Server pid for system psweezer is 672
    FlowFlag for system psweezer ip=0.0.0.0(None), pid=0
    Current Domain = psweezer
        Quit Client       = 0
        Set Domain        = 1
        Set Timeout       = 2
        Get Pid           = 3
        Get TZ Diff       = 4
        Get Time Diff(sec)= 5
        Clear FlowFlag    = 6
        Get FlowFlag      = 7
        Set FlowFlag      = 8
        Do a Command      = 9
        Shutdown          = 10
        Toggle Debug      = 11
*** Command :
```

0 = Entering 0 (zero) 'Quit Client' or the return key will cause ISR_Client / ISR_WinClient to exit.

1 = Entering number 1 'Set Domain' will allow the user to input the domain name of the Image Manager domain you would like to execute further ISR_Client/ISR_WinClient commands against. The ISR_Client/ISR_WinClient can be executed on any Image Manager domain that is configured in the ISR NET and has the Images Services Toolkit (WAL) installed.

2 = Entering number 2 'Set Timeout' will allow the user to set the timeout length for any of the commands given through the ISR_Client/ISR_WinClient. Every command entered from the available ISR_Client/ISR_WinClient command list does an RPC call. On systems that may have a longer delay because of distance and/or network capabilities it may be necessary to set the Timeout to a longer length to allow the command to be issued as well as for ISR_Client/ISR_WinClient to receive its return value. The default is 10 seconds.

At the command prompt enter "2." Then at the Timeout prompt enter the number of seconds you would like this timeout rate to be set at.

```
*** Command : 2
*** Timeout : 120
```

3 = Entering number 3 'Get Pid' will return the Process ID for the ISR_Server/ISR_WinServer on the domain to which you are connected.

4 = Entering number 4 'Get TZ Diff' will return the difference between the Time setting within a Time Zone on the system you are connected to through the ISR_Client/ISR_WinClient and Greenwich Mean Time (GMT). If you are executing the command on a domain that is actually set to GMT there will be no difference.

5 = Entering number 5 'Get Time Diff <sec>' will return the time difference in seconds between the system you are connected to through the ISR_Client/ISR_WinClient and the system the utility is actually executed on. If you have executed the ISR_Client/ISR_WinClient on the same system as the domain you are connected, there will be no difference. If the two systems you execute the command against are completely in sync there will be no difference either.

6 = Entering number 6 'Clear FlowFlag' will allow the bi-directional mode controller flag to be set to zero. This setting causes the interval controller to be neutral. Neither of the two systems in the bi-directional ISR NET will be the Virtual Source nor Virtual Destination while the FlowFlag is at zero. Upon restart of ISR the Physical Destination will set the FlowFlag to itself to begin normal processing within an ISR interval. Clearing the FlowFlag can be necessary when ISR modules are not shutdown cleanly. This can occur when a system loses power or the Image Service has shutdown prior to ISR shutting down. Please see steps in the Troubleshooting section at the end of this document for steps to clear the FlowFlag when abrupt termination of any ISR programs occurs.

7 = Entering number 7 'Get FlowFlag' will allow the user to retrieve the current value of the FlowFlag. This value returned will be the Virtual Source's IP address should there be a Virtual Source at the time the output is displayed. (If ISR is currently not in an interval the FlowFlag will not be set.) This information is also available in the display when starting ISR_Client/ISR_WinClient.

```
FlowFlag for system psweezer ip=0.0.0.0(None), pid=0
```

The above displayed output shows that neither system has control of the FlowFlag as ip=0.0.0.0(None).

8 = Entering number 8 'Set FlowFlag' will allow the user to set the bi-directional mode controller to the cause the Virtual Source to be set. This should only be done at the direction of Professional Services implementation personnel. Manually changing this setting can cause data to become out of synchronization.

The command 8 will return a request for the following input:

SSN
PID

The PID request will be the input request for the Process ID of the ISR_Server/ISR_WinServer on the system the FlowFlag is intended to be set.

9 = Entering number 9 'Do a Command' will allow the user to input a command to execute on the current domain.

"***** Enter Command : "**. This secondary command prompt will allow the user to enter any command supported by their Operating System. A UNIX example might be to enter the command "ls." A "0" zero returned will signify the command completed successfully. Any other numeric returned means there were problems with the completion of the command. The selection of number 9 can be used in conjunction with the selection of number 11 to produce debug output to the display.

The ISR_Server or ISR_WinServer would be manually started on the domain the ISR_Client or ISR_WinClient is connected. Number 11 will toggle on debug and then number 9 can be used to send a command. This process will allow for the output of the command to be displayed on the domain where ISR_Server had been manually started.

10 = Entering number 10 'Shutdown' will stop the ISR_Server on the domain which you are connected. The ISR_Client or ISR_WinClient does not have the capability to restart the server. This must be done at a command prompt on the system where the domain is located.

11 = Selecting number 11 'Toggle Debug' will turn the ISR_Server to debug mode on the system which you are connected. This is helpful when troubleshooting communication problems between peer systems in the ISR NET. The output is only displayed to the display of the sessions that ISR_Server or ISR_WinServer was started.

Example setting ISR_Server in a UNIX environment to debug mode:

The procedure for turning on ISR_Server in debug mode is as follows:

- At a command prompt enter:
  ISR_Server
- When the command prompt returns set debug through the ISR_Client as follows:
  ISR_Client –l (to determine the current **debug** option number)
  ISR_Client
  input 11 (the current debug option)
  input a carriage return
- When the command prompt returns enter:
  ISR_start

Example setting ISR_WinServer in a Windows environment to debug mode:

The procedure for turning on ISR_WinServer in debug mode is as follows:

- At a DOS prompt enter:
  start ISR_Server
- When the command prompt in the DOS window returns, execute the ISR_WinClient and set debug mode as follows:
  ISR_WinClient
  input 11 (or whichever the **debug** choice is)
  input a carriage return
- When the command prompt returns with "`*** Enter Command : `" enter:
  ISR_start

Messages returned from **ISR_Server** or **ISR_WinServer** will now be displayed in the command window or DOS window of the domain where

ISR_Client/ISR_WinClient was connected. A zero that follows a command indicates the command was successful. Any other numeric or character following the command will indicate a command failure.

8.3 Required Script ISR_FlushLogs Customized for Site

ISR will call ISR_FlushLogs at the beginning of each Virtual Destination's interval to execute on the Virtual Source. This script or batch file when run will execute two P8-IM executables: `flush_act_log` and `flush_index_act_log`. These executables will write out the contents of all P8-IM Activity Logging Event buffers and then close logging files so that work is ready for processing.

Single-direction mode environments will see the Physical Destination alert the Physical Source to do a flush of the logs. The ISR_FlushLogs is executed on the Source, thus making P8-IM events ready for processing.

Bi-directional – the Physical (Virtual) Destination will alert the Physical (Virtual) Source to flush the logs first. After completion of that Source work on that Destination, the Physical Source (which becomes the Virtual Destination) will then request the Physical Destination (which becomes the Virtual Source) to Flush the logs so events can be duplicated onto the Physical Source (Virtual Destination).

This second half of the interval is what causes the Physical Source to become the Virtual Destination and the Physical Destination to become the Virtual Source. The completion of these steps would conclude a complete bi-directional interval.

ISR_FlushLogs – The creation of the ISR_FlushLogs script should only be done on the Source System(s) in a single-direction environment. In a bi-directional environment the ISR_FlushLogs must be created and edited for bi-directional mode on both systems in the ISR NET. The file is a script in the UNIX environments and a Batch file in the Windows environments.

IMPORTANT NOTE: The **ISR_FlushLogs.sample** file contains the correct text for both Operating Systems as well as single and bi-directional command lines. The sample file **will need to be copied** to a filename with the correct naming convention to be recognized by the ISR modules and executed. The most recent version of the **ISR_FlushLogs.sample** file was loaded on the system during the last installation

The ISR_FlushLogs should then be edited and tested to ensure it is working properly before implementing in your ISR solution.

Examples:

UNIX:

```
copy ISR_FlushLogs.sample ISR_FlushLogs
chmod 775 ISR_FlushLogs
```

Windows:

```
copy ISR_FlushLogs.sample ISR_FlushLogs.bat
```

The call to execute P8-IM Activity Log buffer flushing is executed at the beginning of an interval, even when **ALL** Image Manager buffering is set to 0 (zero). When in synchronous mode, flushing will close the current log and ready it for processing. Logs should never be removed from the logging area because doing so will cause P8-IM Events to be lost.

UNIX: (single-direction and bi-directional mode) - Copy the file ISR_FlushLogs.sample to "ISR_FlushLogs" and set the script to 775 mode. Edit it in the areas for the logging mode as well as paths to customize to your specific environment: The script sample contains examples for single-direction and bi-directional mode. Your FILENET Implementation Consultant will assist your site with the editing and testing of this script.

Windows:

In a Windows environment, copy the ISR_FlushLogs.sample to ISR_FlushLogs.bat. The single-direction mode is simple, as it contains no site-specific information and can be edited to include only the 4 lines in the sample below. The bi-directional batch file will contain more lines containing site-specific information. This information must be entered correctly, and it must be tuned your environment. The working files are critical to stop events from looping between systems.

Windows: (single-direction mode) – Edit "ISR_FlushLogs.bat" with the following content:

```
flush_act_log
flush_index_act_log
date/t > .isr_flush_log
time/t >> .isr_flush_log
```

Windows: (bi-directional mode) – Copy the sample file to "ISR_FlushLogs.bat". In bi-directional mode, there must be an ISR_FlushLogs.bat on both Physical Source and Physical Destination. These batch files will each contain system-specific information. Please see the next section that explains the required scripting for bi-directional Windows "move" or UNIX tools for DOS "mv" command execution.

8.3.1 ISR_FlushLogs Customized for bi-directional filtering

The "move" command in a Windows environment or "mv" command in UNIX are critical parts of ISR_FlushLogs in bi-directional mode. The available *.isr and *.isr_### working binary files in each system's OKpak directory are mandatory to stop Events from looping. The examples below must be scripted and tested before running in a production environment.

The Virtual Destination will execute an ISR_FlushLogs on the Virtual Source that will move the OKpak *.isr and *.isr_### files from the Virtual Source's WorkingDirectory to the its Activity Logging area.

General Example:

This example represents a typical Unix implementation.

Looking at the ISR_net.csv file will show the Physical Source and Destination of your pair. Use the P8-IM "ssn" command to verify the ssn's on each system.

Physical Source is: psmud = 11001111

Physical Destination is: psfloyd = 11001112

The Working Directory is an NFS mounted file system that both systems have access to:

/Xtra/icer

The Activity logging area for both systems is their own LOCAL /fnsw/local/logs/act directory.

The variables in the "mv" commands below are listed in the UNIX ISR_FlushLogs sample found in "UNIX Sample" section below. Make sure to set the directory variables correctly for your system's environment.

The "mv" command in the ISR_FlushLogs on the Physical Source will look like this:

```
mv -f ${wkd}/11001112/OKpak/*.isr{,_???} ${fna}/act
```

The "mv" command in the ISR_FlushLogs on the Physical Destination will look like this:

```
mv -f ${wkd}/11001111/OKpak/*.isr{,_???} ${fna}/act
```

UNIX explanation and sample:

The UNIX environment requires the following lines be added to the ISR_FlushLogs script. Make sure the "fna" and "wkd" variables match your environment.

UNIX Sample:

```
set fna="/fnsw/local/logs"
set wkd=`egrep -v "^;" ISR.cfg|fgrep -i WorkingDirectory|awk -F= '{print
$2}'`
mv -f ${wkd}/(correctssn)/OKpak/*.isr{,_???} ${fna}/act
```

Windows explanation and sample:

Windows environments have the option of using either the standard Windows "move" command or the UNIX tools for DOS "mv" command. The behavior of each

command differs. The samples below have been tested but your environment's implementation does not have to be limited to these solutions. Any implementation of this move must be tested thoroughly.

Windows Sample #1:

Using the "mv" command will require two separate command lines. One will end in a period (.) and one will not. This is because the Windows behavior using the "mv" command will differ when more than one variable is used in a command line. Since the second line contains the variables "???" the end of the line must not end in a period (.).

```
move \\hq_psverve\icer\11001104\OKpak\*.isr \\hq_psverve\act\.
move \\hq_psverve\icer\11001104\OKpak\*.isr_??? \\hq_psverve\act
```

Windows Sample #2:

Using the "move" command does not require two lines as the Windows "move" command will actually move both *.isr and *.isr_### files with this one command.

```
move d:\icer\11001104\OKpak\*.isr d:\fnsw_loc\logs\act\. (do not overwrite)
```

Windows Sample #3:

Using the "move" command can be done with a /Y to force overwrites. This is not recommended but can be done in special use cases.

```
move /Y d:\icer\11001104\OKpak\*.isr d:\fnsw_loc\logs\act\. (to overwrite)
```

8.4 Optional Script ISR_FinalProc Customized for Site

ISR_FinalProc provides a way to run a user's custom script at the end of each cycle. Similar to ISR_FlushLogs, which runs at the beginning of an interval, ISR_FinalProc runs at the end of each interval. The Virtual Destination remotely executes it on the Virtual Source. Should the script or batch file, ISR_FinalProc, not be present, no error is reported, but the NOHUP file will indicate that the file was not found. This differs from ISR_FlushLogs, which is a required file. Should the ISR_FlushLogs file be missing an error will be logged and the application will stop.

The ability to call Final Procedure(s) via the ISR_FinalProc allows the user to further customize their solution. Any usage of this functionality should be reviewed by your Professional Services System Consultant and tested before adding it to a production environment.

8.5 SecPasswd - Optional Security Object update tool

The program SecPasswd allows a site to further customize their implementation. This tool will provide a way to transfer passwords and expiration dates from the Source domain to the Destination domain. SecPasswd performs two basic functions: export and import. The P8-IM must be up when SecPasswd is executed.

For export, the tool will write object Ids (to be used as a key), encrypted passwords and expiration dates out to a user-specified text file. Then in import mode, it reads this export file, and using the security disposition in ISR_net.csv, it decides which object to update the passwords and expiration dates.

There are four Security configuration dispositions SecPasswd can find in the configuration file (ISR_net.csv). The tool will take different import actions depending upon the disposition or mode configured for Security objects.

1. Disposition = "**NO**" – no action taken

2. Disposition **= "CVT"** – The Security Object user domain to update is set to the domain name which is opposite to the domain **SecPasswd** import is being run on. The Source domain's user password and expiration are used to update the Security Object.

3. Disposition **= "KEEP"** – The Security Object user domain to update is set to the domain name which **SecPasswd** import is being run on. The Source domain's user password and expiration are used to update the Security Object.

4. Disposition **"DUP"** – The Security Object user domain to update is set to both domains in the ISR Net. The Source domain's user password and expiration are used to update the Security Object.

The import process will provide a log file entitled **secpw_update.out** at run time. This file will tell the user how many records were updated. SecPasswd identifies users by user name, not security ID, and it assumes the users exist. Should the tool try to update a user that does not exist, it will put a "Total of 0 records updated" in the **secpw_update.out** file for a line that failed.

Single direction procedure begin:

Export mode syntax is:

SecPasswd local_domain –e <export_file>

Import mode syntax is:

SecPasswd local_domain –i <import_file> (use "export_file" name created from Source export)

End of Single direction procedure

Bi-directional procedure begin:

Export mode on Source:

SecPasswd local_domain –e <export_file>

Import mode on Source:

SecPasswd local_domain –i <import_file> (use "export_file" name created from Source export)

Import mode on Destination:

SecPasswd local_domain –i <import_file> (use "export_file" name created from Source export)

End of Bi-directional procedure

IMPORTANT NOTE: Security changes should only be made from the Physical Source system. In a fail-over situation the Destination becomes the Physical Source and can accommodate Security changes. However, all SEC changes must be processed in full before switching back to the original Physical Source system when the disaster that caused the fail-over has been resolved.

8.6 Bi-directional controller

The process of bi-directional duplication causes two peer systems within the ISR NET to each process work, one domain at a time, during the same interval. This work must be regulated so that the Physical Destination begins the interval by processing any Activity Logs found on the Physical Source. When those logs have been completed, the Physical Source will process any return work or new work created that occurred on the Physical Destination. A filtering process excludes work to be done on the Physical Source from the Physical Destination that originated on the Physical Source. It is this organized process that allows each system to process work produced on its peer system.

- **FlowFlag** – is the internal variable housed in memory that directs the Physical Source and the Physical Destination as to which ISR_Engine can process work during an interval. This runtime functionality creates the situation where the Physical Source can go from being the Virtual Source to the Virtual Destination via the control of the FlowFlag. To view the FlowFlag during runtime use ISR_Client in the UNIX environment and ISR_WinClient in the Windows environment, using procedures found in the ISR_Client/ISR_WinClient section above.

Start **ISR_Client** on one of the systems in the ISR Net. Set a domain and request the current FlowFlag value. The value returned will be the TCPIP address of the system that currently has control of the FlowFlag. A null value or TCPIP address containing all zeros represents no system having control of the interval. This could be because the interval is in process and the peer systems are switching control or it could be because the current interval has passed and both systems are sleeping awaiting the next interval.

9. STARTING AND STOPPING ISR

9.1 Using the ISR_start and ISR_stop programs

Whether running in a UNIX or a Windows Server environment, the command line syntax is the same. Differences in techniques for using ISR_start and ISR_stop will be found when running via cron in UNIX or as a Service in a Windows server environment.

IMPORTANT NOTE: ISR_stop will take at least 60 seconds to stop, and in some cases, it takes as long as the active processes need to complete their current work. When stopping through an automated process, such as cron or Scheduled Tasks, it is necessary to manually test the stop time while processing the highest anticipated level of work. This time needs to be added to any automatic P8-IM shutdown method to allow ISR processes to finish first.

9.2 Executing ISR_start and ISR_stop on the command line

ISR_start must always be run on the Source system first. (In a bi-directional environment this would be referred to as the Physical Source.) Running ISR_start on the Destination (Physical Destination in a bi-directional environment) can be done within seconds of executing ISR_start on the Source system.

ISR_stop must always be run on the Destination system first. (In a bi-directional environment this would be referred to as the Physical Destination.) In a single-direction environment the Destination should be allowed to stop before initiating ISR_stop on the Source. When running in bi-directional mode the user can run ISR_stop on the Physical Source within a few seconds of executing ISR_stop on the Physical Destination. The programs will continue to run on both systems until the interval has completed all work.

9.3 Running ISR as a Windows Service

Starting and stopping ISR can be managed using Window's Services when running a Windows Server platform. This can be implemented once ISR has been installed.

1. In a DOS prompt cd to the Home directory:

   Default Home directory:
   <drive>:\fnsw_loc\bin

IMPORTANT NOTE: DO NOT set the "FileNet ISR Service" to auto start. Since its successful startup depends upon the P8-IM being fully started, an auto start is not recommended unless the service dependency is configured in the registry.

2. Enter the command below to create the FileNet ISR Service: (The command entered this way will NOT start the service, only create it.)

   ISR_start -service

   The display will report back after the Service has been installed:

   "ISR Service successfully installed"

3. On Windows - open the Control Panel and select the Services icon

   In the Services window select "FileNet ISR Service" and then select the "Startup" button

   In the "Log On As:" frame select the "This Account:" radio button

   Go to step #5

4. On Windows 2000 Server – open the Control Panel / Administrative Tools / Services icon

   In the Services Applet window select the "FileNet ISR Service" with a right mouse click and then select the "Properties" menu item

   Select the Log On tab

   Under "Log on as:" select the "This account::" radio button

   Go to step #5

5. In the "This Account" text box enter the "fnsw" user or a user in the fnusr group specifically created for running the ISR Service. Enter and Confirm the password for the account then select the "OK" button.

IMPORTANT NOTE: Starting on the Source system first is a requirement. DO NOT attempt to run ISR_start or FileNet ISR Service until you have completed the section entitled "Configuring ISR."

6. Once the login properties have been set and all configuration items have been completed, start the "FileNet ISR Service" in the Services window by right-clicking and selecting "Start" from the pull-down menu.

IMPORTANT NOTE: The Service Logon Account MUST be configured before the Service is used or the start up will fail. The default "Local System" account that is present with each newly created Service cannot be used for the FileNet ISR Service. The service must run as "fnsw" or the equivalent. The failure notice could take more than a minute to report back that account login information is incorrect.

IMPORTANT NOTE: When using the Service Applet to stop the "FileNet ISR Service" the applet may receive premature notice that ISR has stopped. However, some processes may still be shutting down and immediate restart through the applet can cause corrupted multiple processes to be run. Check the Windows Task Manager to ensure all ISR processes have stopped.

7. The following procedure will remove the "FileNet ISR Service":

   Check the Service through the Service applet window and ensure the service has been stopped.

   open a DOS window

   cd <drive>:\fnsw_loc\bin

   ISR_start -noservice

10. CONFIGURING ISR TO RUN 3RD PARTY (SINGLE-DIRECTION)

10.1 Purpose of a 3rd party environment

The purpose of a 3rd party client configuration for ISR runtime is to remove ISR/WAL integration from the Source and Destination systems. This offers an alternative method of running ISR in an ISR Net where ISR/WAL version requirements may conflict with those of other applications such as eProcess or another WAL custom application. The information below will designate the modules required on each of the systems as well as the configuration and runtime sequences necessary for ISR to operate. It is important to read through all the instructions before beginning.

IMPORTANT NOTE: The following instructions are for single-direction mode ONLY. Do not attempt to configure bi-directional mode using these instructions.

10.2 Requirements in a 3rd party environment

Source = P8-IM / ISR_Server / ISR_FlushLogs

Destination = P8-IM / ISR_Server

3rd Party system = WAL / Full ISR installation

PLUS – The Source and Destination Systems must contain all P8-IM SCRs required for ISR functionality. The 3rd Party system may require WAL SCR fixes. See ISR readme.txt for current fixes and requirements.

10.3 Configuration in a 3rd party environment

Use of the 3rd Party system environment removes WAL software as well as memory, CPU, and Disk I/O overhead from the Destination Server within the ISR NET. This solution will increase network traffic.

The steps to configure the ISR NET using a 3rd party system have been outlined below with the focus on each individual system: Source, Destination and 3rd Party system. The 3rd Party system should look, function and be configured as if it was the logical Destination Server.

10.3.1 Configuring 3rd Party System

- Install the Java Runtime software for the Operating System environment of your 3rd Party system. See the Prerequisites section for details.

- Install WAL – make sure all WAL installation instructions are followed
- Install ISR
  When running in a Windows environment using the WINDOWS Service applet, ISR should never be set to startup automatically.
- PSLSF license key file must be present on the 3rd party system
- Run ISR_Config to configure the ISR NET
- ISR.cfg domain entry MUST point to the Destination System
- Source system Activity Log file repository must be accessible. This means a UNIX NFS mount is mandatory. When running in Windows a Windows Share is mandatory. Refer to Destination instructions regarding this accessibility.
- No Image Manager is required on this system

10.3.2 Configuring Source System(s) in a 3rd party environment

- Copy ISR_FlushLogs.sample from the 3rd party server installation and then rename and modify as needed.
- Copy ISR_Server over from the 3rd party server installation if the Operating Systems are the same on all 3 systems. If the Operating System on the 3rd party system differs from the Source/Destination Operating system it will be necessary to acquire the correct Operating System compile of the ISR_Server module from Professional Services.

IMPORTANT NOTE: Windows to UNIX or UNIX to Windows is not supported as a 3rd party environment or an ISR Net environment.

10.3.3 Configuring Destination System in a 3rd party environment

- Copy ISR_Server over from the 3rd party server installation if the Operating Systems are the same on all 3 systems. If the Operating System on the 3rd party system differs from the Source/Destination Operating system it will be necessary to acquire the correct Operating System compile of the ISR_Server module from Professional Services.

IMPORTANT NOTE: Windows to UNIX or UNIX to Windows is not supported as a 3rd party environment or an ISR Net environment.

10.4 Runtime in a 3rd party environment

10.4.1 Startup in 3rd party environment

Since functionality relies on all executables within the ISR Net, it is important that the correct start order is observed. The Source and Destination ISR_Server / ISR_WinServer must be started before the 3rd party system's ISR install can be started via ISR_start.

- SOURCE and DESTINATION: Start the ISR_Server (UNIX) or ISR_WinServer (Windows) module on both the source and destination in the background using the following command:

  UNIX systems:
  ISR_Server &

  Windows systems:
  start ISR_WinServer

- 3RD PARTY: After ISR_Server is started on both the Source and Destination, start via ISR_start on the 3rd party system.

  On a command line:
  ISR_start

  In a Windows environment the command line ISR_start can be used or the Windows Services applet.

10.4.2 Shutdown in 3rd party environment

The 3rd party server should always be stopped first followed by the shutdown of the ISR_Server / ISR_WinServer on the Destination and then the Source. Currently, in a 3rd party environment, there is no "nice" automated way to stop the ISR_Server or ISR_WinServer.

First use ISR_stop to stop the programs running from the 3rd party system. Ensure enough time for processing and programs to complete before stopping the ISR_Server / ISR_WinServer on the systems replicating within the ISR Net.

Once all processing has stopped, stop the ISR_Server / ISR_WinServer on the replicating systems. The manual way of stopping the Server on the replicating systems within the ISR Net is to start the ISR_Client (UNIX) or ISR_WinClient (Windows), input the domain name of the ISR_Server / ISR_WinServer you wish to stop and entering the correct menu number at the prompt (Shutdown) to stop the process. Further

information on ISR_Client or ISR_WinClient can be found in the section entitled: **ISR Program Functionality**.

The UNIX kill command can be used in this 3rd party environment providing its execution is cleared with your FILENET Implementation Consultant. In a Windows environment the use of the Windows Task Manager to "End" the process must also be cleared with your FILENET Implementation Consultant before performing this in your environment.

11. ISR REPORTING, VALIDATION AND MAINTENANCE

ISR logs and packet files provide information to the System Administrator regarding the Duplication processes that have been requested via P8-IM Activity Logging on Source System(s). The review and cleanup of these files will be a System Administrator activity. It is suggested in the first week of ISR implementation this review of logs should be done every day. As familiarity with the environment and the ISR functionality is achieved this schedule can be moved to a weekly review.

There are four main techniques for validating successful duplication with the ISR application: counting database items, checking database consistency, spot-checking individual items through the IDM Client application, and analysis of Activity Logs and the ISR packets. Currently these methods will rely on looking at raw data, such as table row count deltas, and human auditors retrieving and verifying documents and document or security properties on the Destination against the original data on the Source. The following tools and procedures are meant to assist in validation, error resolution and maintenance on ISR Working files.

11.1 ISR_Reports

The ISR_Reports utility is a Java application that allows the System Administrator the ability to view pack and log files to determine whether logged items from a Source P8-IM are being duplicated onto the Destination System. The complexity of the ISR application does not permit a direct comparison between activity logs and ISR work done during a given time interval. The problem is that ISR filters out redundant or unnecessary commands from the activity logs for the sake of efficiency. For example: An Activity Log with 300 transactions may result in an ISR success packet of only 150 entries with no errors.

The ISR_Reports utility analyzes the ISR packet data and reports on work expected, work completed and work not completed during a specific time frame. A region to examine is the relationship of the ISR work packets that specify all the work that needed to be done and the resultant success or error packets. The success and error packets can be summed together to get a statistic that matches the original work packet delineating the work.

11.2 ISR_Vidamin

The ISR_Vidamin utility has been designed to spot-check the synchronization between two peer systems within an ISR Net. As the input file is processed, ISR_Vidamin will create two output files. The first output file shows the synchronization status of the items requested on the systems in the format of a log file. The second output file will provide the user with Event entries in an Activity Log file that can be introduced in the Source Activity logging area. This may then be used to correct objects detected as being “out of sync.” The ISR_Vidamin report and Activity Log should both be reviewed at the

conclusion of an ISR_Vidamin run. Depending upon your sites implementation, the output log containing events may be used to resync the objects detected as being out of sync. Should the introduction of the ISR_Vidamin log be the method of resync'ing the systems, the log should be introduced into the Activity Log Path. When ISR is started the Destination domain will pull and process these Events.

ISR_Vidamin should be run on the Destination system when in single-direction mode. The program will create log files to correct out-of-sync objects from the Destination perspective. When run on the Destination, the user will need to enter a domain name. The Domain name to enter would be the Source domain. When run in this manner it will allow the System Administrator to introduce any Event logs created by ISR_Vidamin into the Activity Logging area so that when ISR is started those re-sync'ing logs will be pulled and processed.

In bi-directional mode, ISR_Vidamin can be run on either the Physical Source or Physical Destination. The output Activity Logs (if created) would need to be introduced into the Activity Logging area from the perspective of the Virtual Destination.

The following sections provide for instructions on startup, runtime, and output. It is important to note that ISR_Vidamin can produce a very large amount of overhead for the P8-IM to process and therefore it is important to utilize ISR_Vidamin carefully with a focus on spot-checking.

11.2.1 Command line options

ISR_Vidamin will be run from the command line (DOS prompt in a Windows environment). Input the following command to view command line input switches.

ISR_Vidamin –h

The output from this command will be displayed as shown below:

```
hq-pearl(fnsw)/fnsw/local/bin> ISR_Vidamin -h
Usage:  ---------------- ISR_Vidamin Options ----------------
        -dom <domain>   Required - Source Domain
                        Target Domain is defined in cfg
                        two-part NCH domain, e.g. domain:org
        -tz <mins>      Mins +/- from GMT, (Default=Local)
        -out <file>     base name for report(.rpt)and
                        activity(.alog) files
                (Default = sourcedomainname)
        -in <file>      validation prescription file
                (Default = prescription)
        ----------- PS Application General Options -----------
        -h              This help message
        -h <homedir>    Home Directory
                        Location of PSLSF and ISR.cfg
        -q              Quiet mode
        -v [domain]     Display Version - Option domain
```

```
                    Provide domain to include IDMIS version
----------- Special - use only as directed -----------
-S                  Spy mode
-D                  Debug mode
-M                  Debug Memory mode
-C <mins>           Clock Timeout

Image Manager Replication(ISR) - Validation Utility
---------------------------------------------------
This is the ISR Vidamin and should only be run by
hand by Professional Services Personnel.
```

The "-dom" option is mandatory and "-in" option is only mandatory if the input file is named something other than "prescription." The "-tz" and "-out" options are optional. Implementation of optional commands will be dependent upon your site's implementation.

11.2.2 Input File Contents

The input file is a list of items you would like ISR_Vidamin to check. The list of items below can be used to compare all P8-IM objects that ISR has the ability to duplicate. Should your site be using specific object items in the duplication process or only want to check specific object items, it will be necessary to create a default "prescription" file or create a custom input file containing only the object items to be checked. The four areas that can be checked are:

- **documents** = All document information will be checked. This includes document ID existence, document index information, document security attributes, document disposition attributes and document annotations. If the input file only contains the word "documents," then ISR_Vidamin will start at the first docID on the system and do EVERY document ID found in the database. This type of checking is NOT recommended and could cause P8-IM failures during production hours.

- **security** = All Group and User information (Security Objects) will be checked.

- **folders** = All Folder information will be checked.

- **queue** = All Workspace and Queue structure information will be checked.

Below is an example of the default **prescription** file that contains all 4 object items. This example will check all document objects (image, index, annotations) as well as the security, queue and folder objects between the two peer systems.

```
documents
security
queue
folders
```

Example of a custom file with only selected object times. This example will check only one document ID (100040) as well as the security objects between the two peer systems.

```
100040 100040
security
```

Specify document ID's as well as actions:

ISR_Vidamin should never be run on the entire document ID range found on a production P8-IM server. The overhead for this work as well as the time necessary to produce this information for validation would be considered an extreme amount of overhead and out of the scope of what ISR_Vidamin was designed to provide. ISR_Vidamin utilizes WAL calls to check synchronization. These queries into the P8-IM database can be slow as well as cause the P8-IM to slow during production hours. It is recommended that ISR_Vidamin be run during non-production hours or when load levels are at a minimum.

To better spot check a document ID range the input file can be customized to check a specific number of documents within a document ID range. The user can further direct ISR_Vidamin as to what action to take on each specific item. This action would be presented as an Event line created for the item in the Activity Log output file.

Note that there are 7 columns for input when specifying Document objects with a space between each of the columns. A new column parameter will be delineated by that space. It is not necessary to use all 7 columns but it is necessary to input values for columns 1, 2, and 3 for example if you choose to enter an input value for column 4.

Column input description for Document objects:

{[docid|*] { [docid|*] { [increment|*] { {fix|ign|chk|*} {fix|ign|chk|*} {fix|ign|chk|*} {Comment} } } }

starting ID range – is the first document ID to begin checking. The **wildcard option "*"** will check start at the systems lowest document ID.

ending ID range – is the last document ID to complete checking. The **wildcard option "*"** will check to the systems highest last document ID.

increment – is the amount of documents to skip before checking the next document. The **wildcard option "*"** will default to an increment of 1.

Fix – detailed testing, checking existence and data comparison (slow)

Ign – no testing

Chk – Existence testing, but no data comparison

[fix|ign|chk] – column for Document Index information synchronization: Directs ISR_Vidamin to create an Event entry via **fix**, to ignore the disposition of the object via **ign** or to log an entry in the log file via **chk** without creating an Event log entry. The **wildcard option "*"** will be treated as **fix**.

[fix|ign|chk] – column for Document Image presence: Directs ISR_Vidamin to create an Event entry via **fix**, to ignore the disposition of the object via **ign** or to log an entry in the log file via **chk** without creating an Event log entry. The **wildcard option "*"** will be treated as **fix**.

[fix|ign|chk] – column for Document Annotation presence and synchronization: Directs ISR_Vidamin to create an Event entry via **fix**, to ignore the disposition of the object via **ign** or to log an entry in the log file via **chk** without creating an Event log entry. The **wildcard option "*"** will be treated as **fix**.

Comment – is user text information not read by ISR_Vidamin

Examples:

```
100000  599999 1 fix fix fix Full check and fix all
```

Will check every document ID from 100000 through 599999 and create an Activity line entry for all Document Object attributes that are not synchronized. This large amount of checking is not recommended and out of the functionality scope of ISR_Vidamin.

```
100000  100099 7 fix fix fix Spot check and fix all
```

Will check every 7th document ID from 100000 through 100099 and create an Activity line entry for all Document Object attributes that are not synchronized.

```
100000  600099 10 fix ign fix Spot check and fix only doc/ano
```

Will check every 10th document ID from 100000 through 600099 and create an Activity line entry for the Document Object and any annotation that are attributed to that Document Object that are not synchronized.

Your input file may contain numerous lines of document ID ranges with various selections of checking to be done.

```
1000000  1000007  1  ign  fix  fix
1000111  1000555  10  fix  fix  fix
```

Specify Security Objects as well as actions:

Security Objects can be checked in their entirety or by specifying a specific Security Group.

Sec[urity] {object|*} { {fix|ign|chk|*} {Comment} } }

object – is the object item to be checked. The **wildcard option "*"** will be treated as all objects.

[fix|ign|chk|*] – Directs ISR_Vidamin to create an Event entry via **fix**, to ignore the disposition of the object via **ign** or to log an entry in the log file via **chk** without creating an Event log entry. The **wildcard option "*"** will be treated as **fix**.

Comment – is user text information not read by ISR_Vidamin

Example:

```
sec Accounting1 fix Synchronize Accounting1
```

The above example will check the attributes and members of the Accounting1 group only. Activity log entries will be placed in the output file

of any items not synchronized between the Source and the Destination system at the time ISR_Vidamin was run.

Specify Workspace/Queue Objects as well as actions:

Workspace and Queue Objects can be checked in their entirety or by specifying a Workspace and/or Queue to check. The Workspace name or some part of the Workspace name can be entered in this field and it will automatically be wild carded.

Que[ue] {workspace|*[.queue|*] { {fix|ign|chk|*} {Comment} } }

workspace.queue – is the Workspace and/or queue to be checked: An appropriate string of characters for the Workspace can be entered and automatically wild carded by ISR_Vidamin. A queue may not be entered on its own. When checking a queue the queue name entry must be preceded by the workspace name and a period (.).The **wildcard option "*"** will check all ws_queue pairings.

[fix|ign|chk] – Directs ISR_Vidamin to create an Event entry via **fix**, to ignore the disposition of the object via **ign** or to log an entry in the log file via **chk** without creating an Event log entry. The **wildcard option "*"** will be treated as **fix**.

Comment – is user text information not read by ISR_Vidamin

Example:

```
que TestWP1.testque1 fix Synchronize the WS and Queue
que *.* fix Synchronize the WS and Queue
```

The first example will check and fix the structure and attributes of the TestWP1 workspace as well as the testque1 queue. Activity log entries will be placed in the output file of any items not synchronized between the Source and the Destination system at the time ISR_Vidamin was run. The second example will check and fix the structure and attributes of all workspace.queue pairings.

Specify Folder Objects as well as actions:

Folder Objects can be checked in their entirety or by specifying Folder tree for ISR_Vidamin to check.

Fol[der] {folder { {fix|ign|chk} {Comment} } }

folder – is the folder to be checked. The **wildcard option "*"** will be treated as all folders.

[fix|ign|chk|*] – Directs ISR_Vidamin to create an Event entry via **fix**, to ignore the disposition of the object via **ign** or to log an entry in the log file via **chk** without creating an Event log entry. The **wildcard option "*"** will be treated as **fix**.

Example:

```
folder /TestFolder1/TestFolder1a fix Synchronize Folder
folder * fix Synchronize Folder
```

The first example will check the attributes (which includes contents) of everything below TestFolder1a. TestFolder1 and TestFolder1a will NOT be checked. The second example will fix all folders.

11.2.3 Formulating a command line

ISR_Vidamin is run from a UNIX command line or a Windows DOS prompt. The user must always specify the source domain by using the –dom flag. The balance of the command line input is optional as defaults can be utilized for the input file, time zone and output file.

- **Basic Command Line Input:**

  ISR_Vidamin –dom <sourcedomain> (will use the default filename "prescription" file that you will create as your input file)

- **Command Line Input designating your own input file:**

  ISR_Vidamin –dom <sourcedomain> -in <filename> (will use the *filename* you created as the input file)

- **Command Line Input designating your own input file and your own output file naming convention:**

  ISR_Vidamin –dom <sourcedomain> -in <filename> -out <filename> (will use the *filename* you created as the input file as well as write the report log and event activities log to the output *filename* you designate)

- **Command Line Example:**

  ISR_Vidamin –dom boulders –in boulders.lst –out boulders_pebbles

  Output files generated from the above command:

  boulders_pebbles.rpt
  boulders_pebbles.alog

11.2.4 Output files and logs

The *.rpt output file contains detailed log information regarding the synchronization checking done by ISR_Vidamin. An asterisk "*" at the beginning of a line inside the *.rpt output file denotes there is an item out of sync.

As ISR_Vidamin finds an out-of-sync condition between two documents, folders, security items or workspace/queues it will create an Activity Log entry in the *.alog output file. The *.alog file can then be introduced to the Activity log area for ISR to access and use to correct the problem during the next interval or running of ISR.

***.rpt** – is the verbose listing of what ISR_Vidamin checked

***.alog** – is an Activity Log list of Events of all the things that need to be corrected to synchronize the two systems.

11.2.5 Processing Vidamin Activity Logs

After Vidamin has produced a fix-up activity log it should be visually reviewed by the system administrator so that discrepancies uncovered can be looked at for possible patterns of failure.

Once the activity log is determined to be processed it can simply be placed into the Working Directory logs directory. The Engine will pick it up at the next interval and process it just as any other activity log. There is no need to change the name or extension of the file.

IMPORTANT NOTE: When DOC is set to MSC in the ISR_net.csv file, filter automatically eliminates document create work packets. It is therefore not possible to re-create documents one the destination using ISR.

If required, the ISR_net.csv file can be changed, using ISR_Config, to change the DOC setting to 'KEEP' (the MSC security model). Restart ISR with the setting. Once the work has completed, the ISR_net.csv file should be changed back to reflect the true environment.

11.3 Pending Work

ISR processing allows for delays in P8-IM action completion. Work items will be affected when ISR attempts to perform actions on the Destination system that are dependent upon another Object's creation.

Example 1: Document committal can be slow on some systems. Logged events regarding the creation of dependent objects or actions might become available to the Destination before the Document commit has completed. In this condition, dependent actions, like creates of annotations or folder file events, would be sent to pending for processing at the beginning of the next interval or any subsequent interval while awaiting the completion of the Document commit process.

Example 2: ISR uses page_cache during the Document commit process on the Destination system. If cache is full, it will send the Document creates to a **pending** binary packet for retry at a later time. Processing will be tried again during the next interval or a subsequent interval, until the **PendErrHrs** keyword setting in the **ISR.cfg** file has been reached.

A *pending*.log is created along with the ***pending*.pkt.** The ***pending*.pkt** file is the binary file it will read from and write pending items to during each interval. The ASCII

pending.log copy is for convenience when the System Administrator is reviewing pending items. The pending files have similar naming conventions as the Activity logs. There will be groupings for FDR, FLD, IDA (INX,DOC,ANO), WQS, and SEC. The ***pending*.pkt** will be rewritten each interval as items are processed out of it and new items are added to it. The *pending*.log files are static and will remain for each pending group for each interval. This functionality allows the SysAdmin user to view pending items in each area. Review of these items can assist in performance analysis and the subsequent time setting of intervals. These files are created in the <WorkingDirectory> / <ssn> / APPpak directory.

Should an action remain in the *pending*.log for more than a day, where a day is the maximum <default> setting for a pending item to remain pending, the system will treat this action as an error by sending it to an **ERRpak**. It will be necessary for the System Administrator to perform the steps in identifying the failure. The time period an action can remain pending can be adjusted using the **PendErrHrs** keyword in the **ISR.cfg**. One hour is the minimum for an item to remain in a pending state. Values can be set to hourly time periods from 1 to 24 where 24 is one day and the maximum.

11.4 Error Detection, Recovery and Clean-up

ISR requires some manual error checking. When it is turned off, usually during system backup times, the WorkingDirectory/ERRpak folder can be checked for errors. The ERRpak contents can also be checked while it is running, but it is important to NEVER verify other folder's contents during runtime, as the structure is a Working area for processing data files.

ISR will process the available Activity Logs in a logical, orderly way. This process uses "logic and filtering" to process Activity Log Events quickly. It also uses a "process order" that will safeguard the Events during their processing. The "process order" allows it to pick up an interval's events that may have not successfully been completed because of a system failure. The following outline describes the route and processing functions an Activity Log will take:

1. Logs are taken from the Activity logging area of the Virtual Source system and placed in the WorkingDirectory/logs folder on the Virtual Destination.

2. While the initial ASCII Activity Logs remain in WorkingDirectory/logs the ISR_Builder will make APPpak/*.pkt binary packet files out of them. When this completes the WorkingDirectory/logs/* are moved to WorkingDirectory/logs/old for safe keeping. In this "old" location the original ASCII logs will not be accessed again.

3. It will then take the WorkingDirectory/APPpak binary pkt files and Filter them into WorkingDirectory/OBJpak binary files. When the process has completed the WorkingDirectory/APPpak binary files will be moved to

WorkingDirectory/APPpak/old for safe keeping. In this "old" location they will not be accessed again. Should the process from APPpak to OBJpak fail, it will utilize the APPpak binary again and reinitiate the Filtering into an OBJpak binary.

4. The WorkingDirectory/OBJpak binary pkt files are then filtered and built into WorkingDirectory/WKRpak binary files for processing. When the process has completed the WorkingDirectory/OBJpak binary pkt files will be remain in place while the WorkingDirectory/WKRpak binary pkt events are actually processed. Should any part of the WorkingDirectory/WKRpak binary pkt Events processing fail, it will recreate the WorkingDirectory/WKRpak binary pkt files to make sure all Events have been processed. When successful WorkingDirectory/WKRpak binary files have completed, ISR_Worker converts them to ASCII log files (*.**log***).

5. ERRpak

   The **ERRpak** is the Error folder in the WorkingDirectory. The files in this directory will be in two forms: 1) the binary packet (pkt) file used to process an event, and 2) the ASCII form of the same binary packet file for easy viewing.

   yyyymmddtttt.pkt.FDR – binary of folder file actions
   yyyymmddtttt.pkt.FDR.log – ASCII of the above binary with the same date/time filename
   yyyymmddtttt.pkt.FLD – binary of folder actions
   yyyymmddtttt.pkt.FLD.log – ASCII of the above binary with the same date/time filename
   yyyymmddtttt.pkt.WQS – binary of workspace or queue actions
   yyyymmddtttt.pkt.WQS.log – ASCII of the above binary with the same date/time filename
   yyyymmddtttt.pkt.SEC – binary of security actions
   yyyymmddtttt.pkt.SEC.log – ASCII of the above binary with the same date/time filename
   yyyymmddtttt.pkt.IDA00 – binary of index, annotation or document actions
   yyyymmddtttt.pkt.IDA00.log - ASCII of the above binary with the same date/time filename

   View the ASCII file to determine what failed. Review of the eng* and/or wkr* journal log as well as the /tmp/ISR_Monitor.yyyymmdd nohup file when detecting the reason for a failure. Verify the cause of the failure and correct the problem if possible. If the problem has been resolved the System Administrator can execute the ISR_Worker using the binary format ERRpak file as the input file to complete the work. When the work has been completed successfully the ERRpak files associated with that work can be deleted. The ERRpak binary pkt can also be re-introduced at the WorkingDirectory/APPpak level where it will be picked up at the beginning of the next interval.

   Should the error be verified as a logical failure that cannot be corrected, do not rerun the log file through the system. Save the binary log and it's companion

ASCII log file to a repository for your site's environment should you want to keep a record of the accepted failure or delete the files as they will not be utilized.

To rerun a binary Error log packet, go to the installation directory and on a command line type:

In UNIX:
cd /fnsw/local/bin
ISR_Worker –dom *sourcedomainname* –in /*path*/*to*/ERRpak/*yyyymmddtttt*.pkt.*filename*

In Windows :
cd <drive>:\fnsw_loc\bin
ISR_Worker –dom *sourcedomainname* –in <drive>:*path**to*\ERRpak*yyyymmddtttt*.pkt.*filename*

11.4.1 WKRpak

The **WKRpak** is the last place a binary packet will visit. This folder may be checked for integrity. The top level of this folder (i.e. /WorkingDirectory/ssn/WKRpak) should not contain any files after an ISR_stop. All binary working files will be converted to ASCII log files placed in the WKRpak/old directory

Should any binary log files be found in WKRpak, it will be necessary to verify the work in the packet was completed successfully. Use ISR_Builder as outlined below to view the contents of the binary packet. Once the action in the binary packet has been read and understood it will be necessary to verify the work was completed. When the work has been verified as being completed the corresponding packet file can be removed.

If the work was not completed, it may become necessary to rerun the binary packet file through ISR_Worker as shown in the step above. This should only be done if the **APPpak** and **OBJpak** folders are clean. If the **APPpak** and **OBJpak** folders contain any binary packet files above their "old" directory the recovery functionality will take place causing the available packet files to be run through the system again.

To view the contents of a binary WKRpak log packet: Change directory to the Home directory and on a command line issue the following command:

In UNIX:
cd /fnsw/local/bin
ISR_Builder –dom domainname –ssn <SSN> –in /*path*/*to*/WKRpak/*yyyymmddtttt*.pkt.

In Windows :
cd <drive>:\fnsw_loc\bin
ISR_Builder –dom domainname –ssn <SSN> –in <drive>:*path**to**WKR*pak*yyyymmddtttt*.pkt

11.5 Cleaning ISR Data and Working files

When ISR is working, it creates log files in the journals directory and data working files in the WorkingDirectory. Currently, ISR.cfg has keyword named "CleanUpHrs"

to assist in cleaning up data files. The default is for this functionality to be turned OFF. Please refer to the section which covers "CleanUpHrs" for more information on this configurable keyword.

The WorkingDirectory consists of the following structure:

- **APPpak** – This directory will contain pending*.log files whether ISR is running or not and binary packets files during runtime only. There will be two types of binary *pkt files. When not running the 200*.pkt* binary packet files should only be present if there has been a ISR system failure. These binary pkt files will be picked up when it is restarted. The pending*.pkt files can be present whether ISR is running or not and should never be deleted as the Engine will send them to ERRpak, should it not be able to process the Event. (See the section on pending functionality for more details.) The pending*.log files can be reviewed for performance and pending work reasons. They can be deleted when pending work has completed and after appropriate review has taken place. NEVER delete ANY binary *.pkt files from this directory without proper review as work can be lost.
- **OBJpak** – This directory will contain binary packets only during runtime. The binary packets should only be present if there has been a ISR system failure. These binary pkt files will be picked up when it is restarted. NEVER delete ANY binary *.pkt files from this directory without proper review as work can be lost.

- **WKRpak** – This directory will contain binary packets being processed and ASCII ***.log.*** files of completed work. Should any binary pkt files be present when it is not running, it will be necessary to review these files to verify the work has been completed. After review and resolution to any problems regarding the binary packet file, the file can then be deleted, as it will not be picked up on restart.
- **OKpak** – This directory will contain ASCII log files of all the work successfully completed.
- **NOpak** – A NOpak "dir_create" packet is created as a result of WKRpak (correctly) processing an IGNORE. This is not an error, and periodically the contents of this directory can be deleted.
- **ERRpak** – These files should be reviewed and problems corrected before they are reintroduced into the system via the Error recovery directions found in this document. Once review and resolution is complete, the files can be deleted.
- **Logs** – This directory will contain ASCII log files that have not been processed. ASCII logs files in this directory should NEVER be deleted as work can be lost.
- **APPpak/old** – can be cleaned by CleanUpHrs keyword in the ISR.cfg
- **OBJpak/old** – can be cleaned by CleanUpHrs keyword in the ISR.cfg
- **WKRpak/old** – can be cleaned by CleanUpHrs keyword in the ISR.cfg
- **logs/old** – This directory will contain ASCII log files that have been processed. Once the ISR NET pair has been verified as synchronized, the logs dated before the date of the sync check can be deleted.

Any *.dbg files found in any directory structure are output files from ISR_start being run in debug mode. These files can be deleted when the problem requiring the debug run has been resolved.

11.6 System Sync Validation

Further validation of the existence of P8-IM items and/or updates on a Destination system may become necessary for various internal reasons. Checking the log files using ISR_Reports and occasional spot-checking through one of the IDM Client interfaces can provide information to validate data. However, should a lower level of integrity checking be necessary, the following P8-IM tools can be used to provide data checking information.

Using ixdb_stat to verify document, folder, index and WQS counts:

ixdb_stat – This P8-IM system tool can create a report on document, folder, Document Class, Index and WQS activity on a server. It can create one report based on the entire Index Database. It gives general information about documents: total number, document ID range, archive, and deletion statistics. For folders, it gives the total number of folders, documents filed, archive, deletion and closed statistics. It lists all the document classes, number of documents in each class, and index value counts. Index information includes the count of "Non-Null Values" for each Index field as well as how many documents house the "Non-Null Values" in their Index properties. Finally it gives simple WorkFlo Queue data including the number of queues and number of rows in each queue.

Running **ixdb_stat** prior to running ISR for the first time can provide for high-level statistical comparisons. This data can be saved to create a baseline of the systems within the ISR NET. Periodic runs of **ixdb_stat** can be done, once it has been running for a period of time and compared to the established baseline data. The direct comparison of the changes in recorded P8-IM database areas will be a significant indicator of the its success rates.

Note: The P8-IM utilities should be run after hours as some can cause a performance slowdown on the system if run during production hours.

Run in a DOS prompt on Windows server:

<drive>: ixdb_stat –a

Run via a command line entry in the UNIX environment:

> **ixdb_stat -a**

Note: The output will not be displayed to the screen. It is directed automatically to a file. The output file location will be displayed to the screen for future reference and access. The default output are on your P8-IM is:

UNIX = /fnsw/local/logs/inx_logs/ixdbs_*

Windows = <drive>:\fnsw_loc\logs\inx_logs\ixdbs_*

Using MKF_tool to verify document and annotation counts:

MKF_tool – This P8-IM tool provides for informational output regarding amount of documents in the document table as well as the amount of annotations attributed to a specific document.

Using the count command the user can quickly receive the total number of documents or annotations in the database.

Run in a DOS prompt on Windows server:

<drive>: MKF_tool

<MKF_tool> count docs *

<MKF_tool> count annotations *

Run via a command line entry in the UNIX environment:

> **MKF_tool**

<MKF_tool> count docs *

<MKF_tool> count annotations *

Note: Output will be displayed to the screen. No output file will be created.

Using dbverify to verify database consistency:

dbverify – This P8-IM system tool can be used to validate the consistency of the permanent (locator) database. It makes a set of three report files that list inconsistencies among the permanent database, index database, and the optical disk contents.

Using **dbverify** within the ISR NET can explain missing data on the Destination domain. For example, if a document exists in the permanent database but has no corresponding entries in the index database, the document's indexing information cannot be duplicated on the target domain. One approach would be to fix all the reported inconsistencies on the Destination domain before running ISR_start, and then monitoring source database consistency during the course of duplication.

Run in a DOS prompt on Windows server:

<drive>: dbverify

Run via a command line entry in the UNIX environment:

> **dbverify**

Note: The user will be asked to enter the following information as input values in order for dbverify to run. The time it takes to receive the output will depend upon the range of document IDs **dbverify** has been asked to report on. Output will be displayed to the screen as well as to the 3 report files listed to the display.

Enter starting document number or 'h' for help:

Enter ending document number or 'h' for help:

Identifies document missing in optical disk <y/n/h>:

Using Application Executive to check Security objects:

Application Executive – The Application Executive (Xapex) can be used to run a Summary Report for Users and Groups. The summaries will give a total row count (number of users or groups) and also list the user, or group names.

On Windows server:

Select – **Programs / FileNet IDM Image Manager / Server Applications / Application Executive**

Login to the Application Executive

Once logged in select – **Applications / Security Administration**

In the Security Administration window either **Users** or **Groups** can be selected along with **View Summary**

In a UNIX environment run **Xapex** via a command line entry

Login to the Application Executive

Once logged in select – **Applications / Security Administration**

In the Security Administration window either **Users** or **Groups** can be selected along with **View Summary**

12. FILTERING

ISR_Filter controls what, how and when events are processed. A logical order of events to be processed is created for proper object creation and well as enhanced performance. The below information pertains to single-direction mode ONLY.

12.1 Document creates

When running ISR for Doc CREATES:
doc create and dir create events are logged for the one action
The DOC Activity log -> doc create
The INX Activity log -> dir create
ISR filter will set the following actions:
PROCESS doc create is set in the OBJpak binary packet
IGNORED dir create is set in the OBJpak binary packet

This same process as above will take place when the following action and events are logged:
PROCESS doc create is set in the OBJpak binary packet
IGNORED dir import is set in the OBJpak binary packet
The user will see a doc create followed by the dir import instead of dir create when COLD or Fast Batch entry mechanisms are used to create documents. If a doc create is present then the dir import will always be set to IGNORED

When running MSC for Doc CREATES:
Solution #1:
Document create Activity logging will be turned OFF
The INX Activity log -> dir create
PROCESS dir create is set in the OBJpak binary packet
BUT the Engine will not process the dir create as the Engine has been coded to never process a dir create.

Should COLD or Fast Batch produce a dir import then they will be processed as there is no doc create for Filter to read and Filter them out.

Solution #2:
Document create Activity logging will be turned ON
Set the DOC mode through ISR_Config to “MSC”
Both DOC create and dir create or dir import associated with a DOC create will be set to IGNORE

When running Report Manager Cross Cataloging:
only a dir import will be logged
ALL dir imports that do not have an associated doc create will be processed

13. TROUBLESHOOTING

13.1 Procedure for Reporting Problems

When a problem with the functionality of a Professional Services Solution is perceived, a sample run with the minimum amount of data needs to be executed. Adding the debug option '-D', the spy option '-S' and/or the memory option '-M' may be done when requested. These debugging messages will be added to the journal log file. This file along with a description of the event that presented itself should be sent to the appropriate support individuals. Failure to provide all output files will delay the ability to diagnose the problem.

Currently FILENET Customer Service and Support (CSS) will receive calls regarding ISR problems. However, they will only open the case and log initial report data. The case will then be moved to Professional Services for the appropriate support personnel to handle. When calling CSS please identify your case as a FILENET Professional Services Solution named Multi System Synchronization.

1. Describe the behavior along with any commands given that received a failure in writing. Make sure to give a step-by-step outline along with the description.
2. System Configuration
    a. Operating System platform and version
    b. Database platform and version
    c. Run – fn_util whichfn – supply output to Professional Services
    d. IS version and stamp /fnsw/lib/shobj/SysV
    e. WAL version
    f. WAL stamp /fnsw/client/shobj/SysV
    g. Directory listing of WAL installation point to verify permissions
    h. (ls –alR /fnsw/client > *filename*)
    i. Is WAL/Utility installed directly on P8-IM or on a remote system? If remote, supply the Operating System for remote environment.
    j. PS Utility version (i.e. ISR_start –v)
    k. PS Utility stamp on UNIX systems only (i.e. stamp ISR_*)
    l. Directory listing of PS Utility installation point to verify permissions (ls –al /fnsw/local/bin > *filename*)
    m. Single or Bi-directional mode
3. Clean out or rename the journal files in the journal directory

    /fnsw/local/bin/journals/*
    <drive>:\fnsw_loc\bin\journals*

4. Tar of the WorkingDirectory
5. Export Security Database (SEC_tool)
6. ISR.cfg and ISR_net.csv (configuration files)
7. All journal files

Files to send (all that apply): Please see Error logging section above that explains the log file as well as default paths.

1. Nohup files – located in /tmp
2. Journal logs - located under the Home directory in the journals directory unless configured in the ISR.cfg file to reside in an alternate area
3. Configuration files on Source – act_log_conf file used for P8-IM logging activity
4. Configuration files on Source and Destination – ISR.cfg & ISR_net.csv (located in the ISR home directory)
5. WAL Error logs
6. P8-IM Error Logs

13.2 Utility / Operating System and WAL version relationship

The ISR version selected depends on the Operating System and the WAL version used. The utility has been compiled to run with all supported Operating Systems and WAL versions. The installation CD automatically detects the Operating System and the version of WAL in a UNIX environment. The UNIX installer will select the correct compile of the selected Utility for installation on the UNIX system. When installing in a Windows environment the user will be responsible for knowing the WAL version and making that selection when asked during the setup procedure. Should the user need to download a version of a Utility off the CSS download site, it will be necessary to verify the Operating System version level and the WAL version before selecting a download. The Operating System can be verified by running fn_util and the WAL version can be verified by running a stamp command on a WAL module in the WAL install directory. Examples of both procedures can be found below.

Example – verify the OS version level:

UNIX system:
>fn_util whichfn
>stamp /fnsw/lib/shobj/*SysV*

WINDOWS system:
fn_util whichfn
stamp <drive>:\fnsw\lib\shobj\sysv.dll

Example – stamp a WAL module:

UNIX system:
stamp /fnsw/client/shobj/*SysV*

WINDOWS system:
stamp <drive>:\fnsw\client\shobj\wal_sysv.dll

The "system" number and "rel_type" are both important informational items to take into account when selecting the correct utility version.

13.3 Error Logging Files

Errors that may occur during the operation of a Professional Services Utility can be reported in several places, depending on the error type. When an Image Service related error is encountered, the FILENET error tuple and error text, if available, will be reported. If a Utility or File System error is encountered, a Utility error code will be reported. The Utility error codes are generally related to an incorrect configuration, bad input file format, or a problem with the file system. File system errors (Code 1024) will almost always be the responsibility of the System Administrator to resolve.

The following is a list of error logging file names and locations related to the utility.

Note: Viewing log files when ISR is running in a Windows environment can be limited because of file access conflicts. It will be necessary to stop the ISR software or use the Windows "more" command in a DOS prompt. An example of how to view a ISR_Engine log file is below:

open a DOS prompt
cd <drive>:\fnsw_loc\bin\journals
more < eng########

1. Nohup logs – all standard out / standard error messages (system errors, preamble and post amble logging from the utility output generally captured via standard out that would be suppressed when using –q for quiet mode, etc)

   Default UNIX - /tmp/{product}.*YYYYMMDD*
   Default WINDOWS – WINDOWS Event logs / Applications and System

2. P8-IM error logs – all P8-IM related messages and errors (OSAR, database, security, etc.) Applicable to ALL utilities.

   Default UNIX - /fnsw/local/logs/elogs
   Default WINDOWS - <drive>:\fnsw_loc\logs\elogs

3. WAL error logs – any internal WAL error (an P8-IM error log specifically for the WAL environment) Applicable to ALL utilities.

   Default UNIX - /fnsw/client/logs/*walYYYYMMDD*
   Default WINDOWS - <drive>:\fnsw\client\logs*walYYYYMMDD*

4. Utility error logs – any batch progress and general error messages are reported in the journal files

   Journal files will be entitled as follows:
   mon*yyyymmdd*
   eng*yyyymmdd*
   wkr*yyyymmdd*
   cfg*yyyymmdd*
   bld*yyyymmdd*

   Journal log files will be found in the journal directory or parent directory (depending on the path set in the **ISR.cfg** file)

   Default UNIX - /fnsw/local/bin/journals
   Default WINDOWS - <drive>:\fnsw_loc\bin\journals

5. Should Source System become unavailable the Destination system's engyyyymmdd journal log could report some or all of the following errors:
   Can't execute **ISR_FlushLogs** on '*SourceSystem*'
   Can't move ISR_logs from '*ActivityLogPath*' to '*WorkingDirectoryPath*'
   ISR_Builder execution failed
   ISR_Filter execution failed
   Err <0,11,1024>
   Errno=11 [Resource temporarily unavailable]

13.4 Monitor fails to run

– Login problem

The most common problem causing ISR_Monitor is failure to logon to the IS. Review the journal log entry or the output to the display if running ISR from a command prompt. Run CheckSSN to verify that you can logon to the P8-IM with WAL. Verify that the user logging in has enough logins configured in Xapex.

ISR_Monitor expects to find ISR_Server (ISR RPC server) to be running on both the Source and Destination before starting ISR_Engine. If there is any problem with the RPC Service the Monitor will not run. Use ISR_Client (ISR_WinClient on Windows) to check if RPC is working correctly.

– Corrupted PSLSF file

Monitor will not run if the PSLSF file is missing or corrupted. The way to detect this is to start the ISR_Server and ISR_Monitor independently at a command prompt. The output to the display may give you more information than found in the ISR_start display or the journal log output. This is also true even when running ISR in debug mode.

Start the ISR_Server and ISR_Monitor using the following instructions for your OS:

UNIX:

> cd /fnsw/local/bin
> start ISR_Server
> ISR_Monitor

Windows:

cd <drive>:\fnsw_loc\bin
start ISR_WinServer
start ISR_Monitor

– ISR_net.csv could be out of sync

Created by the ISR_Config, the ISR_net.csv file can be hand edited. This is not a supported practice, but since the file is in ASCII form it can be done. If the base line numeric, the first field in the second line, gets out of sync with the configured lines numeric, then ISR_Monitor may hang or not start, and an error may or may not be logged.

The base line numeric is in the second full line in the file example below. This is followed by "n" number of configured lines. The amount of configured lines will depend upon the amount of Source machines in your ISR NET. In the example below, the base line shows a beginning numeric of "2". ISR_Monitor will look for two lines defining two pairs of ISR systems. If the number of lines is not two, ISR_Monitor will fail.

```
Count,Domains,,Method,Interval,Sub-Systems,,,,,,ActivityLog,WQS,WQS,Comment
2,Source,Target,,,Doc, . . .
1,pearl,boulders,Duplication,30,CVT, . . .
2,pebbles,boulders,Duplication,5,CVT, . . .
```

13.5 ISR_Engine improper shutdown

The ISR_Engine can end on error or get in a confused state when a system unit looses power or the P8-IM is shutdown without first shutting down using an ISR_stop. In bi-directional processing this could cause the FlowFlag to be left in a state that it would be set incorrectly for a restart. To reset the FlowFlag use the following steps.

- shutdown both Source and Destination using ISR_stop
- Start ISR_Client
- Enter the source domain
- Select "clear flow-flag"
- Enter the process ID of ISR_Engine on the Destination
- Exit ISR_Client
- Run wal_purge on both Source and Destination
- Restart ISR using ISR_start on Source and Destination

13.6 Problems running ISR_Config

Most problems encountered reflect a missing jre or java executable or the path is not set pointing to the jre or java executable in the environment you are using.

When working remotely in a UNIX environment, the user must export the DISPLAY and set the xhost + capability. Sometimes merely setting the DISPLAY to your local machine name is not adequate. You may need to use your system tcpip address when setting the DISPLAY variable.

When the ISR_Config executable ends with an error in the display or simply hangs it will be necessary to review the /fnsw/local/bin/journals/cfg* file on UNIX or the <drive>:\fnsw_loc\bin\journals\cfg* file on Windows for input as to why the Java application could not run.

Examples of Display output and error found in journal cfg* log file:

```
seawolf(fnsw)/fnsw/local/bin> ISR_Config
    Beginning ISR_Config Program
    Setting Default 'cfg' log file
    Starting 'cfg' Journal Log
    File is './journals/cfg20011108'
        Start Time: 2001/11/08 @ 10:42:37
        Options   : Home is '.'
                Verbose Mode, Debug Off, Spy Off, MemDbg Off
    Preliminaries Completed
    Working Directory is '/icer'
    Config File Processed - No Errors
        Platform        SunOS
        System Shell    /bin/ksh
        PS Lib Vers     2.4.0
        ISR Pgm Vers    2.1.3
        WAL Release     3.6.10
        IDMIS Release   3.6.30 (seawolf:FileNet)
    Starting ISR_Config
    Domain 'seawolf:FileNet'    Display '10.1.246.84:0'
    Total Time = 0 Secs
    ISR_Config has terminated due to error

seawolf(fnsw)/fnsw/local/bin> less j*/cf*
    2004/09/16 12:23:53 <fnsw> (013762)
    Starting 'cfg' Journal Log ./journals/cfg20040916
    2004/09/16 12:23:53 <fnsw> (013762)
    IMS Logon to psmud:FileNet completed - Getting SSN
    2004/09/16 12:23:53 <fnsw> (013762)  Platform       SunOS
    2004/09/16 12:23:53 <fnsw> (013762)  System Shell   /bin/ksh
    2004/09/16 12:23:53 <fnsw> (013762)  PS Lib Vers    2.4.0
    2004/09/16 12:23:53 <fnsw> (013762)  ISR Pgm Vers   2.1.3
    2004/09/16 12:23:53 <fnsw> (013762)  WAL Release    3.6.10
    2001/11/08 10:42:37 <fnsw> (013762) Starting ISR_Config
    2001/11/08 10:42:37 <fnsw> (013762)
    Domain 'seawolf:FileNet'    Display '10.1.246.84:0'
--- 2001/11/08 10:42:37 <fnsw> (013762) Err <0,0,1024>
    Can't locate Java Interpreter (jre or java).  Not in command PATH.
    ISR Err: Can't exec command
    2001/11/08 10:42:37 <fnsw> (013762) Ending ISR_Config
    Total Time = 0 Secs
    2001/11/08 10:42:37 <fnsw> (013762)
    ISR_Config has terminated due to error
```

Should you receive the errors listed above when trying to run ISR_Config, try to resolve the problem listed in the journals cfg* file. If the information in the log file is unclear you can run one of the following commands when you are located in the ISR home directory to further detect the reason(s) for the Java application failure.

jre -cp ISR_cfg.zip ISRcfg

or:

java -cp ISR_cfg.zip ISRcfg

If both commands result in a file not found error, check that you have Java installed and that the java or jre commands are in the command path. If there is a Java class not found error, make sure that the directory containing Java Runtime Library (rt.jar) is in the CLASSPATH variable.

IMPORTANT NOTE: The Java executables required input is case sensitive even on Windows.

13.7 Source log files not picked up in an interval

When the P8-IM Activity logs on a Source system are not picked up for processing there are a number of configurable items to check:

1. The time clocks could be out of sync. Please see the section entitled "Operating System Date / Time Properties" for more information.

2. The UNIX mounted file system or the Windows shared drive may have become inaccessible.

3. File system permissions may have changed causing permission problems.

13.8 Full File System limitation

P8-IM Activity logging creates logs in /fnsw/local/logs/act by default. Should logs continue to accumulate and not be processed, the /fnsw or /fnsw/local file systems could run out of space. When this happens, the full file system situation can cause both source and destination P8-IM systems to hang without clear log messages as to what the problem is. The majority of the symptoms (SEC_tool, CSM_tool, CheckSSN failures) will most likely be found on the source, which could result in the full /fnsw file system on the destination.

13.9 When to use wal_purge – UNIX ONLY

The WAL program /fnsw/client/bin/wal_purge can clean up numerous problem but it can also be used incorrectly when WAL programs are running and cause application/program failures. Only run wal_purge when ALL WAL applications are stopped. This can be done via a script that will run immediately after bringing up the FILENET software but before starting any WAL applications/programs.

The following cases are times when wal_purge can be used to clean up an application problem.

1. If the FILENET software is shut down before the ISR utility stops.

2. If any changes are made to the ISR.cfg file.

3. If a newly created doc class or index is not recognized. A wal_purge is necessary in conjunction with the P8-IM requirement dictating a recycle of the P8-IM software whenever a document class or index is changed through Database Maintenance.

4. Various security problems.

13.10 Journal Log Error Entries

The ISR Journal logs designate errors by the preceding 3 dashes "---". These errors can direct the user as to why a specific Activity Logged Event failed. A failure notice in the Journal logs is usually accompanied by an ERRpak log being created. The following cases are known events that will create an ERRpak as well as an eng* or wkr* Journal log error entry.

13.10.1 Folder Security Attributes

Security permissions on a folder can present a situation where ISR cannot get the folder attributes or access write permissions on a folder. When attempting to locate a document involved in this scenario, remember to check the root level for the document. Also, if possible, correct the folder's attribute or write permissions that may have caused the condition.

13.10.2 Document / Folder Filing

Filing documents in folders can be delayed because of P8-IM overhead. When this occurs, the "folder file" log event will be sent to pending, to give the P8-IM enough time to commit the document. However, a new folder created with documents added in the same interval or a renamed folder containing documents will cause the P8-IM to attempt to file documents on the Destination system as it found them on the Source system. This P8-IM functionality may produce an error to the journal log Please note this is not an ISR function but an P8-IM folder function on a folder create or rename.

P8-IM logging will still create a "folder file" event for ISR to process. Since ISR processes events in a logical order, the P8-IM logged "folder file" event will be sorted to a logical order following the "document create." Should ISR find it cannot process the "folder file" event, the action will be sent to pending in an effort to give the P8-IM time to commit necessary documents. The warning in the journal log tells the user to check that P8-IM committal is functioning and not waiting for optical platters or network access for MSC committal to take place. Should all committal processes be functioning, this message can be ignored.

13.10.3 Folder Renaming *(bi-directional only)*

ISR can handle the process of renaming a folder hierarchy. However, if a folder is renamed more than once in the same interval, the outcome may not be complete. The new folder structure and all filings will be done but the original folder may remain. An error may be logged in the ISR journal logs as well as an ERRpak log could be created in the WorkingDirectory structure.

If the wkr* journal log reports an error it may become necessary for the System Administrator to check the folder structure and manually delete the original folder structure.

Two ways to improve behavior when renaming folders:

1. Set the LoadLimit=1 so that only one hour of Activity Logs are processed at a time.
2. Verify one series of renames executed through your client interface before executing another series of renames.

13.10.4 Missing SCRs for Annotation create support

When the SCRs supporting new annotation creation in 2.1.x are missing from the ISR Net systems the Engine will shutdown and the following errors will be seen in the journals/eng* or journals/wkr* output. Resolution will require P8-IM SCR updates.

```
->DOC_get_and_lock_annotation-OverRide=False@14:16:25
<-DOC_get_and_lock_annotation-OverRide=False@14:16:25
     (call @../../../ISR/src/isr_worker.c:2723) err=80,0,21, errno=0
->DOC_create_annotation_id-Doc=100140, pg=1, id=1100206705.1@14:16:25
```

/tmp/ISR_Monitor (nohup file output)

File Mode /export/home/fnsw/ISR/1100106282/WKRpak/2003070810.pkt.IDA00
ld.so.1: ./ISR_Engine: fatal: relocation error: file ./ISR_Engine: symbol DOC_create_annotation_id: referenced symbol not found
ISR_Engine(19707) for IPS_TEST ==> IPS_DEV - Shutdown

13.11 Procedure for creating/deleting Security Objects

When creating a User or Group on the Source, it is important to know what mode (DUP, KEEP, CVT) of security duplication is configured on the Destination system. There are assumptions made in the duplication process, depending upon the mode setting. These assumptions enhance the duplication process for Security Objects in case a switchover should need to take place.

In bi-directional mode, the Security Object duplication mode will always be DUP.

Creation or deletion of a User or Group must occur on the Source system only. When in bi-directional mode, the creation or deletion of the User or Group should always occur on the User or Group's HOME system. When adding Users and Groups, each object is duplicated automatically for both systems.

13.11.1 ISR Security Object mode set to DUP (single-direction)

The System Administrator should only make one Group or User on the Source domain, as the ISR DUP mode will create the other object for the Destination domain automatically. Should this object be a group, ISR will also automatically make the Groups members of each other.

Example of what will be created on the Source system after an interval is complete:

User1:domainA
Group1:domainA

Example of what will be created on the Destination system after an interval is complete:

User1:domainA
User1:domainB
Group1:domainA
Group1:domainB

Extended Membership – Group's Information: (on the Destination – domainB)

Group1:domainB:FileNet->Group1:domainA:FileNet
Group1:domainA:FileNet->Group1:domainB:FileNet

Example of what will be created on the Destination system: If on the Source, user1 is made a member of group1.

Extended Membership – User Information: (on domainB)

User1:domainB:FileNet->Group1:domainA:FileNet->Group1:domainB:FileNet
User1:domainB:FileNet->Group1:domainB:FileNet->Group1:domainA:FileNet

User1:domainA:FileNet->Group1:domainA:FileNet->Group1:domainB:FileNet
User1:domainA:FileNet->Group1:domainB:FileNet->Group1:domainA:FileNet

The deletion of a User or Group should only be done on the Source or Home system. It will automatically remove both the Source user/group and the Destination user/group from the Destination system, as part of the work processed during the interval the delete user/group is received.

13.11.2 ISR Security Object mode set to DUP (bi-directional)

When running in bi-directional mode the Event will loop back to the Source system creating the user/group of the peer domain on the Source. This

functionality creates 2 users/groups on both Source and Destination systems. Should this object be a group, ISR will also automatically make the Groups members of each other.

Example of what will be created on both Physical Source and Physical Destination after 2 intervals are completed:

User1:domainA
User1:domainB
Group1:domainA
Group1:domainB

Extended Membership – Group's Information: (on domainA)

Group1:domainA:FileNet->Group1:domainB:FileNet
Group1:domainB:FileNet->Group1:domainA:FileNet

Extended Membership – Group's Information: (on domainB)

Group1:domainB:FileNet->Group1:domainA:FileNet
Group1:domainA:FileNet->Group1:domainB:FileNet

Extended Membership – User Information: (on domainA)

User1:domainA:FileNet->Group1:domainA:FileNet->Group1:domainB:FileNet
User1:domainA:FileNet->Group1:domainB:FileNet->Group1:domainA:FileNet

User1:domainB:FileNet->Group1:domainB:FileNet->Group1:domainA:FileNet
User1:domainB:FileNet->Group1:domainA:FileNet->Group1:domainB:FileNet

Extended Membership – User Information: (on domainB)

User1:domainB:FileNet->Group1:domainA:FileNet->Group1:domainB:FileNet
User1:domainB:FileNet->Group1:domainB:FileNet->Group1:domainA:FileNet

User1:domainA:FileNet->Group1:domainA:FileNet->Group1:domainB:FileNet
User1:domainA:FileNet->Group1:domainB:FileNet->Group1:domainA:FileNet

In bi-directional mode, Deletion of a User or Group should be done on the Physical Source or Home system of the user/group. ISR will automatically remove both the Source user/group and the Destination user/group from the Destination system as part of the work processed during the interval the delete user/group is received. Then, ISR will loop back to the Source system deleting the user/group of the peer domain on the Source.

Example:

domainA contains
User1:domainA – (ISR will delete this object)
User1:domainB – (ISR will delete this object)
domainB contains:
User1:domainA – (ISR will delete this object)
User1:domainB – (ISR will delete this object)

IMPORTANT NOTE: Security changes should only be made from the Physical Source system. In a fail-over situation the Destination becomes the Physical Source and can accommodate Security changes. However, all SEC changes must be processed in full before switching back to the original Physical Source system when the disaster that caused the fail-over has been resolved.

13.12 Multi System Committal (MSC)

13.12.1 MSC Functional Limitation

When using MSC for document transfer, it is important to insure all Document Classes are set to "migrate" when an OSAR is configured on the system. If this is not done, the documents attributed to the "no migrate" Document Classes will not be transferred to the Destination system.

MSC does work in an OSARLESS environment, but for that to work an OSAR cannot be configured on the system.

13.12.2 MSC Document Commit problems on the Destination

If Multi System Commit fails to import Fast Batches from the Source, the P8-IM logs cache errors in its error logs (elog). In this case, please call FILENET CSS for MSC support.

```
2003/02/27 11:53:22.523 151,0,22 <fnsw> fbc_commit  (19812.2312
0x4d64.908) ...
INXl_import: first error at record 0

2003/02/27 11:53:22.523 64,0,1002 <fnsw> fbc_commit  (19812.2312
0x4d64.908) ...
DIA: Can't update ODT

2003/02/27 11:53:22.523 64,0,1002 <fnsw> fbc_commit  (19812.2312
0x4d64.908) ...
fbc_commit: Error importing fast batch 4160000006 (Run
DOC_dumpfb)

2003/02/27 11:53:22.570 151,0,22 <fnsw> fbc_commit  (19812.2312
0x4d64.908) ...
INXl_import: first error at record 0
```

13.13 Content Federation Services & Non-Cataloged Documents

MSS is not capable of processing documents processed by CFS or that do not have an associated Index Record (Non-cataloged documents).

13.14 Error Codes

This table is a general list of errors that may occur when using any of the Professional Services tools or utilities. Included is a description of their meaning where appropriate. These error code numbers can change in subsequent builds that may not be reflected in this document.

ERROR NUMBER	ERROR TEXT	ERROR INFORMATION
999	lowest error – empty	
1000	Incorrect number of cfg argument	
1001	cfg bad key word	
1002	cfg exceed max	
1003	cfg expect brace	
1004	Cfg file is empty	
1005	cfg get failed	
1006	cfg invalid index type	
1007	cfg invalid keyword	
1008	cfg invalid number	
1009	cfg too many token	
1010	cfg no 2nd quote	
1011	cfg Required Directory missing from Configuration File	
1012	cfg not digit	
1013	Value for Keyword is not valid	
1014	cfg open fail	
1015	cfg too many index	
1016	cfg unexpected eof	
1017	end of file	
1018	EOB file contains bad information	
1019	cannot convert	
1020	Data Error. Checksum of file is bad	
1021	DocClass info not in	

	configuration file	
1022	Number of Documents in EOB and transact.dat mismatch	
1023	Document ID List File is required (-f option)	
1024	Can't exec command	
1025	Index is not defined in document class	
1026	Invalid document number	
1027	Invalid object id	
1028	Invalid object name	
1029	Invalid ISR type	
1030	IO error	Could not create open, read or directory. Usually a permissions problem or the file system ran out of space or an NFS mount was lost
1031	Critical IO error – Program Abort	
1032	Can't kill a child process	
1033	Line too long	
1034	malloc failed	Unable to get an allocation of memory from operating system.
1035	Miscellaneous error	
1036	Source domain must be supplied for this application	
1037	Missing required index in configure	
1038	Required command-line option missing	
1039	No annotations were found for this document	ISR error on attempt to complete Activity Logging Event on an annotation.
1040	Document does not contain any page	
1041	Multiple Classes are not supported by this application	ONLY MRII will receive this error. MRII does not support multi doc classes.
1042	One or more critical resources are not available	
1043	Not a valid FileNet format. See SC_convert.out for details	
1044	Not enough space	For HPII Page Cache is full and for MRII BES Cache is

		full. Check cache.
1045	open file fail	
1046	Number of Pages in EOB and transact.dat mismatch	
1047	Value for required index is missing	
1048	Can't spawn the external report program	
1049	System Error, see Errno	
1050	Program Terminal ID is Required	
1051	Image file did not validate	
1052	Too many docs in subbatch. MaxDocPerSubBatch too small	
1053	Too many docs returned by query. Must query on unique index	
1054	Batch contains more than one family	Multiple Families NOT supported.
1055	too many pages in doc	Document exceeds the max page limit of %d. Ln='%s'
1056	Too many sub batches. MaxBatchSize too small	HPII ONLY: Maximum number of sub batches has been reached. MaxBatchSize in the configuration file may be too small. This must be big enough to hold 1000 documents or 2000 pages.
1057	Transact file contains incorrect format	
1058	Required input file was not provided	
1059	PS Shared Library Mismatch-Contact FileNet Support	
1060	EMC-Centera Support not yet available-No ClipId Produced	
1061	EMC-Centera device IO error	
1062	ISR Validation Failure	
1063	Unsupported SSN	ISR error can occur if all system SSNs in the ISR NET are not in the license key (PSLSF).
1064	Preliminaries Complete	

13.15 Common P8-IM Error codes

15,16,17 – This is an P8-IM network error and will need to be resolved by the System Administrator or by contacting FILENET CSS department for support.

156,0,24 - <NCH,0,24> “A network related error was encountered.”
The most common reasons for this error are:

- They do not have the correct “nch-server” entry in the /etc/hosts file.
- They provided an incorrect Domain or Organization name in the ISR.cfg file.
- The FILENET P8-IM software is stopped.
- There is a network problem.

202,0,9 – SysV segmentation violation: Problem between P8-IM and WAL memory management modules.

202,6,3 - ./CheckSSN (10662) WARNING: Current process is not registered
This error usually means that the WAL libraries are not installed properly and the PS Utility is accessing the P8-IM libraries instead of the WAL libraries.

14. APPENDIX A–EPHEMERAL PORT SETTINGS

14.1 Definition

A TCP/IPv4 connection consists of two endpoints, and each endpoint consists of an IP address and a port number. Therefore, when a client user connects to a server computer, an established connection can be thought of as the 4-tuple of: (server IP, server port, client IP, client port). Usually three of the four are readily known -- client machine uses its own IP address and when connecting to a remote service, the server machine's IP address and service port number are required.

What is not immediately evident is that when a connection is established that the client side of the connection uses a port number. Unless a client program explicitly requests a specific port number, the port number used is an ***ephemeral*** port number. Ephemeral ports are temporary ports assigned by a machine's IP stack, and are assigned from a designated range of ports for this purpose. When the connection terminates, the ephemeral port is available for reuse, although most IP stacks won't reuse that port number until the entire pool of ephemeral ports have been used. So, if the client program reconnects, it will be assigned a different ephemeral port number for its side of the new connection.

Similarly, for UDP/IP, when a datagram is sent by a client from an unbound port number, an ephemeral port number is assigned automatically so the receiving end can reply to the sender.

Symptom and Description:

It may be necessary to tune the "ephemeral port" network settings on your server's OS.

The default settings that come pre-configured with the OS are often insufficient for high-volume FileNet activity. As a result, the OS can intermittently run out of free socket ports. This will momentarily prevent the server from opening any new TCP connections, at which time activity will come to a halt. The system may or may not be able to recover once free ports become available again.

The system might be susceptible to this problem if your application mix involves a high number of short server connections.

Symptoms of the problem may include:

* Client intermittently receives "Method of object failed" during Doc.GetCachedFile
* Image Services Toolkit logs get chronic SPP type errors (e.g. 15,16,17): COR_Open: connect failed with errno 10048
* Workflo Queue intensive app gets: "COR_Open: connect to 30.34.192.237 [32769] failed with errno 227"

There are two methods for tuning the ephemeral ports:
* Increase the range of ports available to the OS
* Decrease the "TIME_WAIT" period for which the OS cannot reuse a closed port number (Reducing TIME_WAIT is NOT CONFIGURABLE on AIX and HPUX 10.20)

Both methods are platform-dependent. Refer to the sections below for the correct method for your OS.

Both methods are independent of each other - you can do one, the other, or both.

We recommend increasing the ephemeral port range when you install the IS server software. We don't recommend modifying TIME_WAIT unless the need arises.

There are no hard and fast guidelines for decreasing TIME_WAIT. The default value is usually 4 minutes. It can often safely be reduced to as little as 2 minutes or even 30 seconds. TIME_WAIT is designed to prevent any "lost packets" from an old connection from having the same port number as a current connection. If you're in a high-latency environment (for example, WAN or satellite traffic), your TIME_WAIT should be longer (e.g. the full four minutes). If you're certain that all packets will be received promptly (within milliseconds), then you can safely reduce TIME_WAIT.

14.2 Windows platform

The ephemeral ports need to be adjusted for all platforms but the Windows platform is the most affected if the ports are not modified.

The modifications need to be made on all the servers that are part of the FileNet domain.

Use the Registry Editor (regedt32.exe) to make the modifications.

14.3 *MaxUserPort*

Description:

Determines the highest port number TCP can assign when an application requests an available user port from the system. Typically, ephemeral ports (those used briefly) are allocated to port numbers 1024 through 5000.

Note: Windows 2000 does not add this entry to the registry. You can add it by editing the registry or by using a program that edits the registry.

Location:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Data Type: REG_DWORD

Range: 5,000-65,534 (port number)

Default Value: 5000

Recommended value: 65534 (65534 DEC)

14.4 *TcpMaxConnectTransmissions*

Location:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Data Type: REG_DWORD

Recommended value: 5 (5 DEC)

14.5 *TcpMaxConnectRetransmissions*

Description:

Determines how many times TCP retransmits an unanswered request for a new connection. TCP retransmits new connection requests until they are answered or until this value expires

Location:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Data Type: REG_DWORD

Range: 0–255 (*retransmission attempts*)

Default Value: 2

Recommended value: 5 (5 DEC)

14.6 *TcpTimedWaitDelay*

Description:

Determines the time that must elapse before TCP can release a closed connection and reuse its resources. This interval between closure and release is known as the TIME_WAIT state or 2MSL state. During this time, the connection can be reopened at much less cost to the client and server than establishing a new connection.

RFC 793 requires that TCP maintain a closed connection for an interval at least equal to twice the maximum segment lifetime (2MSL) of the network. When a connection is released, its socket pair and TCP control block (TCB) can be used to support another connection. By default, the MSL is defined to be 120 seconds, and the value of this entry is equal to two MSLs, or 4 minutes. However, you can use this entry to customize the interval.

Reducing the value of this entry allows TCP to release closed connections faster, providing more resources for new connections. However, if the value is too low, TCP might release connection resources before the connection is complete, requiring the server to use additional resources to reestablish the connection.

Note: Normally, TCP does not release closed connections until the value of this entry expires. However, TCP can release connections before this value expires if it is running out of TCP control blocks (TCBs). The number of TCBs the system creates is specified by the value of the MaxFreeTcbs <58770.asp> entry.

Note: Windows 2000 does not add this entry to the registry. You can add it by editing the registry or by using a program that edits the registry.

Location:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Data Type: REG_DWORD

Range: 30-300 seconds

Default Value: 240 seconds = 4 minutes

Recommended value: 45 seconds (45 DEC)

14.7 *FN_COR_QLEN*

15,16,17 errors – WSAECONNREFUSED 10061 – indicates that the NLS_Archiver is not able to connect to the IS COR_Listen die to unavailability of COR queue space.

It appears that the FN_COR_QLEN variable needs to be explicitly set in the Windows, or the default of 5 is not enough.

To do so, go to **My Computer >Advance >. Environment Variables** and add the variable FN_COR_QLEN with a value of 20-25 (this is the default used on UNIX systems). This should resolve the 15,15,17 errors.

14.8 UNIX Platform

Possible issues with ephemeral ports on UNIX platforms

On the IS server check to see if the UDP low ephemeral port is set incorrectly. We've only seen this on AIX so far, but it might happen on Solaris as well.

On AIX enter 'no -a | grep udp_ephemeral'

On Solaris enter ' ndd /dev/udp udp_smallest_anon_port' , then ' ndd /dev/udp udp_largest_anon_port'

On HP 10.30/11.x enter ' ndd /dev/udp udp_smallest_anon_port' , then ' ndd /dev/udp udp_largest_anon_port'

On HP 10.20, this shouldn't be a problem, but you can check to see if high_port_enable flag has been set by entering '/usr/contrib/bin/nettune -l | grep high' (the switch is a lower case L)

In all cases the number for the low end of the range should be higher than 32771. By default most UNIX implementations start this value at 32768, and this can allow other programs to come in and step on NCH_daemon.

14.9 IBM AIX

On AIX, check the /etc/rc.net file (or on AIX 5.2 in the /etc/tunables/nextboot file) for the following lines:

/usr/sbin/no -o udp_ephemeral_high=65535

/usr/sbin/no -o udp_ephemeral_low=42767

Steps to modify the /etc/rc.net file as documented in the IS 4.0 Installation and Configuration Procedures for AIX/6000, please consult for additional details.

Perform the steps in this section on all servers. Backup the /etc/ rc.net file. As root user, make sure you have write permission on the above files by entering:

chmod 754 /etc/ rc.net

Use your preferred text editor (such as vi) to modify the /etc/ rc.net file.

Locate the following 'if' statement near the end of the file.

```
##################################################
# The socket default buffer size (initial advertised TCP window) is being
# set to a default value of 16k (16384). This improves the performance
# for ethernet and token ring networks. Networks with lower bandwidth
```

```
# such as SLIP (Serial Line Internet Protocol) and X. 25 or higher bandwidth
# such as Serial Optical Link and FDDI would have a different optimum
# buffer size.
# ( OPTIMUM WINDOW = Bandwidth * Round Trip Time )
#####################################################
if [ -f /usr/ sbin/ no ] ; then
        /usr/sbin/ no -o tcp_sendspace= 16384
        /usr/sbin/ no -o tcp_recvspace= 16384
fi
```

Add the following lines:

```
/usr/sbin/ no -o tcp_keepidle= 80
/usr/sbin/ no -o tcp_keepintvl= 20
/usr/sbin/ no -o tcp_ephemeral_ high= 65535
/usr/sbin/ no -o tcp_ephemeral_ low= 42767
/usr/sbin/ no -o udp_ephemeral_ high= 65535
/usr/sbin/ no -o udp_ephemeral_ low= 42767
```

so the “if” statement looks like this:

```
if [ -f /usr/sbin/ no ] ; then
        /usr/sbin/ no -o tcp_sendspace= 16384
        /usr/sbin/ no -o tcp_recvspace= 16384
        /usr/sbin/ no -o tcp_keepidle= 80
        /usr/sbin/ no -o tcp_keepintvl= 20
        /usr/sbin/ no -o tcp_ephemeral_ high= 65535
        /usr/sbin/ no -o tcp_ephemeral_ low= 42767
        /usr/sbin/ no -o udp_ephemeral_ high= 65535
        /usr/sbin/ no -o udp_ephemeral_ low= 42767
fi
```

Save your changes and exit from the file.

Reboot the server (s)

14.10 HP-UX

HPUX 10.20 ONLY

Perform the steps in this section on all servers. You can make your FileNet system run more efficiently by making changes to the /etc/rc.initfnsw file. The modification expands the number of available ephemeral ports. These modifications are not required, but have been found to be optimal when running FileNet software. So unless you have set these options for other system reasons, we recommend that you make these changes.

Ephemeral ports are temporary ports assigned by a server's IP stack, and are assigned from a designated range of ports for this purpose. When network traffic is extremely heavy, it’s possible to run out of ephemeral ports unless

you specify the high_ port_ enable option in /etc/rc.initfnsw. Backup the /etc/rc.initfnsw file. As root user, make sure you have write permission on the this file by entering:

chmod 754 /etc/rc.initfnsw

Use your preferred text editor (such as vi) to modify the /etc/rc.initfnsw file.

Locate the following statement near the end of the file: # **Set up network options HPUX (parameters in half- seconds)**

Add the following line somewhere after that statement:

/usr/contrib/bin/nettune -s tcp high_ port_ enable 1

Save your change and exit from the file.

Reboot server (s)

HPUX 10.30/HPUX 11.x

As root user, vi /sbin/rc2.d/S340net =>

ndd -set /dev/udp udp_smallest_anon_port 42767
ndd -set /dev/udp udp_largest_anon_port 65535
ndd -set /dev/tcp tcp_smallest_anon_port 42767
ndd -set /dev/tcp tcp_largest_anon_port 65535
ndd -set /dev/tcp tcp_time_wait_interval 30000 (for HPUX 10.30/11.x)

Save your change and exit from the file.

Reboot server (s)

14.11 Sun Solaris

Perform the steps in this section on all servers. On Solaris, you set these parameters in /etc/rc2.d/S69inet - double check the install documentation for syntax, or contact CSS Network Support for assistance.

You can make your FileNet system run more efficiently by making changes to the /etc/rc2.d/S69inet file. The modification expands the number of available ephemeral ports and reduces the time-out delay. These modifications are not required, but have been found to be optimal when running FileNet software. So unless you have set these options for other system reasons, we recommend that you make these changes.

Ephemeral ports are temporary ports assigned by a server's IP stack, and are assigned from a designated range of ports for this purpose. When network traffic is extremely heavy, it's possible to run out of ephemeral ports unless you specify a wider range of port numbers in /etc/ rc2.d/S69inet.

The tcp_close_wait_interval parameter determines the length of time the server waits before reusing a closed ID socket. Although the default value is typically around 240000 milliseconds (four minutes), this parameter can safely be reduced to as little as 30000 milliseconds (30 seconds) on high-speed networks.

Make a backup copy of the /etc/rc2.d/S69inet file before you modify it.

As root user, make sure you have write permission on the file by entering: chmod 754 /etc/rc2.d/S69

Use your preferred text editor (such as vi) to modify the /etc/rc2.d/S69inet file.

Add the following lines somewhere near the end of the file:

ndd -set /dev/udp udp_smallest_anon_port 42767
ndd -set /dev/udp udp_largest_anon_port 65535
ndd -set /dev/tcp tcp_smallest_anon_port 42767
ndd -set /dev/tcp tcp_largest_anon_port 65535
ndd -set /dev/tcp tcp_close_wait_interval 30000 (for Solaris 2.x only)
ndd -set /dev/tcp tcp_time_wait_interval 30000 (for Solaris 8 and above)

Save your change and exit from the file.

Reboot server (s)