Supplemental Inspectors are the simple way
to extend the z/OS and sub-component
Baselines, Inspections and Change Detection
functions found in the
Integrity Controls Environment (ICE)

Supplemental Inspectors

Release 18.0

USER GUIDE

Contact us for additional information:

NewEra Software Technical Support

800-421-5035 or 408-520-7100
Or text support requests to 669-888-5061

support@newera.com

www.newera.com Rev: 2023-9-18

1 Forward

1.1 Copyright, Trademark and Legal Notices

1.1.1 Copyrights

This User Guide and the related Software Product(s) are protected under a Copyright dated 2020 by NewEra Software, Inc. All rights are reserved.

1.1.2 License Agreement

This User Guide describes the installation and operation of Image FOCUS, its environment and applications. It is made available only under the terms of a license agreement between the licensee and NewEra Software, Inc. No part of this Guide or the related Software Product(s) may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of NewEra Software, Inc.

1.1.3 Trademarks and Copyrights of Others

The following products and/or registered trademarks of International Business Machines Corporation are referenced in this document: MVS, VM, RACF, z/OS, SYSPLEX, JES, VTAM, TSO, ISPF, ICKDSF, DFSMSdss, and DF/DSS. Other company, product or service names may be trademarks or service marks of others.

1.2 Technical Support Information

Around-the-clock-support

NewEra Software is dedicated to providing the highest level of technical support to meet our customers' growing needs. In order to meet these needs, NewEra provides technical support, 7 days a week, 24 hours a day.

Reach us by Telephone during Business Hours

Please use the following phone numbers to reach our technical support staff during normal business hours (6 AM to 4 PM Pacific Time):

- In North America, dial 1-800-421-5035
- Outside North America, dial 1-408-520-7100
- Support inquiries may also be texted to 669-888-5061

Reach us by Telephone during non-Business Hours

In case of an emergency, during non-business hours, phone the above numbers to receive instructions on how to contact a Technical Support Representative or a Technical Support Manager.

Sending Email

Our technical support staff can be reached by email at support@newera.com. Email messages will be answered by the next business day. Product technical questions or product recommendations may be sent via email.

Help through the NewEra website

You can access technical support from www.newera.com. Click the Support tab at the top of the screen to reach our Technical Support Request page.

Service Levels

NewEra is committed to providing the highest level of quality to our customers by adopting the following criteria for responding to customer requests:

- All critical questions received by phone during working hours will be answered within 15 minutes of receiving the request;
- Technical questions sent by email, or messages sent through our Technical Support Request page, will be answered by the next business day.

We Want Your Suggestions!

NewEra understands the significance of providing our customers with the highest quality support and welcomes all suggestions as to how we may improve Technical Support.

1.3 About this Document

This document is designed to describe the operation of three individual Inspectors: ISNLOAD, ISNMBRS, and ISNCSDS. While each is unique in its own right and designed to address a specific z/OS configuration component, all share common configuration and interface elements. To avoid redundancy in the text, these shared elements are documented only once. For example:

- In the chapter titled *Using the Supplemental Inspectors* the discussion is general enough that when you are finished, you will know how to access and use all three as either a Component, Workbench or Production Inspector.

- In both *Appendix "A" and "B"* you will learn the general methods used for downloading, installing and configuring any of the inspectors.

The specifics of each inspector are addressed in chapters with titles like *Working with ISNLOAD.* In these, you will find the detail that will help you customize and exploit each.

1.4 Recent Enhancements to the Supplemental Inspectors

- Addition of the CSDS Supplemental Inspector: This new Supplemental Inspector evaluates and blueprints configuration information from named CICS System Definition files;
- Addition of an optional Posting Process that routes Summary Inspection and Change Reports to the ICE Control Journals. Posted reports are accessed using The Control Editor Journal and History functions;
- Numerous minor improvements in the optional "In-Line Interface" as well as in overall product documentation, availability, reliability and serviceability.

1.5 System Requirements and Limitations

1.5.1 Prerequisites

To use the Supplemental Inspectors, you will need Integrity Controls Environment 15.0 or higher. You will find the latest release of Integrity Controls Environment at www.newera.com.

1.5.2 The License Key

A License Key is required to activate, MAKE ACTIVE, the Supplemental Inspectors. Once the License Key is inserted, the Supplemental Application functions will be unlocked the next time you logon to Integrity Controls Environment.

1.5.3 Releases Prior to Release 18.0

If you are a current Integrity Controls Environment user and have not yet upgraded to Integrity Controls Environment Release 18.0, special care should be taken when you do upgrade to 18.0 to remove all pre-existing Integrity Controls Environment Libraries. All pre-existing Integrity Controls Environment Inspection Reports (logs) and Package/Blueprints (packages) are fully supported by Release 18.0.

1.6 Solving Real-World Control Problems

"…As z/OS integrity becomes more important to your organization, the requirement to extend control over Authorized Programs beyond the Dataset Level will become a requirement. There may be 40,000 to 50,000 modules in a typical APF Authorization cycle. Generally, they are defined in the LPALst and LNKLst concatenations and/or mandated in Vendor and/or User application program load libraries. These libraries, in turn, are qualified for APF authorization when they are named in a prevailing/SET PROGxx ParmLib Member. This "authorized" collection of APF modules work together to provide system and application function. But the collection may not be static as modules can be added and/or deleted and become *targets of change* via defined programmatic processes, SMP/E or more ad hoc Operator Commands or processes commonly called "ZAPs". ISNLOAD can detect such changes, allowing you to identify and account for all module/object changes and/or anomalies; demonstrating this accounting to those with a need to know, such as auditors, is important to both the perceived and real integrity of z/OS."

2 Table of Contents

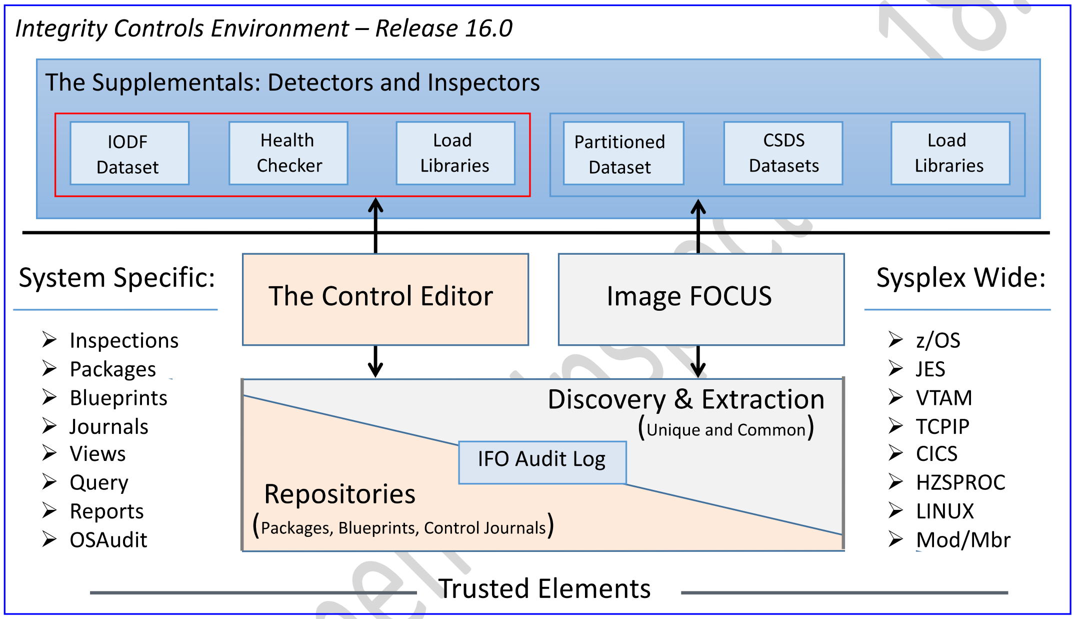
3 The ICE Environment

The Integrity Controls Environment (ICE) is a VTAM Application that provides access to ICE Applications:

NewEra Software

z/OS Integrity and Compliance

z/OS Configuration Management and Compliance – ICE 16.0 - Overview



3.1 Image FOCUS

The Image FOCUS Application set automatically discovers, extracts, blueprints and inspects the z/OS configuration components that comprise a Sysplex and its Images. Process findings are shared with other ICE applications via a Sysplex Audit Log.

3.2 The Control Editor

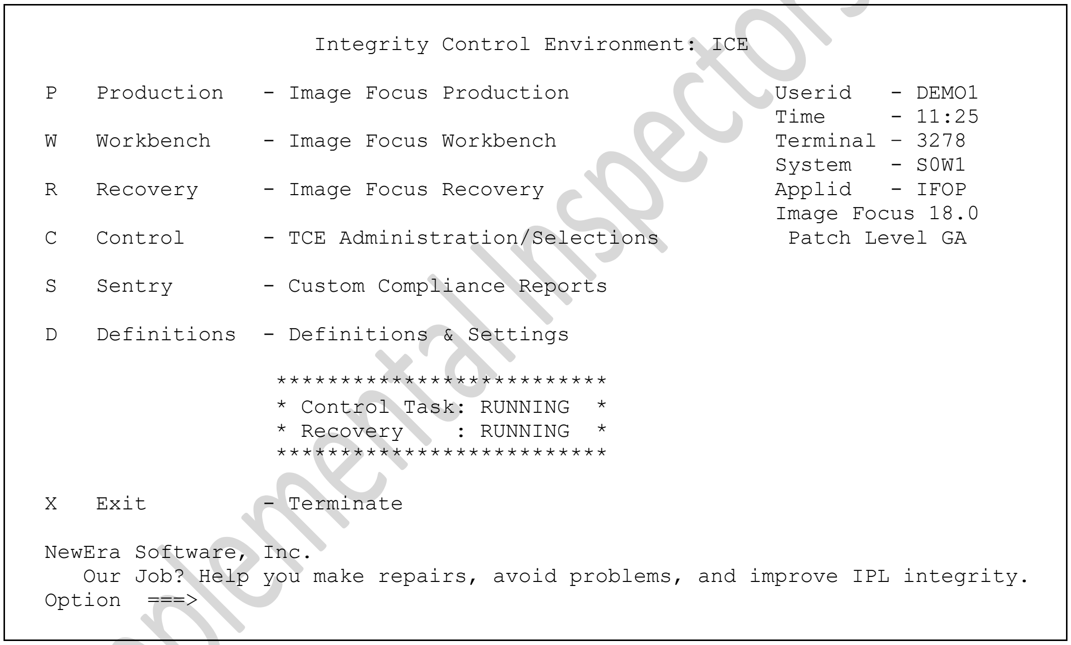
The Control Editor is a "Compensating Control" that provides a layer of non-invasive security over the z/OS configuration components housed in defined sets of partitioned datasets. TCE significantly enhances the level of security generally provided by the site's External Security Manager (ESM).

3.3 Image SENTRY

The Image SENTRY Application set addresses issues commonly associated with the organization, analysis, reporting and documentation of the major z/OS operational components: the IODF configuration, the organization of CICS regions and the status and standing of RACF.

3.4 ICE Primary Menu

To access the Supplemental Inspectors, logon to ICE. Next, from the ICE Primary menu, select the Workbench using option “W”.

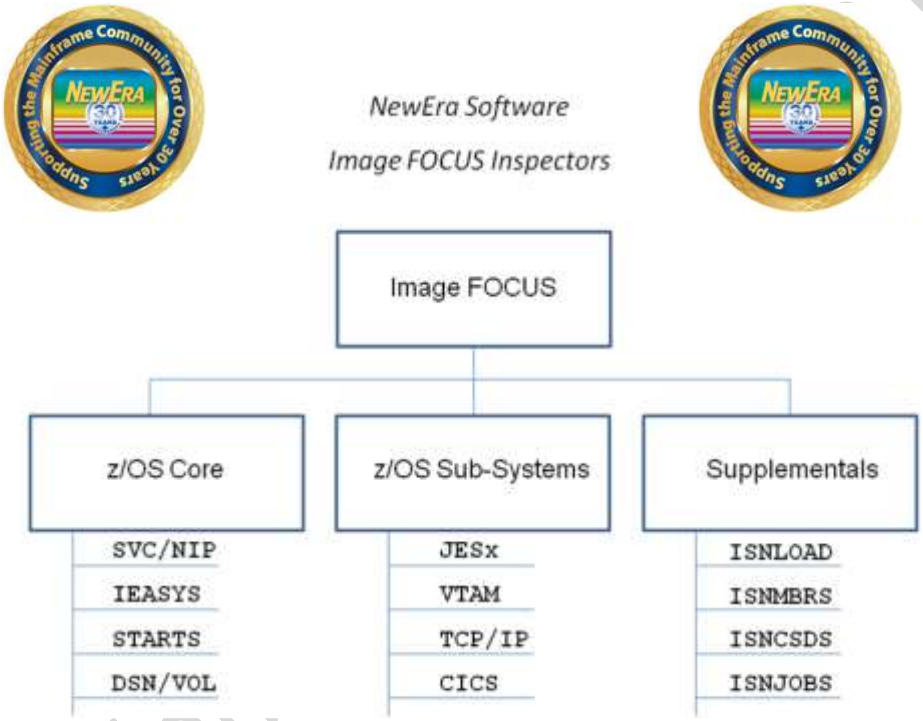


```
                     Integrity Control Environment: ICE

P   Production   - Image Focus Production                Userid   - DEMO1
                                                         Time     - 11:25
W   Workbench    - Image Focus Workbench                 Terminal - 3278
                                                         System   - S0W1
R   Recovery     - Image Focus Recovery                  Applid   - IFOP
                                                         Image Focus 18.0
C   Control      - TCE Administration/Selections          Patch Level GA

S   Sentry       - Custom Compliance Reports

D   Definitions  - Definitions & Settings

                  **************************
                  * Control Task: RUNNING  *
                  * Recovery    : RUNNING  *
                  **************************

X   Exit         - Terminate

NewEra Software, Inc.
   Our Job? Help you make repairs, avoid problems, and improve IPL integrity.
Option  ===>
```

4 Unlocking the Latent Value of Integrity Controls Environment (ICE)

4.1 z/OS Core and Subsystem Inspectors

The Image FOCUS z/OS Core and Subsystem Inspectors automatically discover, inspect, blueprint and detect changes in the Image and Sysplex configuration components *ACTUALLY USED* during z/OS initialization. These *PREVAILING* configuration components are automatically drawn from a number of defined dataset sources, i.e. ParmLib, ProcLib, VTAMLib and VTAMLst, LPALib, LINKLib and LINKLst and shared, as needed, with the Supplemental Inspectors.



4.1 Integrating the Supplemental Inspectors

When integrated into the Integrity Controls Environment (foreground and background), the Supplemental Inspectors complement and add depth to the Image FOCUS z/OS Core and Subsystem Inspectors. As needed, they leverage information derived from Image FOCUS inspections, blueprinting and reporting processes. Their primary directive is to expand the scope of inspection and blueprinting by bringing both prevailing components and their *NON-PREVAILING* dataset cohorts into view. This integration is easily accomplished as users leverage Image FOCUS Auto Discovery and automatically identify *Dataset Control Points* or create one or more of their own design. A Dataset Control Point is, in fact, a logical grouping of datasets accessed by the Supplemental Inspectors during execution. They provide the operational basis for Supplemental Inspections, blueprinting and change detection. Three Image FOCUS Supplemental Inspectors are available and include:

4.1.1 ISNLOAD

The z/OS operating system, its subsystems and vendor applications are composed of tens of thousands of individual load modules. Each is uniquely encoded to assure z/OS integrity while at the same time fulfilling its role in providing end user functionality. Updates such as *PTFs, ZAPs and PATCHes* can significantly affect functionality and/or integrity. This Supplemental Inspector evaluates the content of one or more Dataset Control Point concatenations for duplications and orphaned aliases, blueprints the concatenation, detects changes in its composition, and reports its findings as directed.

4.1.2 ISNMBRS

Members in Partitioned Datasets are the primary source of z/OS configuration components, i.e. ParmLib, ProcLib and VTAMLib. Members that do not prevail during an IPL are often used later during *Dynamic System Updates*. This Supplemental Inspector evaluates the content of one or more Dataset Control Point concatenations for z/OS and non-z/OS members, blueprints the concatenation, detects changes in its composition, and reports its findings as directed.

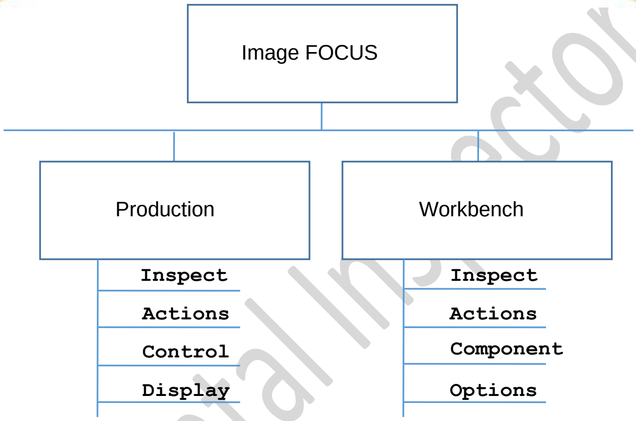
4.1.3 ISNCSDS

The CICS System Configuration Dataset (CDS) contains the configuration parameters used in the initialization of one or more CICS Regions. The purpose of this Supplemental Inspector is to extract *All or Named GROUPs* from the CDS, evaluate their configuration parameters against a set of user defined standards, blueprint the GROUPs, detect changes, and report findings as directed.

5 Using the Supplemental Inspectors

5.1 Common Interface Elements

All Supplemental Inspectors share a common Integrity Controls Environment (ICE) interface and can be accessed as either Component Inspectors or Image Inspectors. All Image Definitions created under the Workbench that defines the activation of one or more of the Supplementals are automatically included and run with background processing when such an Image Definition is promoted to *Production Status*.



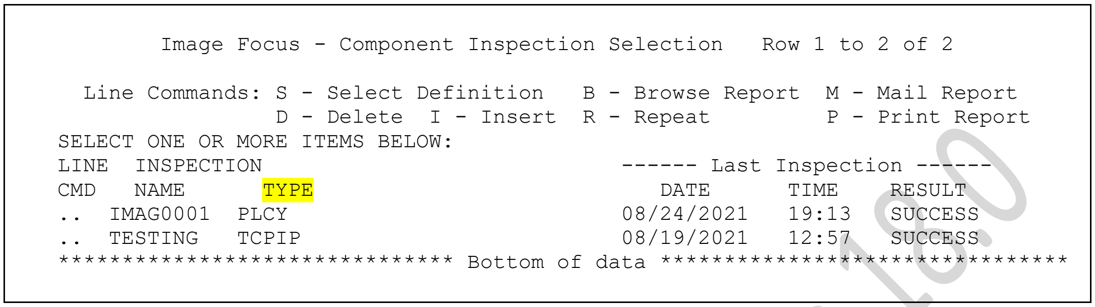
5.2 Component Inspection

When used for a Component Inspection, Supplemental Inspectors target *Control Point Datasets* specified in a user defined Source List. There are two primary advantages to using a Supplemental as a Component Inspector:

- First, since only a single inspector at a time can be called, results will present faster;
- Second, because the Source List is user specified, many Lists can be tested quickly.

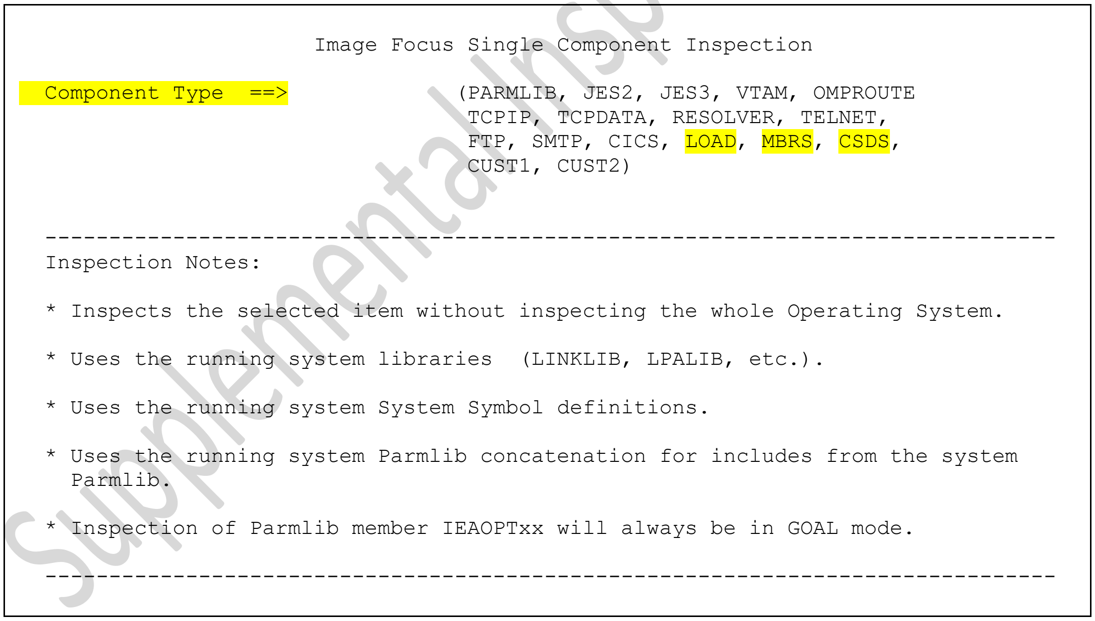
5.2.1 Defining a Component Inspector

To run a Supplemental Component Inspection, logon to ICE and select the WORKBENCH Option. Next select COMPONENT. This will display a list of currently active Component Inspectors, as shown below:



```
        Image Focus - Component Inspection Selection   Row 1 to 2 of 2

  Line Commands: S - Select Definition   B - Browse Report  M - Mail Report
                 D - Delete  I - Insert  R - Repeat         P - Print Report
SELECT ONE OR MORE ITEMS BELOW:
LINE  INSPECTION                            ------ Last Inspection ------
CMD   NAME      TYPE                           DATE      TIME    RESULT
..  IMAG0001  PLCY                          08/24/2021   19:13   SUCCESS
..  TESTING   TCPIP                         08/19/2021   12:57   SUCCESS
******************************* Bottom of data ********************************
```

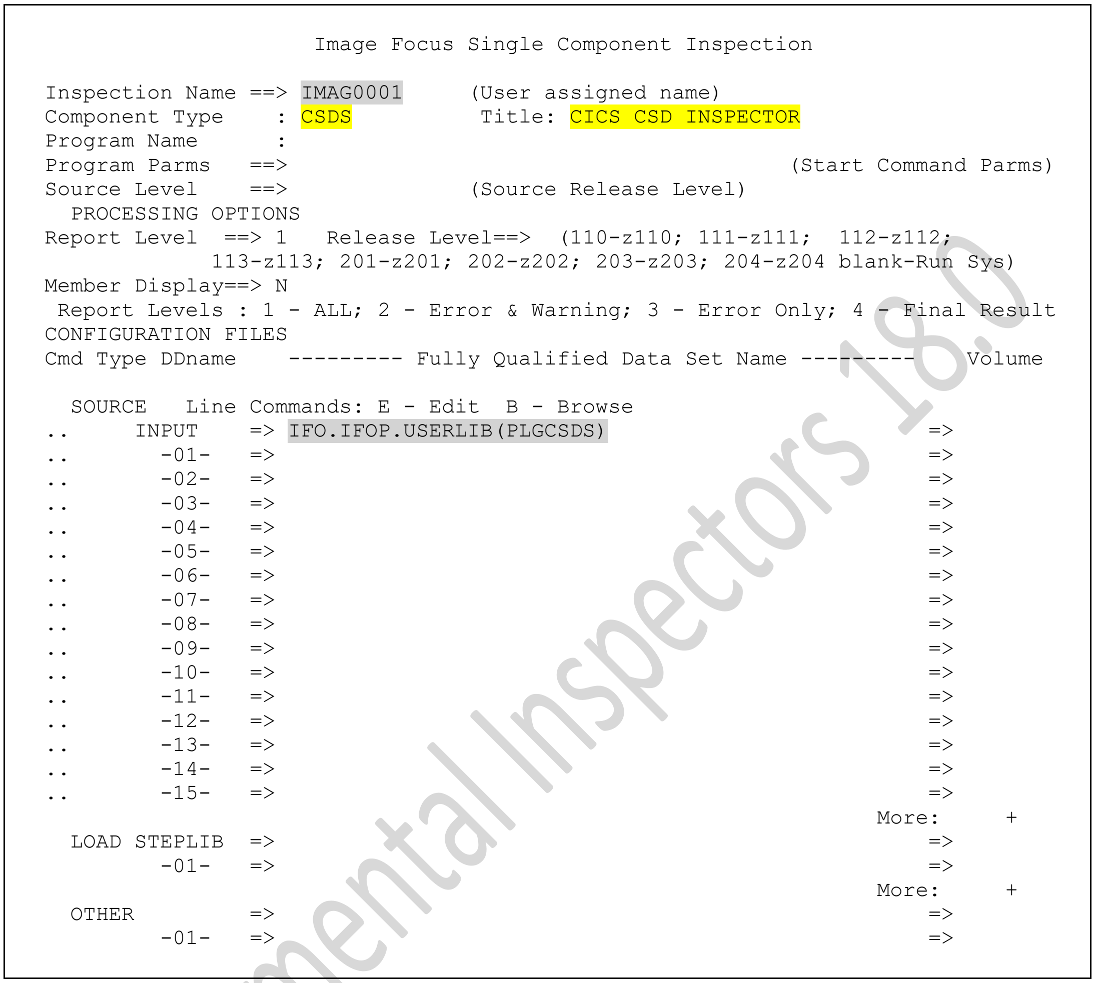
Take note of the values shown in the columns below "TYPE" and "NAME". If you find the Supplemental matched with Name you are interested in, you may reuse it by entering an "S" on the command line and pressing enter. If not, you will need to create a new entry; to accomplish this, enter "I" on the Command Line and press enter. This will display the Inspector Selection Menu shown below:



```
                     Image Focus Single Component Inspection

Component Type  ==>             (PARMLIB, JES2, JES3, VTAM, OMPROUTE
                                 TCPIP, TCPDATA, RESOLVER, TELNET,
                                 FTP, SMTP, CICS, LOAD, MBRS, CSDS,
                                 CUST1, CUST2)


-------------------------------------------------------------------------------
Inspection Notes:

* Inspects the selected item without inspecting the whole Operating System.

* Uses the running system libraries  (LINKLIB, LPALIB, etc.).

* Uses the running system System Symbol definitions.

* Uses the running system Parmlib concatenation for includes from the system
  Parmlib.

* Inspection of Parmlib member IEAOPTxx will always be in GOAL mode.

-------------------------------------------------------------------------------
```

From the list of Inspector Types, select one of the Supplemental Inspectors (LOAD, MBRS, or CSDS) and enter its label as the value of Component Type and press enter.

This will display the Component Inspection Interface for the selected Inspector. In this example, the CICS CSDS Inspector interface is shown below.



```
                     Image Focus Single Component Inspection

Inspection Name ==> IMAG0001     (User assigned name)
Component Type    : CSDS          Title: CICS CSD INSPECTOR
Program Name      :
Program Parms   ==>                                       (Start Command Parms)
Source Level    ==>              (Source Release Level)
  PROCESSING OPTIONS
Report Level  ==> 1   Release Level==>  (110-z110; 111-z111;  112-z112;
             113-z113; 201-z201; 202-z202; 203-z203; 204-z204 blank-Run Sys)
Member Display==> N
 Report Levels : 1 - ALL; 2 - Error & Warning; 3 - Error Only; 4 - Final Result
CONFIGURATION FILES
Cmd Type DDname    --------- Fully Qualified Data Set Name ---------   Volume

  SOURCE   Line Commands: E - Edit  B - Browse
..     INPUT    => IFO.IFOP.USERLIB(PLGCSDS)                       =>
..       -01-   =>                                                  =>
..       -02-   =>                                                  =>
..       -03-   =>                                                  =>
..       -04-   =>                                                  =>
..       -05-   =>                                                  =>
..       -06-   =>                                                  =>
..       -07-   =>                                                  =>
..       -08-   =>                                                  =>
..       -09-   =>                                                  =>
..       -10-   =>                                                  =>
..       -11-   =>                                                  =>
..       -12-   =>                                                  =>
..       -13-   =>                                                  =>
..       -14-   =>                                                  =>
..       -15-   =>                                                  =>
                                                                 More:     +
  LOAD STEPLIB  =>                                                  =>
         -01-   =>                                                  =>
                                                                 More:     +
  OTHER         =>                                                  =>
         -01-   =>                                                  =>
```

5.2.2 Running the Component Inspector

Using this interface, you will need to provide only TWO values:

- First, Inspection Name – a user defined label identifier for this specific setup,

- Second, INPUT – The matching Dataset (member) Source List for the inspector.

When this information is entered and/or modified and you have validated its accuracy, press enter to begin running the Inspector and optionally display the “Supplemental In-Line Interface.” Note that as you exit out to the ICE Main Menu, a new selection entry will appear for any newly named Inspector.

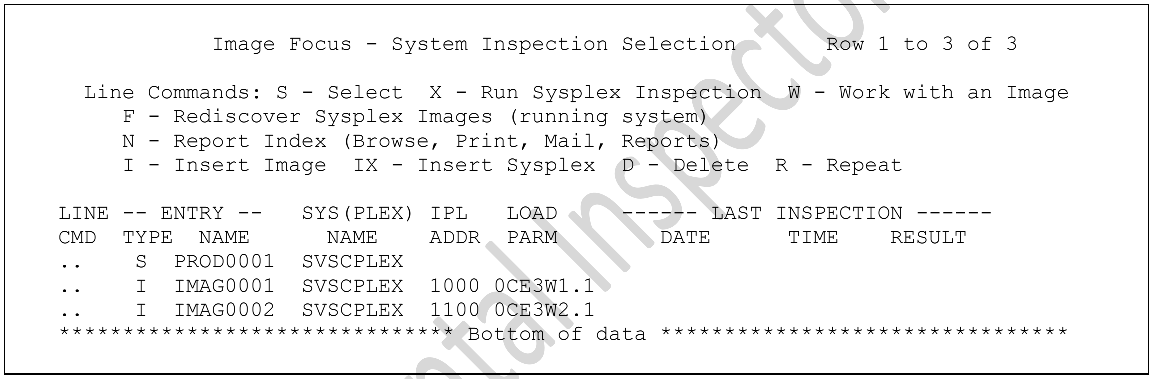
5.3 Workbench Inspection

When used for a Workbench Inspection, Supplemental Inspectors target *Control Point Datasets* specified by the user in a pre-defined Source List or automatically determined by Image FOCUS during a Core or Subsystem Inspection. There are two primary advantages to using a Supplemental as a Workbench Inspector:

- First, the correlation of “Auto Discovered” Source List information with a defined Image;
- Second, a full Core Inspection ensures overall process integrity.

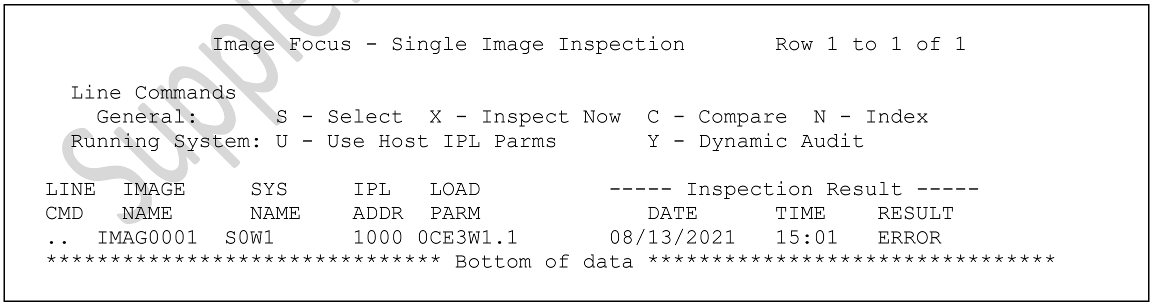
5.3.1 Defining a Workbench Inspector

To define a Supplemental Workbench Inspector, logon to ICE and select the WORKBENCH Option. Next select INSPECT. This will display a list of currently active IMAGE (System) Inspectors, as shown below:



```
              Image Focus - System Inspection Selection       Row 1 to 3 of 3

    Line Commands: S - Select  X - Run Sysplex Inspection  W - Work with an Image
       F - Rediscover Sysplex Images (running system)
       N - Report Index (Browse, Print, Mail, Reports)
       I - Insert Image  IX - Insert Sysplex  D - Delete  R - Repeat

  LINE -- ENTRY --   SYS(PLEX) IPL   LOAD   ------ LAST INSPECTION ------
  CMD  TYPE  NAME      NAME    ADDR  PARM         DATE      TIME    RESULT
  ..     S  PROD0001  SVSCPLEX
  ..     I  IMAG0001  SVSCPLEX  1000 0CE3W1.1
  ..     I  IMAG0002  SVSCPLEX  1100 0CE3W2.1
  ******************************* Bottom of data ********************************
```

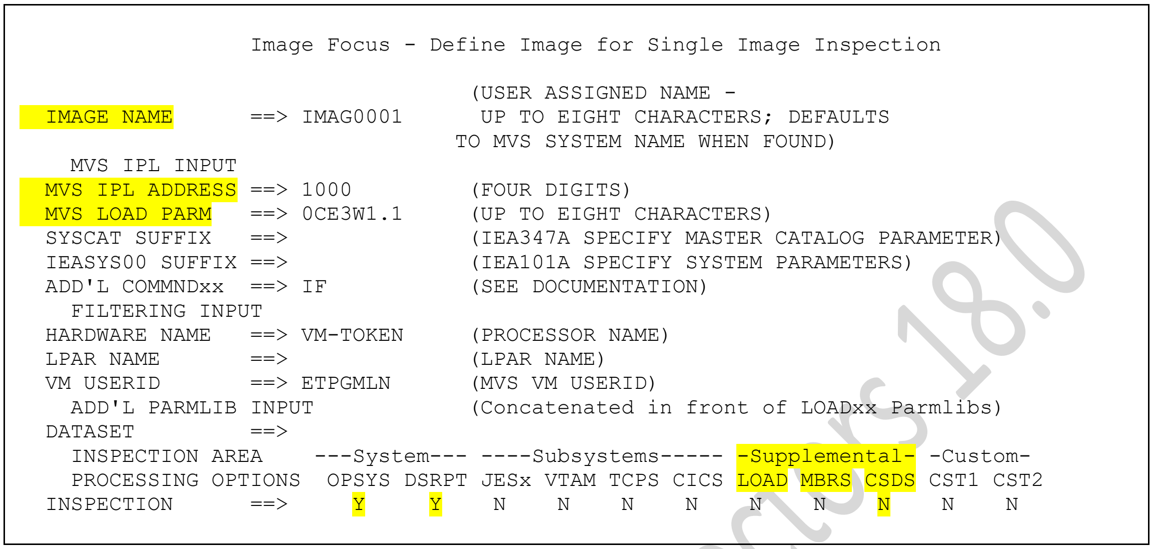
To run an IMAGE Inspection, select the image target by placing a “W” (Work with an Image) on the command line before the Image (Entry) Name and press enter. This will display a secondary panel, shown below, that provides specific Image analytic functions.



```
             Image Focus - Single Image Inspection       Row 1 to 1 of 1

  Line Commands
    General:      S - Select  X - Inspect Now  C - Compare  N - Index
  Running System: U - Use Host IPL Parms       Y - Dynamic Audit

 LINE  IMAGE     SYS     IPL   LOAD          ----- Inspection Result -----
 CMD   NAME      NAME    ADDR  PARM             DATE      TIME    RESULT
 ..  IMAG0001  S0W1      1000 0CE3W1.1       08/13/2021   15:01   ERROR
 ******************************* Bottom of data ********************************
```

To display the Image Inspection Interface, place an “S” on the command line before the Image Name and press enter.

The Image Inspection Interface, shown below, requires certain z/OS initialization parameters (IPL INPUT PARMS) before it can become fully operational. If you are unaware of these, ask your z/OS System Programmer for assistance.



```
                   Image Focus - Define Image for Single Image Inspection

                                  (USER ASSIGNED NAME -
  IMAGE NAME       ==> IMAG0001    UP TO EIGHT CHARACTERS; DEFAULTS
                                 TO MVS SYSTEM NAME WHEN FOUND)
    MVS IPL INPUT
  MVS IPL ADDRESS ==> 1000        (FOUR DIGITS)
  MVS LOAD PARM   ==> 0CE3W1.1    (UP TO EIGHT CHARACTERS)
  SYSCAT SUFFIX   ==>             (IEA347A SPECIFY MASTER CATALOG PARAMETER)
  IEASYS00 SUFFIX ==>             (IEA101A SPECIFY SYSTEM PARAMETERS)
  ADD'L COMMNDxx  ==> IF          (SEE DOCUMENTATION)
    FILTERING INPUT
  HARDWARE NAME   ==> VM-TOKEN    (PROCESSOR NAME)
  LPAR NAME       ==>             (LPAR NAME)
  VM USERID       ==> ETPGMLN     (MVS VM USERID)
    ADD'L PARMLIB INPUT           (Concatenated in front of LOADxx Parmlibs)
  DATASET         ==>
    INSPECTION AREA    ---System--- ----Subsystems----- -Supplemental- -Custom-
    PROCESSING OPTIONS  OPSYS DSRPT JESx VTAM TCPS CICS LOAD MBRS CSDS CST1 CST2
  INSPECTION      ==>     Y     Y    N    N    N    N    N    N    N    N    N
```

5.3.2 Running the z/OS Core Inspector

To begin a Workbench Inspection, you will need to provide the following:

- First, IMAGE NAME – If matched with Component Inspector; Name can share Source List,
- Second, MVS IPL ADDRESS – The UNIT ADDRESS of the IPL Volume,
- Third, MVS LOAD PARM – A System Volume, LOAD Suffix, MIPSI Character, and Nucleus.

The optional MVS IPL INPUT, FILTERING INPUT and ADD'L PARMLIB INPUT should conform to the specifics of the Image to be inspected. Since the z/OS Core Inspector is *ALWAYS ON,* it is considered a good practice to run a z/OS Core Inspection first with each new Image definition to *VALIDATE THE IPL INPUT PARMS BEFORE* you attempt adding any other inspectors to the inspection process.

5.3.3 Adding additional inspectors to the Inspection Process

Once the z/OS Core Inspection "Runs Clean", adding additional inspectors is accomplished by toggling the "N" below one or all of the available inspectors to "Y".

5.4 The Supplemental “In-Line” Interface

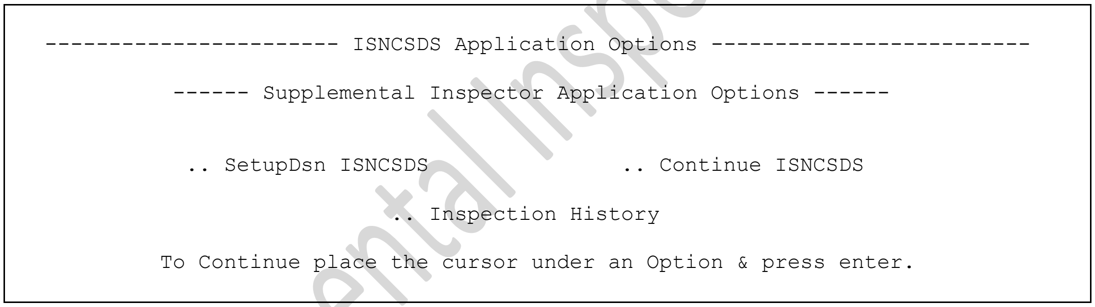
The “Supplemental In-Line Interface” is an optional product feature. To Activate/Inactivate you will need to modify the Supplemental Configuration Settings, See Appendix A for details. In this case, you will need to modify the value of the “RPTMENU=” keyword where the first four characters following the keyword (in the case below NNN) are used to represent the setting of each of the three Supplemental Inspectors: ISNLOAD, ISNMBR, and ISNCSDS.

```
RPTMENU= AAA
```

When a matching character is set to “N” the interface for that Inspector is turned OFF. When the character is set to “A” the interface for that inspector is turned ON. By default all inspection interfaces are turned ON. In addition, if the interface option is turned on, an additional optional “PASSWORD” can be added to prevent access to Source Lists and Configuration Files. By default the password for all inspectors is set to AMAZING. The correct syntactical presentation of the optional password with the “RPTMENU=” keyword is shown below:

```
RPTMENU= AAA AMAZING
```

A sample of the Primary Menu of the In-Line Interface is shown below:



```
----------------------- ISNCSDS Application Options -------------------------

         ------ Supplemental Inspector Application Options ------


          .. SetupDsn ISNCSDS                  .. Continue ISNCSDS

                         .. Inspection History

        To Continue place the cursor under an Option & press enter.
```

The Primary Menu offers three options that are selected by placing “S” on the command line before an option name or placing the cursor under the option name and pressing enter.

1. SetupDsn ISNxxxx – Provides password access to Source List and Configuration Files.
2. Inspection History – Provides access to Inspector specific historical information.
3. Continue – Continues/Resumes the inspection process.

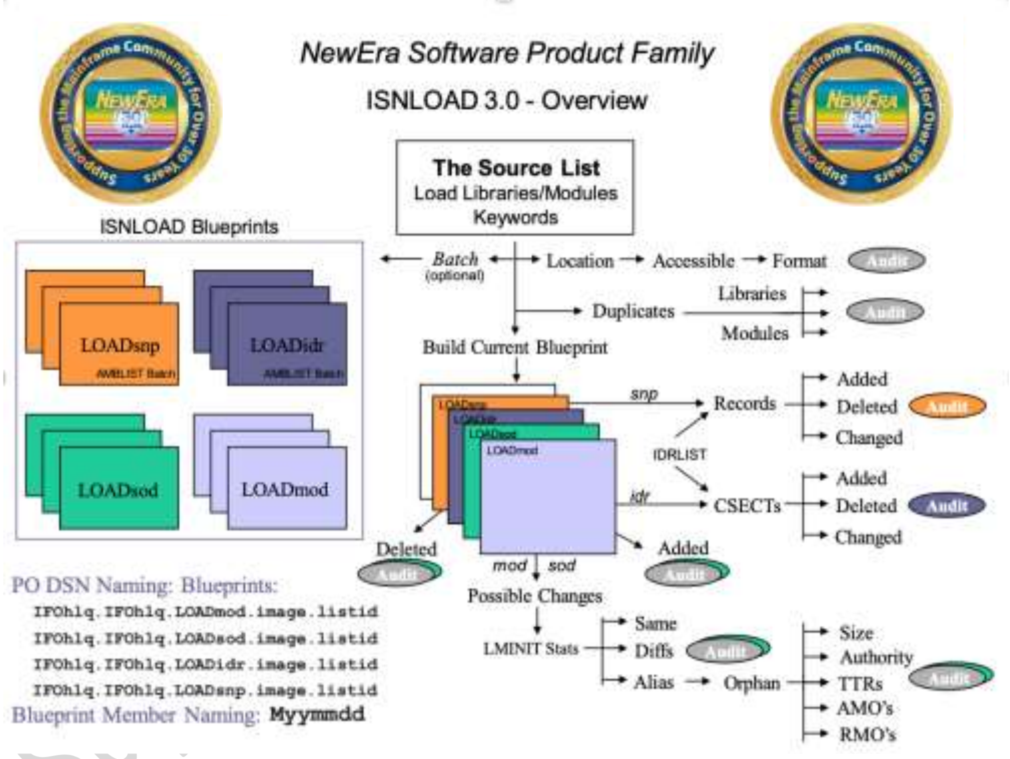
As processing continues, the various inspection and blueprinting status indicators will be displayed. When the process is complete, the Inspection results will be displayed as a report.

6 Working with ISNLOAD

ISNLOAD is a z/OS software application that can be used to effectively identify and track changes and anomalies in load libraries, the modules/objects they contain, their CSECTs and their internal structure.

6.1 Application Overview

The z/OS operating system, its subsystems and vendor applications are composed of tens of thousands of individual load modules. Each is uniquely encoded to assure z/OS integrity while at the same time fulfilling its role in providing end user functionality. Updates such as *PTFs, ZAPs and PATCHes* can significantly affect functionality and/or integrity. This Supplemental Inspector evaluates the content of one or more Dataset Control Point concatenations for duplications and orphaned aliases, blueprints the concatenation, detects changes in its composition, and reports its findings as directed.



6.2 Controlled Datasets and Source List

A Controlled Dataset candidate is any dataset that is of valued importance to your understanding of the integrity and fitness of the z/OS system environment. Candidate datasets become Controlled Datasets when they are defined to the Integrity Controls Environment and/or its Applications using a Dataset Control/Source List.

6.3 Configuring the ISNLOAD application

ISNLOAD, like all the other Supplemental Inspectors, shares certain common shared configuration definitions. In addition to those common shared definitions, ISNLOAD responds to its own unique configuration definition set.

6.3.1 ISNLOAD Configuration Options

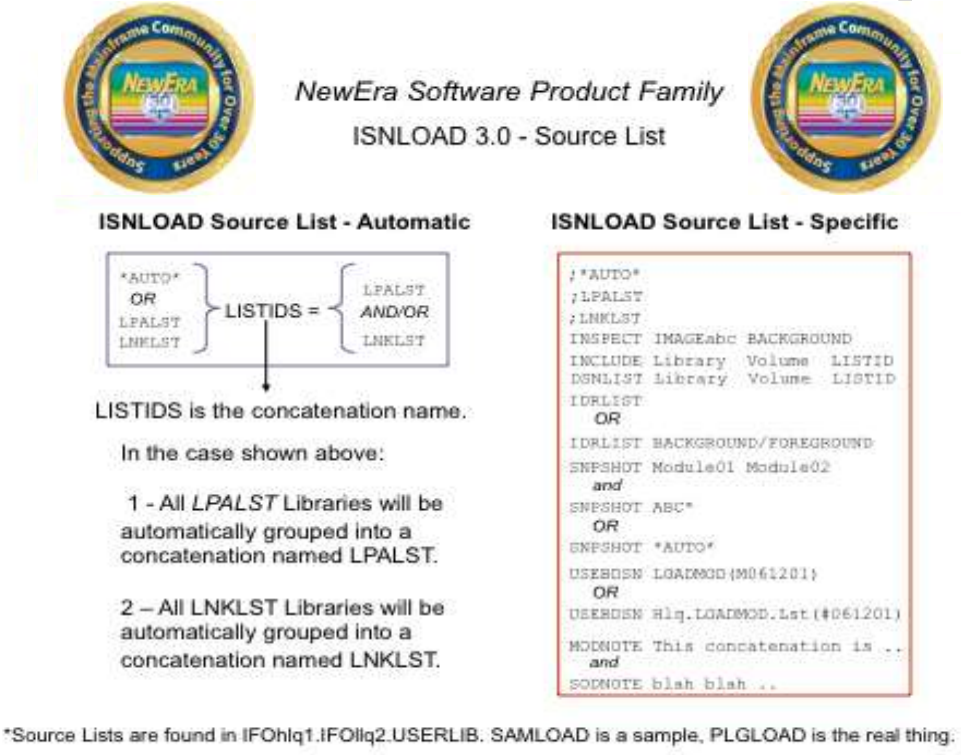
Keyword	Default/Optional	Functional Description
MODLIST=	AUTO/NONE	Lists the content of the Inspection Source List in the Inspection Report.
MODPRSS=	AUTO/NONE	Lists inspection process steps in the Inspection Report
MODDUPS=	AUTO/NONE	Identifies and Reports Duplicate Module in a Library Concatenation (Source List)
MODORPH=	AUTO/NONE	Identifies and displays the Modules that have Orphaned Aliases
MODDELT=	AUTO/NONE	Used to turn ON/OFF Blueprint Comparison and Change Detection
TTRSOFF=	AUTO/NONE	Turn ON/OFF Module TTRs as a factor in reporting detected changes
MODMDOC=	MODSUMS	Names the ISNLOAD Report to be sent when Email Notification is active.

6.4 Creating an ISNLOAD Source List

When using ISNLOAD you have a choice – run “Out of the Box” and let Image FOCUS discover the LPALST and LNKLST for each Image defined for inspection or customize the ISNLOAD Source List as described below. The ISNLOAD Source List is member PLGLOAD and can be found (along with a sample SAMLOAD) in the Integrity Controls Environment USERLIB. PLGLOAD can be edited directly under TSO for Image FOCUS or via the optional Supplemental In-Line Interface.

6.4.1 ISNLOAD Source List Overview

The Source List tells the Inspector what to Inspect and Blueprint. Its content could be as simple as a single dataset or list of datasets that are in some way related. In a more complex form the Source List contains special blueprinting instructions that specify the level of detail to be included in a blueprint and/or what exactly is to be compared when detecting changes.



NewEra Software Product Family

ISNLOAD 3.0 - Source List

ISNLOAD Source List - Automatic

LISTIDS is the concatenation name.

In the case shown above:

1 - All *LPALST* Libraries will be automatically grouped into a concatenation named LPALST.

2 – All LNKLST Libraries will be automatically grouped into a concatenation named LNKLST.

ISNLOAD Source List - Specific

```
;*AUTO*
;LPALST
;LNKLST
INSPECT IMAGEabc BACKGROUND
INCLUDE Library  Volume  LISTID
DSNLIST Library  Volume  LISTID
IDRLIST
   OR
IDRLIST BACKGROUND/FOREGROUND
SNPSHOT Module01 Module02
   and
SNPSHOT ABC*
   OR
SNPSHOT *AUTO*
USEBDSN LOADMOD(M061201)
   OR
USEBDSN Hlq.LOADMOD.Lst(#061201)
MODNOTE This concatenation is ..
   and
SODNOTE blah blah ..
```

*Source Lists are found in IFOhlq1.IFOllq2.USERLIB. SAMLOAD is a sample, PLGLOAD is the real thing.

6.4.2 ISNLOAD Source List Options

The following table details the ISNLOAD Source List Keywords and their possible values.

Keyword	Default/Optional	Functional Description
AUTO		Tells ISNLOAD to query Image FOCUS for the LPALST and LNKLST for named IMAGE
LPALST		Tells ISNLOAD to query Image FOCUS for the LPALST only for named IMAGE
LNKLST		Tells ISNLOAD to query Image FOCUS for the LNKLST only for named IMAGE
INSPECT	Image_name	Allows for multiple Image definitions in a single Source List
DSNLIST	Dataset Volume ListId	Used, line by line, to tell the inspector to acquire its Source List from the dataset/volume named. Only INCLUDE entries within the named dataset that match ListId will be included in the Inspection. See also the ***DSNLIST Specifics*** that follow this table.
INCLUDE	Dataset Volume ListId	Defines the name and location of Libraries to be included in a named concatenation. The value of ListId will be used as the concatenation name and the 4th NODE of the Blueprint Dataset. Repeat the INCLUDE Keyword for each Library to be included in a Concatenation. To begin a new concatenation, specify a new and different ListId value.
SOODODS	Pseudo Dataset Name	Used in conjunction with the INCLUDE keyword to create a pseudo dataset blueprint that can be used for comparing modules in a concatenation without regard for their

	originating libraries names. See also *SOODODS Specifics* following this table.
IDRLIST	Each library to be included in the IDRLIST blueprint must be specified using the INCLUDE keyword and then followed with the IDRLIST Keyword. Specifying the IDRLIST keyword will automatically launch the AMBLIST Batch Application Blueprinting Process.
SNPSHOT	Used to automatically launch an AMBLIST Batch Application Blueprinting Process in which structural configuration of a module is extracted and stored in the blueprint.
USEBSDN	Used to specify the specific blueprint member, other than the last stored member, that will be used during compare operations for a specific Blueprint type and specific ListId.

6.4.3 DSNLIST Specifics

Use the DSNLIST Keyword to define and name sequential datasets and/or PDS members that contain the actual Source Lists to be used by the Inspector. If the Datasets named using the DSNLIST keyword are located and available as defined and their contents are constructed using the syntax described below, the Inspector will extract and inspect list entries that have matching LISTIDs. All other List entries are ignored.

Syntax Model

```
Dataset.Sequential.&SYMBOLS/Dataset.Partitioned.&SYMBOLS
```

DSNLIST Model

```
KEYWORD ------DSNAME-----VOLUME LISTID
DSNLIST MY.LIST.DS(TWO)   SRC002 LSTTWO
DSNLIST MY.LIST.DS.ONE    SRC002 LSTONE
```

6.4.4 SOODODS Specifics

In the example below, processing occurred as normally expected with the addition of the creation and storing of a new blueprint member in a new blueprint dataset. This new Dataset will contain the NODE name SOODOMOD. As the new blueprint is built, the actual dataset name will be replaced with the pseudo name assigned by the SOODODS Keyword

SOODODS Model

```
INCLUDE FULL.QUALIFY.DATASET.ONE              VOLUME LISTID
SOODODS LPALIST.DATASET.ONE
INCLUDE FULL.QUALIFY.DATASET.TWO               VOLUME LISTID
SOODODS LPALIST.DATASET.TWO
```

.

6.5 Source List Examples

6.5.1 Default ISNLOAD Source List

```
****** ***************************** Top of Data ******************************
000001 -INC/EXC- -----------------DATASET NAME----------------- -VOLUME-LISTID-
000002 *AUTO*
```

6.5.2 Sample ISNLOAD Source List

The Sample ISNLOAD Source List shown below is best read in conjunction with the Option Descriptions that appear above.

```
BROWSE    IFO.IFOP.USERLIB(PLGLOAD) - 01.66          Line 00000000 Col 001 080
Command ===>                                                  Scroll ===> PAGE
********************************* Top of Data *********************************
*AUTO*
;LPALST
;LNKLST
;INSPECT IMAGMBR FOREGROUND
;DSNLIST MY.DATASET.LIST                            MYVOLS MYLIST
;INCLUDE SYS1.VTAMLIB                               S7RES1 TSTLST
;INCLUDE SYS1.LPALIB                                S7RES1 TSTLST
;SOODODS LPALIB.ONE
;IDRLIST
;SNPSHOT ACYAPC*
;INCLUDE ISP.SISPLPA                                S7RES1 TSTLST
;SOODODS LPALIB.TWO
;IDRLIST
;SNPSHOT FLMB FLMCMD FLMDDL FLMCXGPD
;INCLUDE TCPIP.SEZALPA                              S7RES3 TSTLST
;SOODODS LPALIB.THREE
;IDRLIST
;SNPSHOT *AUTO*
;MODNOTE 1) THIS SYSTEM IS SETUP TO RUN A LOAD MODULE COMPARE
;MODNOTE 2) ON SYSTEM IMAGMBR. YOU CAN CHANGE THE CONCATENATION
;MODNOTE 3) BY ADDING INCLUDE KEYWORD AS SHOWN ABOVE. IF YOU
;MODNOTE 4) WANT TO ADD TO THE CONCATENATION JUST COPY AND PASTE
;MODNOTE 5) AN INCLUDE LINE AND CHANGE THE LIBRARY AND VOLSER.
;USEBDSN IFO.IFOP.LOADMOD.IMAGMBR.TSTLST(M061019)
;USEBDSN IFO.IFOP.LOADSOD.IMAGMBR.TSTLST(S061019)
;USEBDSN IFO.IFOP.LOADIDR.IMAGMBR.TSTLST(#061019)
;USEBDSN IFO.IFOP.LOADSNP.IMAGMBR.TSTLST(#061019)
****************************** Bottom of Data ********************************
```

6.6 ISNLOAD Inspections

ISNLOAD performs two inspections. First, it identifies duplicate modules in a concatenation. Second, it identifies modules with orphaned aliases.

6.6.1 Duplicate Modules

A concatenation is a merger of the content (in this case the modules) of all the datasets defined to a dataset group, i.e. LPALST, LNKLST, VTAMLST. The end result of the concatenation will reside and be called from either memory or DASD. The order in which the datasets are defined within the group will determine the ultimate position of an individual module in the concatenation. When duplicates are present, confusion can arise as to which module is actually in use.

ISNLOAD identifies and reports duplicates within the concatenation.

6.6.2 Orphaned Aliases

A module may be defined by its author as being an alias for another module. Confusion and/or system failures may arise when the referenced module is not found in the concatenation.

ISNLOAD identifies and reports orphaned aliases.

6.7 ISNLOAD Blueprinting

ISNLOAD blueprinting is a process of identifying certain specific attributes of load modules, making an organized record of those attributes, and subsequently using those records as the basis for detecting changes in module aliasing, size, location, authorization, and optionally structure.

6.7.1 Blueprints Types

In order to accommodate the need to detect changes at different levels and under different circumstances, ISNLOAD can create four distinct types of blueprints. Each serves a specific purpose and some may be optionally used to fulfill specific needs.

The Default Blueprint Format

The Default ISNLOAD Blueprint is created using information derived from an examination of each module in a Library using standard system LMINIT functions. A sample of the format is shown below. Note that the blueprint is broken into two major sections. The first section is used to record Source Library Names, their Volumes, Module Counts and Date of Last Access. The second section is used to record (in their order of concatenation) modules, their aliases, size, TTRs, Authorization, AMO and RMO.

6.7.2 Sample Default ISNLOAD Blueprint

```
/******************************************************************************/
/* OPERATION= INSPECTION EXECUTION IS IN THE FOREGROUND.                      */
/* BLUEPRINT= IFO.IFOP.LOADMOD.IMAGMBR.LPALST(M090519)                        */
/* DATEBUILD= Tuesday, 19 Sep 2021 at 16:46:32                                */
/******************************************************************************/
/* DSNLST001= SYS1.LPALIB                                S7RES1  1671 2021/092*/
/* DSNLST002= USER.LPALIB                                S7SYS1     0 2021/092*/
/* DSNLST003= ADCD.Z17S.LPALIB                           S7RES1     3 2021/092*/
/* DSNLST004= EQA610.SEQALPA                             S7RES2     4 2021/092*/
/* DSNLST005= SYS1.SERBLPA                               S7RES1    51 2021/092*/
/* DSNLST006= NET520.SCNMLPA1                            S7RES2     7 2021/092*/
/* DSNLST007= FAN140.SEAGLPA                             S7RES2     9 2021/092*/
/* DSNLST008= ISF.SISFLPA                                S7RES1     2 2021/092*/
/* DSNLST017= AUT310.SINGMOD3                            S7RES2    11 2021/092*/
/**** END DATASET LIST ****************************************  2027 *********/
-Cat- -------------- Full Module Path -------------- -Alias-- --Size-- -TTRs- AC
00001 SYS1.LPALIB(ACYAPCIP)                          ACYAPCNP 00007338 00760C 00
00002 SYS1.LPALIB(ACYAPCNP)                          -------- 00007338 00760C 00
00003 SYS1.LPALIB(ACYAPCPP)                          ACYAPCNP 00007338 00760C 00
00004 SYS1.LPALIB(ACYAPDRP)                          ACYAPCNP 00007338 00760C 00
00005 SYS1.LPALIB(ACYAPD1P)                          ACYAPCNP 00007338 00760C 00
00006 SYS1.LPALIB(ACYAPFLP)                          ACYAPCNP 00007338 00760C 00
00007 SYS1.LPALIB(ACYAPMAP)                          ACYAPCNP 00007338 00760C 00
00008 SYS1.LPALIB(ACYAPQCP)                          ACYAPCNP 00007338 00760C 00
00009 SYS1.LPALIB(ACYAPQRP)                          ACYAPCNP 00007338 00760C 00
00010 SYS1.LPALIB(ACYAPRGP)                          ACYAPCNP 00007338 00760C 00
```

The Pseudo Blueprint Format

The format is the same as the Default Blueprint Format with one exception. The Blueprint is driven by the use of the SOODODS Keyword, allowing for the replacement of the actual library name in the second section of the Blueprint to be replaced with the value of SOODODS. This can be useful during software upgrades when library names are changing but module names, for the most part, remain the same.

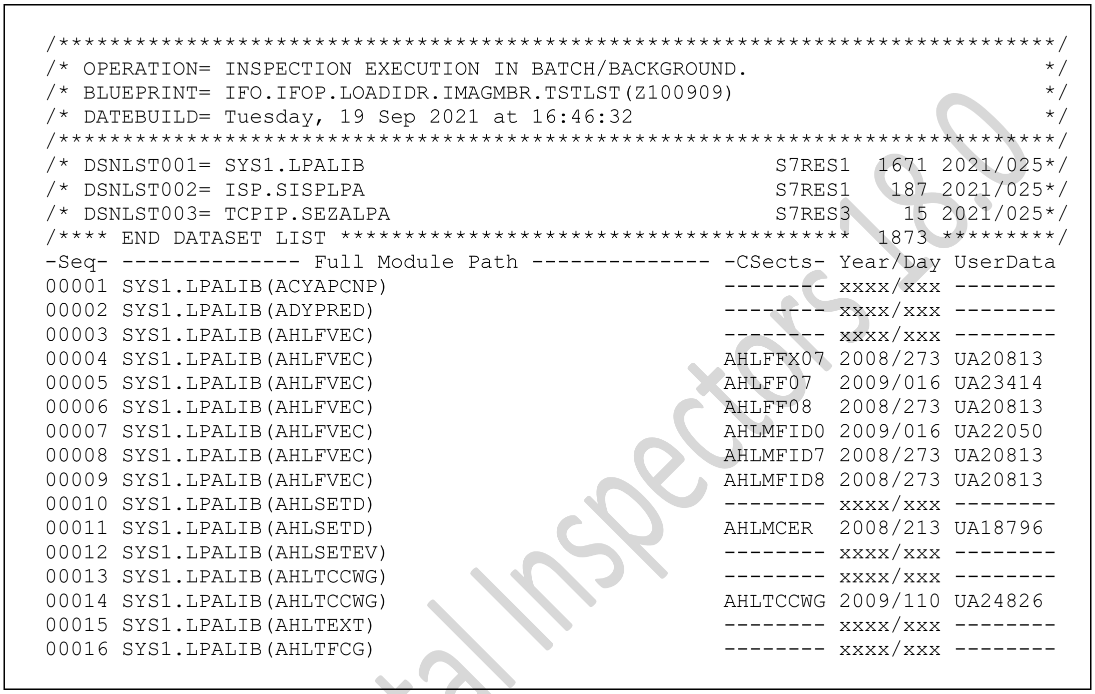
Sample Pseudo Blueprint

```
/*****************************************************************************/
/* OPERATION= INSPECTION EXECUTION IS IN THE FOREGROUND.                      */
/* BLUEPRINT= IFO.IFOP.LOADMOD.IMAGMBR.LPALST(M090519)                        */
/* DATEBUILD= Tuesday, 19 Sep 2021 at 16:46:32                                */
/*****************************************************************************/
/* DSNLST001= SYS1.LPALIB                                S7RES1  1671 2021/092*/
/* DSNLST002= USER.LPALIB                                S7SYS1     0 2021/092*/
/* DSNLST003= ADCD.Z17S.LPALIB                           S7RES1     3 2021/092*/
/* DSNLST004= EQA610.SEQALPA                             S7RES2     4 2021/092*/
/* DSNLST005= SYS1.SERBLPA                               S7RES1    51 2021/092*/
/* DSNLST006= NET520.SCNMLPA1                            S7RES2     7 2021/092*/
/* DSNLST007= FAN140.SEAGLPA                             S7RES2     9 2021/092*/
/* DSNLST008= ISF.SISFLPA                                S7RES1     2 2021/092*/
/* DSNLST017= AUT310.SINGMOD3                            S7RES2    11 2021/092*/
/**** END DATASET LIST ***************************************  2027 *********/
-Cat- -------------- Full Module Path -------------- -Alias-- --Size-- -TTRs- AC
00001 DATASET.ONE(ACYAPCIP)                         ACYAPCNP 00007338 00760C 00
00002 DATASET.ONE(ACYAPCNP)                         -------- 00007338 00760C 00
00003 DATASET.ONE(ACYAPCPP)                         ACYAPCNP 00007338 00760C 00
00004 DATASET.ONE(ACYAPDRP)                         ACYAPCNP 00007338 00760C 00
00005 DATASET.ONE(ACYAPD1P)                         ACYAPCNP 00007338 00760C 00
00006 DATASET.ONE(ACYAPFLP)                         ACYAPCNP 00007338 00760C 00
00007 DATASET.ONE(ACYAPMAP)                         ACYAPCNP 00007338 00760C 00
00008 DATASET.ONE(ACYAPQCP)                         ACYAPCNP 00007338 00760C 00
00009 DATASET.ONE(ACYAPQRP)                         ACYAPCNP 00007338 00760C 00
00010 DATASET.ONE(ACYAPRGP)                         ACYAPCNP 00007338 00760C 00
```

The CSECT Profile Blueprint Format

This blueprint format is created when the IDRLIST Keyword is used. The process used to build the blueprint called AMBLIST extracts one or more CSECT Profiles from each module.

Sample CSECT Profile Blueprint

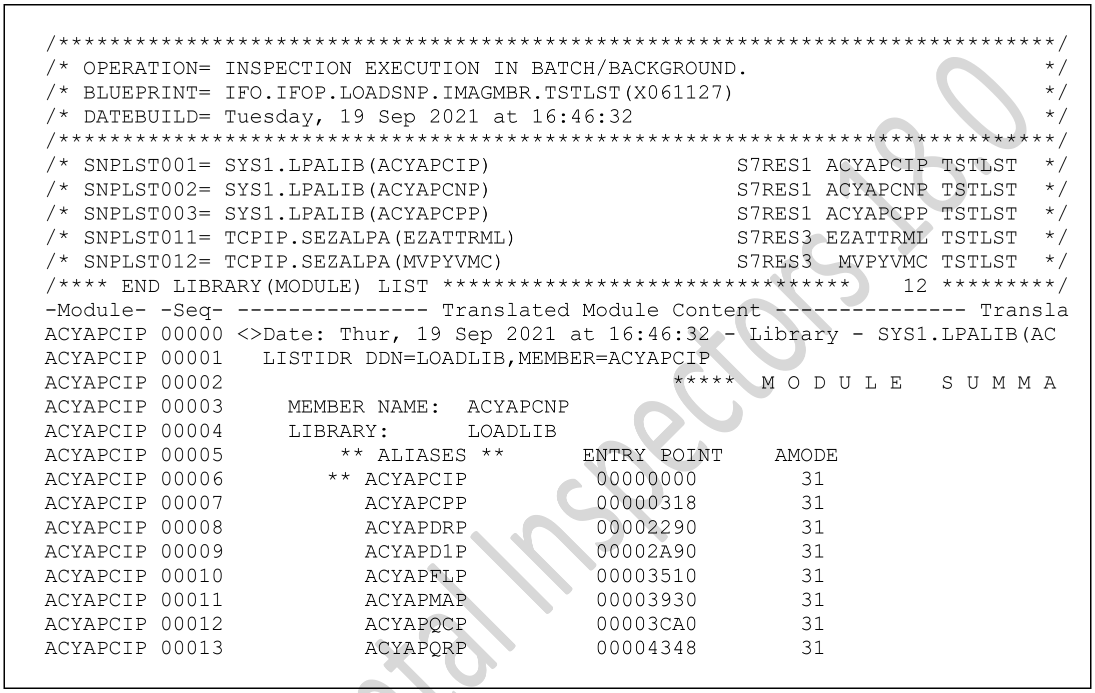


```
/******************************************************************************/
/* OPERATION= INSPECTION EXECUTION IN BATCH/BACKGROUND.                       */
/* BLUEPRINT= IFO.IFOP.LOADIDR.IMAGMBR.TSTLST(Z100909)                        */
/* DATEBUILD= Tuesday, 19 Sep 2021 at 16:46:32                                */
/******************************************************************************/
/* DSNLST001= SYS1.LPALIB                                S7RES1  1671 2021/025*/
/* DSNLST002= ISP.SISPLPA                                S7RES1   187 2021/025*/
/* DSNLST003= TCPIP.SEZALPA                              S7RES3    15 2021/025*/
/**** END DATASET LIST ****************************************  1873 *********/
-Seq- -------------- Full Module Path -------------- -CSects- Year/Day UserData
00001 SYS1.LPALIB(ACYAPCNP)                          -------- xxxx/xxx --------
00002 SYS1.LPALIB(ADYPRED)                           -------- xxxx/xxx --------
00003 SYS1.LPALIB(AHLFVEC)                           -------- xxxx/xxx --------
00004 SYS1.LPALIB(AHLFVEC)                           AHLFFX07 2008/273 UA20813
00005 SYS1.LPALIB(AHLFVEC)                           AHLFF07  2009/016 UA23414
00006 SYS1.LPALIB(AHLFVEC)                           AHLFF08  2008/273 UA20813
00007 SYS1.LPALIB(AHLFVEC)                           AHLMFID0 2009/016 UA22050
00008 SYS1.LPALIB(AHLFVEC)                           AHLMFID7 2008/273 UA20813
00009 SYS1.LPALIB(AHLFVEC)                           AHLMFID8 2008/273 UA20813
00010 SYS1.LPALIB(AHLSETD)                           -------- xxxx/xxx --------
00011 SYS1.LPALIB(AHLSETD)                           AHLMCER  2008/213 UA18796
00012 SYS1.LPALIB(AHLSETEV)                          -------- xxxx/xxx --------
00013 SYS1.LPALIB(AHLTCCWG)                          -------- xxxx/xxx --------
00014 SYS1.LPALIB(AHLTCCWG)                          AHLTCCWG 2009/110 UA24826
00015 SYS1.LPALIB(AHLTEXT)                           -------- xxxx/xxx --------
00016 SYS1.LPALIB(AHLTFCG)                           -------- xxxx/xxx --------
```

The Module Snapshot Blueprint Format

This blueprint format is created when the SNPSHOT Keyword is used. The process used to build the blueprint called AMBLIST extracts and records a structural Profile from each module.

Sample Module Snapshot Blueprint

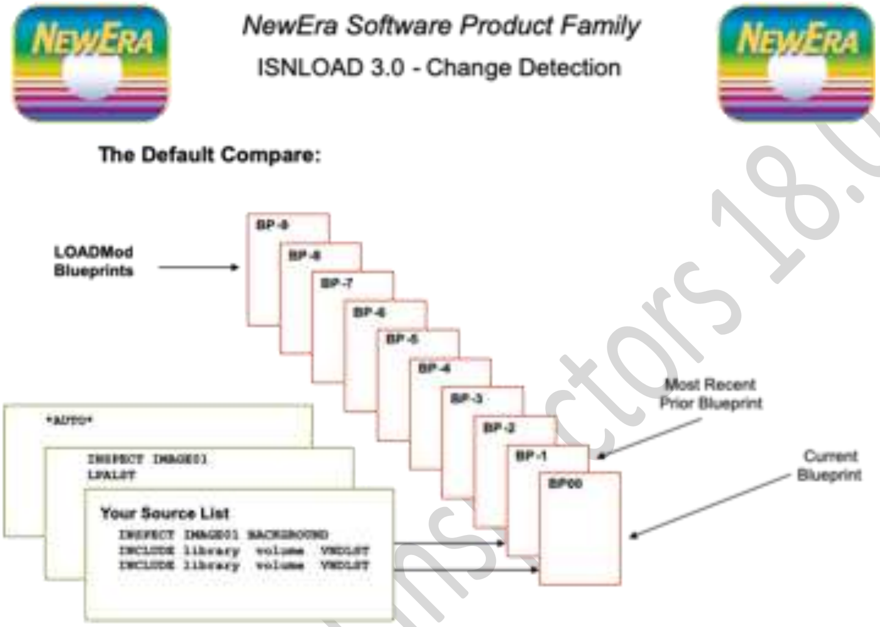


```
/******************************************************************************/
/* OPERATION= INSPECTION EXECUTION IN BATCH/BACKGROUND.                       */
/* BLUEPRINT= IFO.IFOP.LOADSNP.IMAGMBR.TSTLST(X061127)                        */
/* DATEBUILD= Tuesday, 19 Sep 2021 at 16:46:32                                */
/******************************************************************************/
/* SNPLST001= SYS1.LPALIB(ACYAPCIP)                   S7RES1 ACYAPCIP TSTLST  */
/* SNPLST002= SYS1.LPALIB(ACYAPCNP)                   S7RES1 ACYAPCNP TSTLST  */
/* SNPLST003= SYS1.LPALIB(ACYAPCPP)                   S7RES1 ACYAPCPP TSTLST  */
/* SNPLST011= TCPIP.SEZALPA(EZATTRML)                 S7RES3 EZATTRML TSTLST  */
/* SNPLST012= TCPIP.SEZALPA(MVPYVMC)                  S7RES3  MVPYVMC TSTLST  */
/**** END LIBRARY(MODULE) LIST *******************************   12 *********/
-Module- -Seq- --------------- Translated Module Content --------------- Transla
ACYAPCIP 00000 <>Date: Thur, 19 Sep 2021 at 16:46:32 - Library - SYS1.LPALIB(AC
ACYAPCIP 00001   LISTIDR DDN=LOADLIB,MEMBER=ACYAPCIP
ACYAPCIP 00002                                   *****  M O D U L E   S U M M A
ACYAPCIP 00003     MEMBER NAME:  ACYAPCNP
ACYAPCIP 00004     LIBRARY:      LOADLIB
ACYAPCIP 00005         ** ALIASES **      ENTRY POINT    AMODE
ACYAPCIP 00006        ** ACYAPCIP          00000000        31
ACYAPCIP 00007           ACYAPCPP          00000318        31
ACYAPCIP 00008           ACYAPDRP          00002290        31
ACYAPCIP 00009           ACYAPD1P          00002A90        31
ACYAPCIP 00010           ACYAPFLP          00003510        31
ACYAPCIP 00011           ACYAPMAP          00003930        31
ACYAPCIP 00012           ACYAPQCP          00003CA0        31
ACYAPCIP 00013           ACYAPQRP          00004348        31
```

6.8 Comparing ISNLOAD Blueprints

6.8.1 Compare Process Types

Default Compare



NewEra Software Product Family

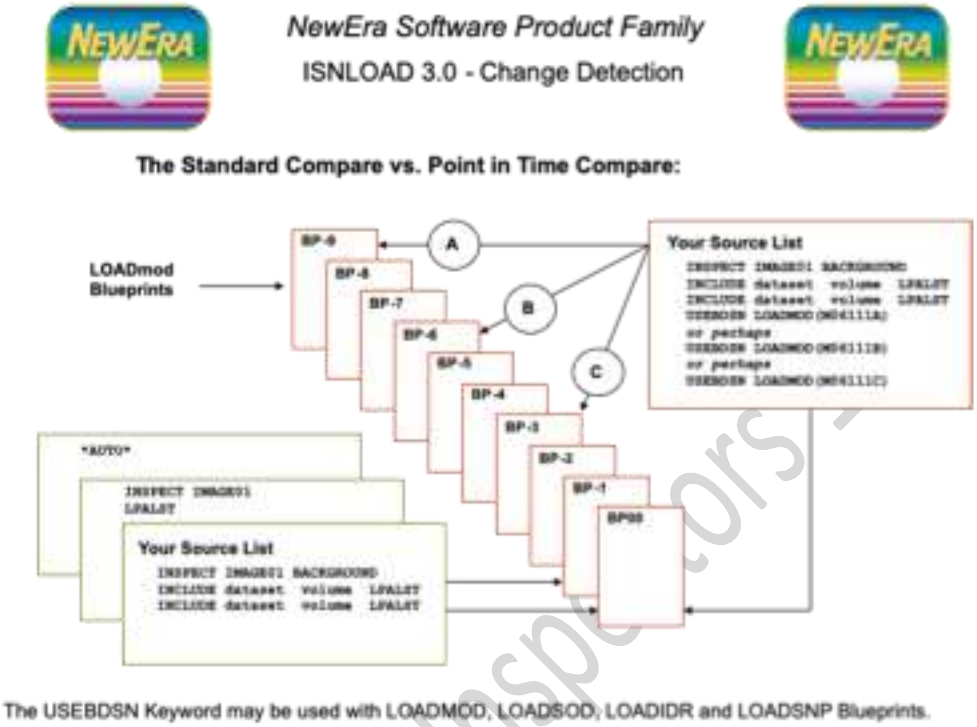
ISNLOAD 3.0 - Change Detection

The Default Compare:

```
*AUTO*

INSPECT IMAGE01
LPALST

Your Source List
  INSPECT IMAGE01 BACKGROUND
  INCLUDE library  volume  VNDLST
  INCLUDE library  volume  VNDLST
```

The Default Compare process compares the Current Blueprint against the most recent Prior Blueprint.

Point in time compares



NewEra Software Product Family

ISNLOAD 3.0 - Change Detection

The Standard Compare vs. Point in Time Compare:

Your Source List

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset  volume  LPALST
INCLUDE dataset  volume  LPALST
USEBDSN LOADMOD(MD41111A)
or perhaps
USEBDSN LOADMOD(MD41111B)
or perhaps
USEBDSN LOADMOD(MD41111C)
```

```
*AUTO*
```

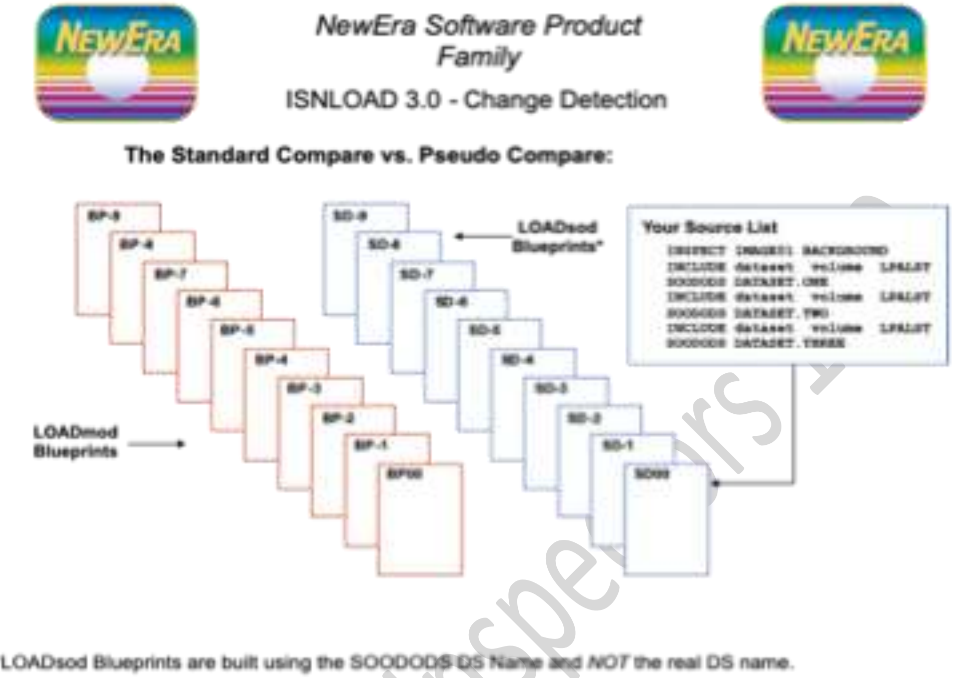
```
INSPECT IMAGE01
LPALST
```

Your Source List

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset  volume  LPALST
INCLUDE dataset  volume  LPALST
```

The USEBDSN Keyword may be used with LOADMOD, LOADSOD, LOADIDR and LOADSNP Blueprints.

Pseudo Compares



NewEra Software Product Family

ISNLOAD 3.0 - Change Detection

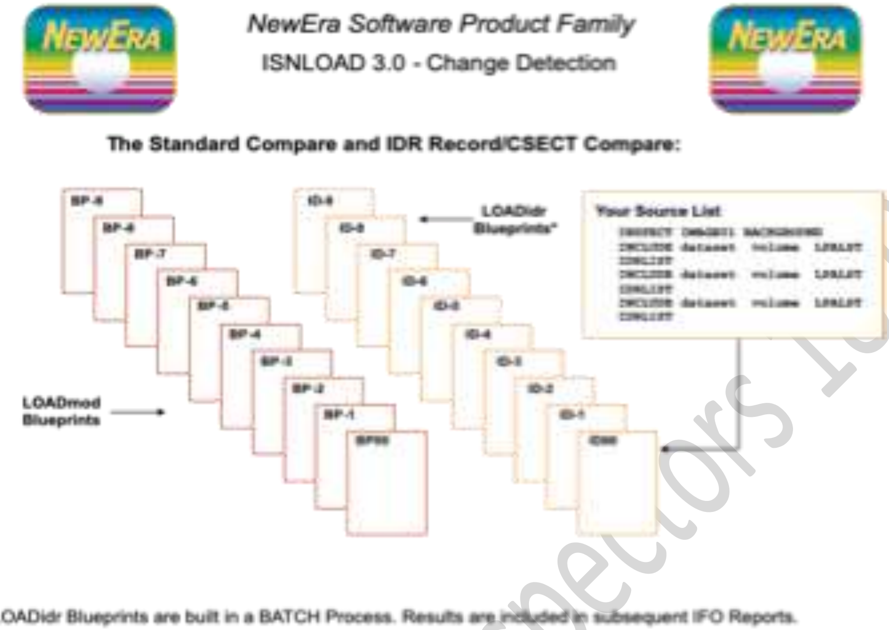
The Standard Compare vs. Pseudo Compare:

Your Source List

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset  volume  LPALST
SOODODS DATASET.ONE
INCLUDE dataset  volume  LPALST
SOODODS DATASET.TWO
INCLUDE dataset  volume  LPALST
SOODODS DATASET.THREE
```

*LOADsod Blueprints are built using the SOODODS DS Name and *NOT* the real DS name.

IDR Record/Csect Compare



NewEra Software Product Family

ISNLOAD 3.0 - Change Detection

The Standard Compare and IDR Record/CSECT Compare:

Your Source List

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset volume LPALST
IDRLIST
INCLUDE dataset volume LPALST
IDRLIST
INCLUDE dataset volume LPALST
IDRLIST
```

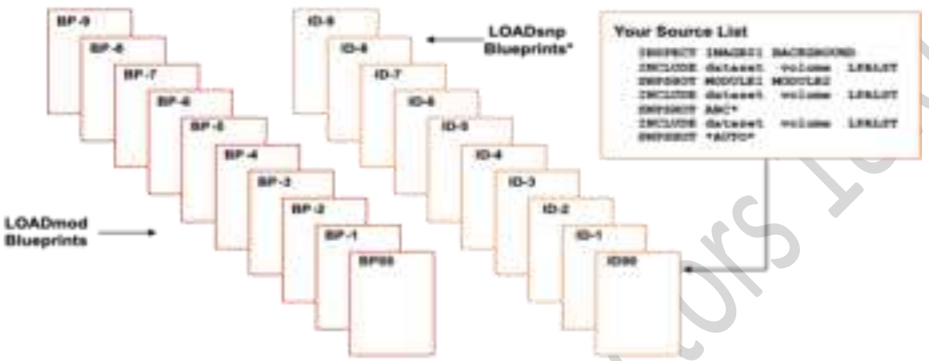
*LOADidr Blueprints are built in a BATCH Process. Results are included in subsequent IFO Reports.

Module Structural Compare

NewEra Software Product Family

ISNLOAD 3.0 - Change Detection

The Standard Compare and Module/Object Structural Compare:



Your Source List

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset  volume  LPALST
SNPSHOT MODULE1 MODULE2
INCLUDE dataset  volume  LPALST
SNPSHOT ABC*
INCLUDE dataset  volume  LPALST
SNPSHOT *AUTO*
```

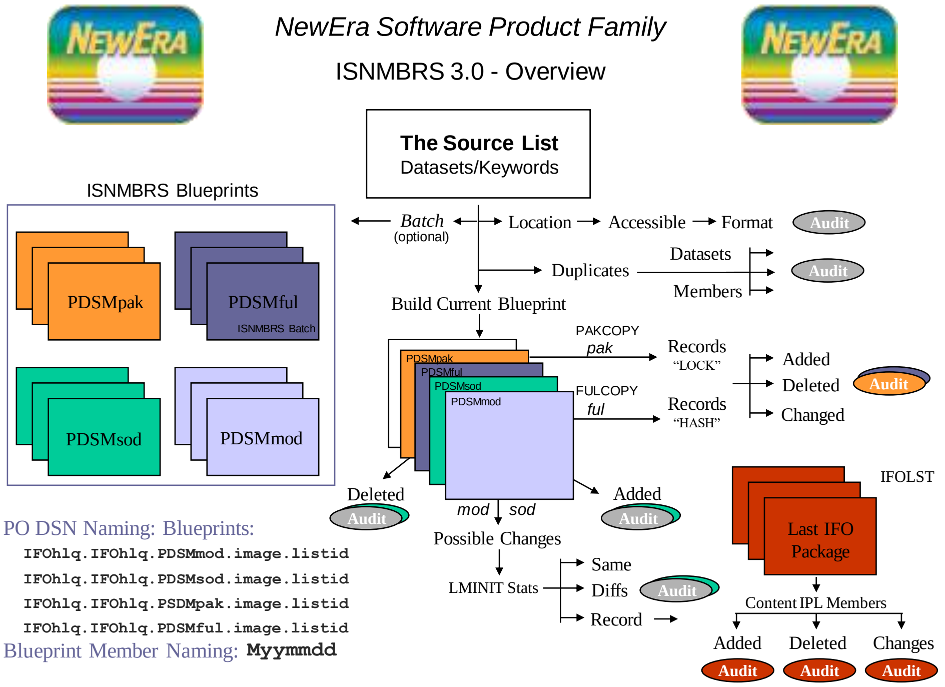
*LOADsnp Blueprints are built in a BATCH Process. Results are included in subsequent IFO Reports.

7 Working with ISNMBRS

ISNMBRS is a z/OS software application that can be used effectively to identify and track changes and/or anomalies in partitioned datasets and the members that they contain.

7.1 Application Overview

Members in Partitioned Datasets are the primary source z/OS configuration components, i.e. ParmLib, ProcLib and VTAMLib. Members that do not prevail during an IPL are often used later during *Dynamic System Updates*. This Supplemental Inspector evaluates the content of one or more Dataset Control Point concatenations for z/OS and non-z/OS members, blueprints the concatenation, detects changes in its composition, and reports its findings as directed.



7.2 Controlled Datasets and Source List

A Controlled Dataset candidate is any dataset that is of valued importance to your understanding of the integrity and fitness of the z/OS system environment. Candidate datasets become Controlled Datasets when they are defined to the Integrity Controls Environment and/or its Applications using a Dataset Control/Source List.

7.3 Configuring the ISNMBRS application

ISNMBRS, like all the other Supplemental Inspectors, shares certain common shared configuration definitions. In addition to those common shared definitions, ISNMBRS responds to its own unique configuration definition set.

7.3.1 ISNMBRS Configuration Options

Keyword	Default/Optional	Functional Description
MBRLIST=	AUTO/NONE	Lists the content of the Inspection Source List in the Inspection Report.
MBRPRSS=	AUTO/NONE	Lists inspection process steps in the Inspection Report
MBRDUPS=	AUTO/NONE	Identifies and Reports Duplicate Members in a Dataset Concatenation (Source List)
MBRPACK=	AUTO/NONE	Identifies and displays the changes in the Image FOCUS Packages
MBRDELT=	AUTO/NONE	Used to turn ON/OFF Blueprint Comparison and Change Detection
CLEANUP=	AUTO/NONE	Used as needed to provide a detailed analysis of the content of full ParmLib Concatenation.
VERBOSE=	AUTO/NONE	Used to reduce the amount of output that appears in the final Inspection Report.
STATSON=	AUTO/NONE	Automatically turns on ISPF Stats during an inspection when it is discovered that they are turned off. This function may require special access authority as turning on Stats will appear to the system as an attempted WRITE.

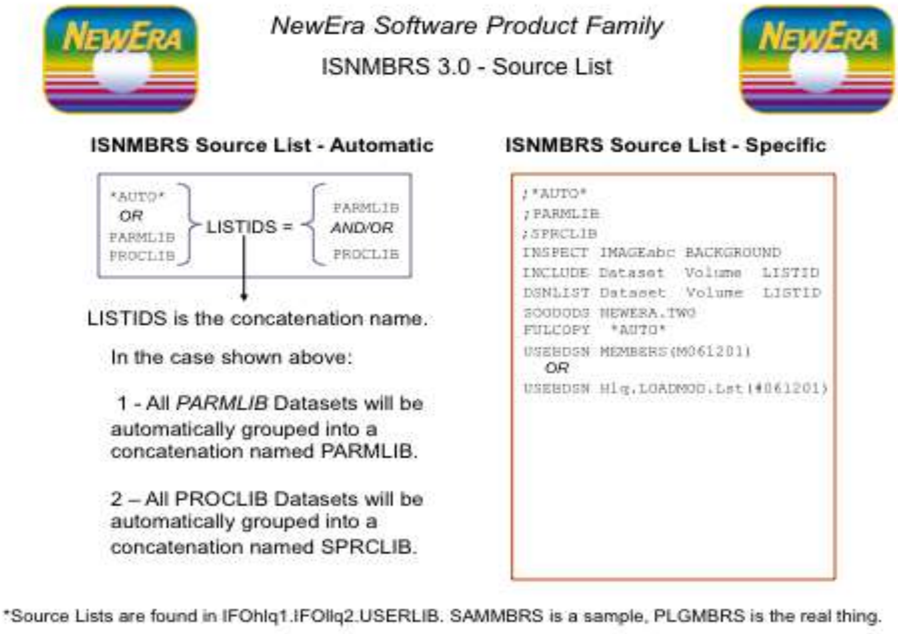
MBRMDOC=	MBRSUMS	Names the ISNMBRS Report to be sent when Email Notification is active.

7.3.2 Creating an ISNMBRS Source List

When using ISNMBRS you have a choice – run “Out of the Box” and let Image FOCUS discover the ParmLib and ProcLib for each Image defined for inspection or customize the ISNMBRS Source List as described below. The ISNMBRS Source List is member PLGMBRS and can be found (along with a sample SAMMBRS) in the Integrity Controls Environment USERLIB. PLGMBRS can be edited directly under TSO and/or Image FOCUS and/or via the optional Supplemental In-Line Interface.

7.3.3 ISNMBRS Source List Overview

The Source List tells the Inspector what to Inspect and Blueprint. Its content could be as simple as a single dataset or list of datasets that in some way are related. In a more complex form, the Source List contains special blueprinting instructions that specify the level of detail to be included in a blueprint and/or what exactly is to be compared when detecting changes.



NewEra Software Product Family

ISNMBRS 3.0 - Source List

ISNMBRS Source List - Automatic

LISTIDS is the concatenation name.

In the case shown above:

1 - All *PARMLIB* Datasets will be automatically grouped into a concatenation named PARMLIB.

2 – All PROCLIB Datasets will be automatically grouped into a concatenation named SPRCLIB.

ISNMBRS Source List - Specific

```
;*AUTO*
;PARMLIB
;SPRCLIB
INSPECT IMAGEabc BACKGROUND
INCLUDE Dataset  Volume  LISTID
DSNLIST Dataset  Volume  LISTID
SOODODS NEWERA.TWO
FULCOPY  *AUTO*
USEBDSN MEMBERS(M061201)
  OR
USEBDSN Hlq.LOADMOD.Lst(#061201)
```

*Source Lists are found in IFOhlq1.IFOllq2.USERLIB. SAMMBRS is a sample, PLGMBRS is the real thing.

7.3.4 ISNMBRS Source List Options

The following table details the ISNMBRS Source List Keywords and their possible values.

Keyword	Default/Optional	Functional Description
AUTO		Tells ISNMBRS to query Image FOCUS for the PARMLIB and PROCLIB for named IMAGE
IPLLST		Tells ISNMBRS to query Image FOCUS for the IPLPARM only for named IMAGE
PRMLST		Tells ISNMBRS to query Image FOCUS for the PARMLIB only for named IMAGE
SPRLST		Tells ISNMBRS to query Image FOCUS for the PROCLIB only for named IMAGE
INSPECT	Image_name	Allows for multiple Image definitions in a single Source List
DSNLIST	Dataset Volume ListId	Used, line by line, to tell the inspector to acquire its Source List from the dataset/volume named. Only INCLUDE entries within the named dataset that match ListId will be included in the Inspection. See also the ***DSNLIST Specifics*** that follow this table.
INCLUDE	Dataset Volume ListId	Defines the name and location of Libraries to be included in a named concatenation. The value of ListId will be used as the concatenation name and the 4th NODE of the Blueprint Dataset. Repeat the INCLUDE Keyword for each Library to be included in a Concatenation. To begin a new concatenation, specify a new and different ListId value.

SOODODS	Pseudo Dataset Name	Used in conjunction with the INCLUDE keyword to create a pseudo dataset blueprint that can be used for comparing modules in a concatenation without regard for their originating libraries names. See also ***SOODODS Specifics*** following this table.
FULLCOPY		A full copy of each member in the target dataset(s) is included in the blueprint.
USEBSDN		Used to specify the specific blueprint member, other than the last stored member, that will be used during compare operations for a specific Blueprint type and specific ListId.

7.3.5 DSNLIST Specifics

Use the DSNLIST Keyword to define and name sequential datasets and/or PDS members that contain the actual Source Lists to be used by the Inspector. If the Datasets named using the DSNLIST keyword are located and available as defined and their contents are constructed using the syntax described below, the Inspector will extract and inspect list entries that have matching LISTIDs. All other List entries are ignored.

Syntax Model

```
Dataset.Sequential.&SYMBOLS/Dataset.Partitioned.&SYMBOLS
```

DSNLIST Model

```
KEYWORD ------DSNAME-----VOLUME LISTID
DSNLIST MY.LIST.DS(TWO)  SRC002 LSTTWO
DSNLIST MY.LIST.DS.ONE   SRC002 LSTONE
```

7.3.6 SOODODS Specifics

In the example below, processing occurred as normally expected with the addition of the creation and storing of a new blueprint member in a new blueprint dataset. This new Dataset will contain the NODE name SOODOMOD. As the new blueprint is built, the actual dataset name will be replaced with the pseudo name assigned by the SOODODS Keyword

SOODODS Model

```
INCLUDE FULL.QUALIFY.DATASET.ONE                  VOLUME LISTID
SOODODS LPALIST.DATASET.ONE
INCLUDE FULL.QUALIFY.DATASET.TWO                  VOLUME LISTID
SOODODS LPALIST.DATASET.TWO
```

7.4 Source List Examples

7.4.1 Default ISNMBRS Source List

```
****** ***************************** Top of Data ******************************
000001 -INC/EXC- -----------------DATASET NAME----------------- -VOLUME-LISTID-
000002 *AUTO*
```

7.4.2 Sample ISNMBRS Source List

```
****** ***************************** Top of Data ******************************
000001 -INC/EXC- -----------------DATASET NAME----------------- -VOLUME-LISTID-
000002 INSPECT  IMAGMBR
000003 ;PAKCOPY
000004 ;INCLUDE  GBAGS2.IFO.INSTLIB                              USR002 GHBLST
000005 ;SOODODS  PAUL.ONE
000006 INCLUDE  IFO.IFO6M.INSTLIB                                USR003 GHBLST
000007 ;SOODODS  PAUL.TWO
000008 ;FULCOPY  *AUTO*
000009 ;FULCOPY IS USED TO MAKE A CONTENT COPY OF MEMBERS IN A DATASET
000010 ;PAKCOPY IS USED WHEN YOU WANT ISNMBRS TO BLUEPRINT IFO PACKAGES.
000011 ;PAKNOTE WHEN YOU REVIEW A PAKCOPY REPORT BE CERTAIN TO REPORT ANY
000012 ;PAKNOTE PROBLEMS TO THE SYSTEM PROGRAM STAFF.
000013 ;USEBSDN IFO.IFOP.PDSMBRS.IMAGMBR.GHBLST(#070117)
```

7.5 ISNMBRS Inspection

During inspection processing, ISNMBRS will check each member in the target dataset to determine if ISPF STATS are turned ON. If not ON, a warning is reported. Optionally, If STATS are not ON, the STATSON keyword may be set to “AUTO” to force ISNMBRS to automatically set them to ON. This optional function may require special access permissions, as it will be viewed by the system as a WRITE to all affected members.

7.6 ISNMBRS Blueprinting

ISNMBRS blueprinting is a process of identifying certain specific attributes of members in partitioned datasets, making an organized record of those attributes and subsequently using those records as the basis for detecting changes in a member’s statistical state and optionally structure.

7.6.1 Blueprints Types

In order to accommodate the need to detect changes at different levels and under different circumstances, ISNMBRS can create four distinct types of blueprints. Each serves a specific purpose and some may be optionally used to fulfill a specific need.

The Default Blueprint Format

The Default ISNMBRS Blueprint is created using information derived from an examination of each member in a dataset using standard system LMINIT functions. A sample of the format is shown below. Note that the blueprint is broken into two major sections. The first section is used to record Source Dataset Names, their Volumes, Member Counts and Date of Last Access. The second section is used to record, in their order of concatenation, members, their statistics (if any) and current record count.

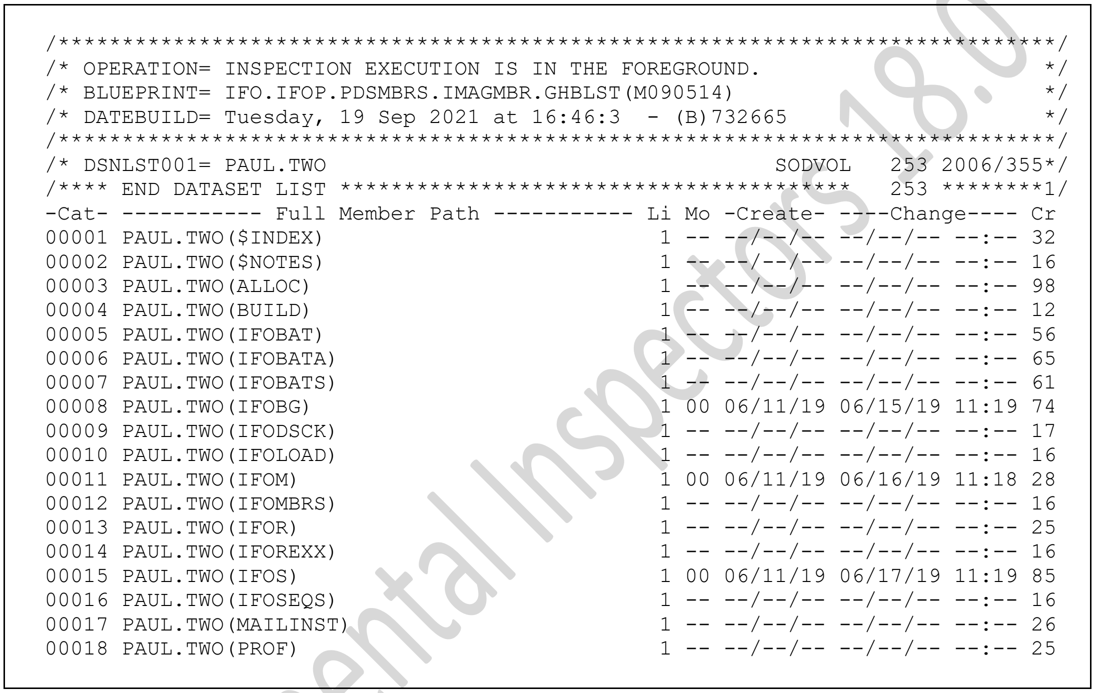
7.6.2 Sample Default ISNMBRS Blueprint

```
/****************************************************************************/
/* OPERATION= INSPECTION EXECUTION IS IN THE FOREGROUND.                      */
/* BLUEPRINT= IFO.IFOP.PDSMBRS.IMAGMBR.GHBLST(M090514)                        */
/* DATEBUILD= Tuesday, 19 Sep 2021 at 16:46:32 - (B)732665                    */
/****************************************************************************/
/* DSNLST001= IFO.IFO6M.INSTLIB                          USR003   253 2006/355*/
/**** END DATASET LIST ****************************************   253 *********/
-Cat- ----------- Full Member Path ----------- Li Mo -Create- ----Change---- Cr
00001 IFO.IFO6M.INSTLIB($INDEX)                 1 -- --/--/-- --/--/-- --:-- 32
00002 IFO.IFO6M.INSTLIB($NOTES)                 1 -- --/--/-- --/--/-- --:-- 16
00003 IFO.IFO6M.INSTLIB(ALLOC)                  1 -- --/--/-- --/--/-- --:-- 98
00004 IFO.IFO6M.INSTLIB(BUILD)                  1 -- --/--/-- --/--/-- --:-- 12
00005 IFO.IFO6M.INSTLIB(IFOBAT)                 1 -- --/--/-- --/--/-- --:-- 56
00006 IFO.IFO6M.INSTLIB(IFOBATA)                1 -- --/--/-- --/--/-- --:-- 65
00007 IFO.IFO6M.INSTLIB(IFOBATS)                1 -- --/--/-- --/--/-- --:-- 61
00008 IFO.IFO6M.INSTLIB(IFOBG)                  1 00 06/11/19 06/15/19 11:19 74
00009 IFO.IFO6M.INSTLIB(IFODSCK)                1 -- --/--/-- --/--/-- --:-- 19
00010 IFO.IFO6M.INSTLIB(IFOLOAD)                1 -- --/--/-- --/--/-- --:-- 16
00011 IFO.IFO6M.INSTLIB(IFOM)                   1 00 06/11/19 06/16/19 11:18 28
00012 IFO.IFO6M.INSTLIB(IFOMBRS)                1 -- --/--/-- --/--/-- --:-- 16
00013 IFO.IFO6M.INSTLIB(IFOR)                   1 -- --/--/-- --/--/-- --:-- 25
00014 IFO.IFO6M.INSTLIB(IFOREXX)                1 -- --/--/-- --/--/-- --:-- 16
00015 IFO.IFO6M.INSTLIB(IFOS)                   1 00 06/11/19 06/19/19 11:19 85
00016 IFO.IFO6M.INSTLIB(IFOSEQS)                1 -- --/--/-- --/--/-- --:-- 16
00019 IFO.IFO6M.INSTLIB(MAILINST)               1 -- --/--/-- --/--/-- --:-- 26
00018 IFO.IFO6M.INSTLIB(PROF)                   1 -- --/--/-- --/--/-- --:-- 25
```

The Pseudo Blueprint Format

The format is the same as the Default Blueprint Format with one exception. The Blueprint is driven by the use of the SOODODS Keyword allowing for the replacement of the actual dataset named in the second section of the Blueprint to be replaced with the value of SOODODS. This can be useful during software upgrades when dataset names are changing but the member name, for the most part, remains the same.

Sample Pseudo Blueprint



```
/******************************************************************************/
/* OPERATION= INSPECTION EXECUTION IS IN THE FOREGROUND.                      */
/* BLUEPRINT= IFO.IFOP.PDSMBRS.IMAGMBR.GHBLST(M090514)                        */
/* DATEBUILD= Tuesday, 19 Sep 2021 at 16:46:3  - (B)732665                    */
/******************************************************************************/
/* DSNLST001= PAUL.TWO                                  SODVOL   253 2006/355*/
/**** END DATASET LIST ***************************************   253 ********1/
-Cat- ----------- Full Member Path ----------- Li Mo -Create- ----Change---- Cr
00001 PAUL.TWO($INDEX)                          1 -- --/--/-- --/--/-- --:-- 32
00002 PAUL.TWO($NOTES)                          1 -- --/--/-- --/--/-- --:-- 16
00003 PAUL.TWO(ALLOC)                           1 -- --/--/-- --/--/-- --:-- 98
00004 PAUL.TWO(BUILD)                           1 -- --/--/-- --/--/-- --:-- 12
00005 PAUL.TWO(IFOBAT)                          1 -- --/--/-- --/--/-- --:-- 56
00006 PAUL.TWO(IFOBATA)                         1 -- --/--/-- --/--/-- --:-- 65
00007 PAUL.TWO(IFOBATS)                         1 -- --/--/-- --/--/-- --:-- 61
00008 PAUL.TWO(IFOBG)                           1 00 06/11/19 06/15/19 11:19 74
00009 PAUL.TWO(IFODSCK)                         1 -- --/--/-- --/--/-- --:-- 17
00010 PAUL.TWO(IFOLOAD)                         1 -- --/--/-- --/--/-- --:-- 16
00011 PAUL.TWO(IFOM)                            1 00 06/11/19 06/16/19 11:18 28
00012 PAUL.TWO(IFOMBRS)                         1 -- --/--/-- --/--/-- --:-- 16
00013 PAUL.TWO(IFOR)                            1 -- --/--/-- --/--/-- --:-- 25
00014 PAUL.TWO(IFOREXX)                         1 -- --/--/-- --/--/-- --:-- 16
00015 PAUL.TWO(IFOS)                            1 00 06/11/19 06/17/19 11:19 85
00016 PAUL.TWO(IFOSEQS)                         1 -- --/--/-- --/--/-- --:-- 16
00017 PAUL.TWO(MAILINST)                        1 -- --/--/-- --/--/-- --:-- 26
00018 PAUL.TWO(PROF)                            1 -- --/--/-- --/--/-- --:-- 25
```

The Pack Copy Blueprint

Using the optional PAKCOPY Keyword allows ISNMBRS to automatically Blueprint Image FOCUS created packages. PAKCOPY Blueprint Operations can be defined to execute in the Foreground or Background or in both. PAKCOPY Compare Operations support the use of the USEBSDN Keyword Functions. The optional #NEWPAK compares the last Blueprint against the Member content as currently found.

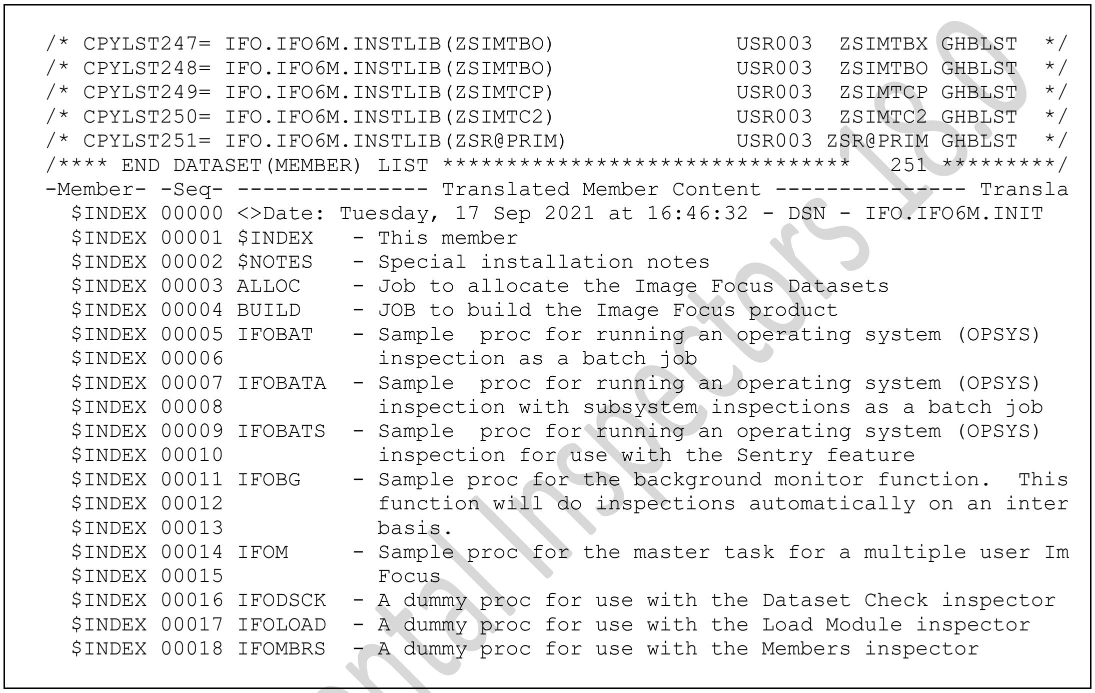
Sample Pack Copy Blueprint

```
/*****************************************************************************/
/* OPERATION= INSPECTION EXECUTION IS IN THE FOREGROUND.                     */
/* BLUEPRINT= IFO.IFOP.PDSMPAK.IMAGMBR.PAKCPY(Z090512)                       */
/* DATEBUILD= Tuesday, 17 Sep 2021 at 16:46:32                               */
/********M********************************************************************/
/* DSNLST001= SYS1.IPLPARM                                    S7SYS1 0AB3    */
/* DSNLST002= ADCD.Z17S.PARMLIB                               S7RES1 0A91    */
/* DSNLST003= USER.PARMLIB                                    USR001 0A91    */
/**** END DATASET LIST ***************************************     3 *********/
/*****************************************************************************/
/* MBRLST001= SYS1.IPLPARM(LOADS7)                    21/08/23 18:54 ADCDMST */
/* MBRLST002= ADCD.Z17S.PARMLIB(IEASYM00)             21/04/13 12:17 IBMUSER */
/* MBRLST003= ADCD.Z17S.PARMLIB(IEASYS00)             21/04/13 12:17 IBMUSER */
/* MBRLST031= ADCD.Z17S.PARMLIB(SMFPRM00)             21/04/13 12:17 IBMUSER */
/* MBRLST032= ADCD.Z17S.PARMLIB(IEAAPP00)             21/04/13 12:17 IBMUSER */
/**** END MEMBERS LIST ***************************************    32 *********/
/*****************************************************************************/
  LOADS7 00000 <>Date: Tuesday, 17 Sep 2021, 16:46:32 - Dataset - SYS1.IPLPARM
  LOADS7 00001 IODF     99 SYS1
  LOADS7 00002 SYSCAT   S7SYS1113CCATALOG.Z112S.MASTER
  LOADS7 00003 SYSPARM  CS
  LOADS7 00004 IEASYM   00
  LOADS7 00005 PARMLIB  USER.PARMLIB                                USR001
  LOADS7 00006 PARMLIB  ADCD.Z112S.PARMLIB                          S7RES1
  LOADS7 00007 PARMLIB  SYS1.PARMLIB                                S7RES1
  LOADS7 00008 NUCLEUS  1
```

Full Copy Blueprints

Using the optional FULCOPY Keyword allows ISNMBRS to automatically build FULL Copies of members in the Dataset List or selected Members. FULCOPY Blueprint Operations can be defined to execute in the Foreground or Background or in both. FULCOPY Compare Operations support the use of the USEBSDN.

Sample Full Copy Blueprints

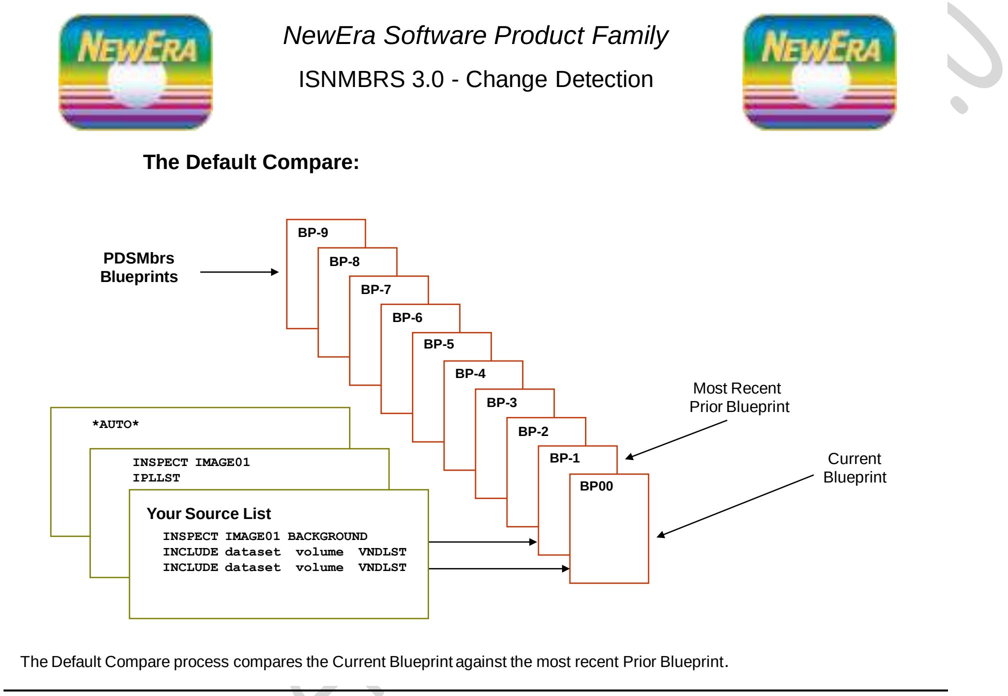


```
/* CPYLST247= IFO.IFO6M.INSTLIB(ZSIMTBO)              USR003  ZSIMTBX GHBLST  */
/* CPYLST248= IFO.IFO6M.INSTLIB(ZSIMTBO)              USR003  ZSIMTBO GHBLST  */
/* CPYLST249= IFO.IFO6M.INSTLIB(ZSIMTCP)              USR003  ZSIMTCP GHBLST  */
/* CPYLST250= IFO.IFO6M.INSTLIB(ZSIMTC2)              USR003  ZSIMTC2 GHBLST  */
/* CPYLST251= IFO.IFO6M.INSTLIB(ZSR@PRIM)             USR003 ZSR@PRIM GHBLST  */
/**** END DATASET(MEMBER) LIST ********************************   251 *********/
-Member- -Seq- --------------- Translated Member Content --------------- Transla
  $INDEX 00000 <>Date: Tuesday, 17 Sep 2021 at 16:46:32 - DSN - IFO.IFO6M.INIT
  $INDEX 00001 $INDEX   - This member
  $INDEX 00002 $NOTES   - Special installation notes
  $INDEX 00003 ALLOC    - Job to allocate the Image Focus Datasets
  $INDEX 00004 BUILD    - JOB to build the Image Focus product
  $INDEX 00005 IFOBAT   - Sample  proc for running an operating system (OPSYS)
  $INDEX 00006            inspection as a batch job
  $INDEX 00007 IFOBATA  - Sample  proc for running an operating system (OPSYS)
  $INDEX 00008            inspection with subsystem inspections as a batch job
  $INDEX 00009 IFOBATS  - Sample  proc for running an operating system (OPSYS)
  $INDEX 00010            inspection for use with the Sentry feature
  $INDEX 00011 IFOBG    - Sample proc for the background monitor function.  This
  $INDEX 00012            function will do inspections automatically on an inter
  $INDEX 00013            basis.
  $INDEX 00014 IFOM     - Sample proc for the master task for a multiple user Im
  $INDEX 00015            Focus
  $INDEX 00016 IFODSCK  - A dummy proc for use with the Dataset Check inspector
  $INDEX 00017 IFOLOAD  - A dummy proc for use with the Load Module inspector
  $INDEX 00018 IFOMBRS  - A dummy proc for use with the Members inspector
```

7.7 Comparing ISNMBRS Blueprints

7.7.1 Compare Process Types

The Default Compare



NewEra Software Product Family

ISNMBRS 3.0 - Change Detection

The Default Compare:

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset  volume  VNDLST
INCLUDE dataset  volume  VNDLST
```

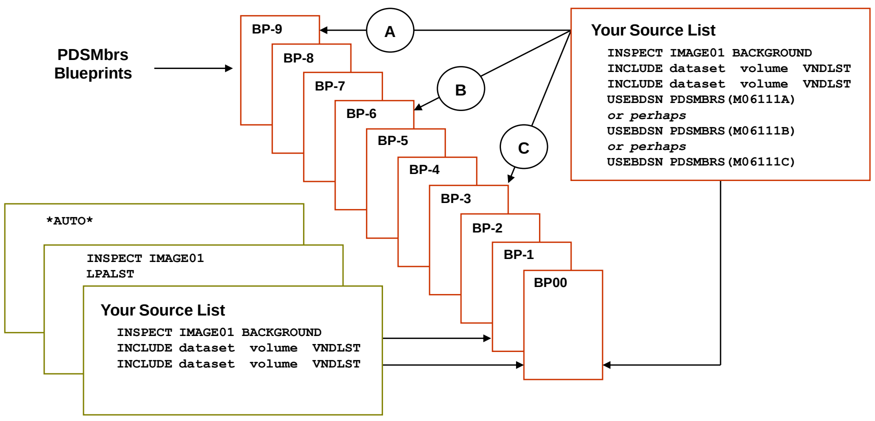
The Default Compare process compares the Current Blueprint against the most recent Prior Blueprint.

Point in Time Compare

NewEra Software Product Family

ISNMBRS 3.0 - Change Detection

The Default Compare vs. Point in Time Compare:



Your Source List

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset  volume  VNDLST
INCLUDE dataset  volume  VNDLST
USEBDSN PDSMBRS(M06111A)
or perhaps
USEBDSN PDSMBRS(M06111B)
or perhaps
USEBDSN PDSMBRS(M06111C)
```

```
*AUTO*
```

```
INSPECT IMAGE01
LPALST
```

Your Source List

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset  volume  VNDLST
INCLUDE dataset  volume  VNDLST
```

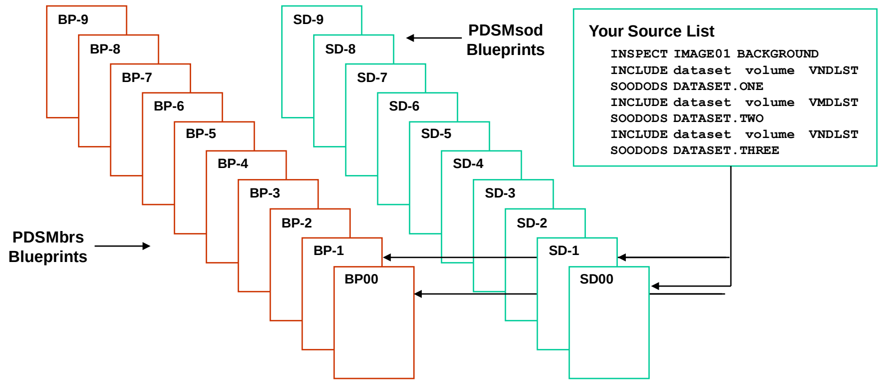
The USEBDSN Keyword may be used with PDSMBRS, PDSMSOD, PDSMPAK and PDSMFUL Blueprints.

Pseudo Compare

NewEra Software Product Family

ISNMBRS 3.0 - Change Detection

The Standard Compare vs. Pseudo Compare:



Your Source List

```
INSPECT IMAGE01 BACKGROUND
INCLUDE dataset  volume  VNDLST
SOODODS DATASET.ONE
INCLUDE dataset  volume  VMDLST
SOODODS DATASET.TWO
INCLUDE dataset  volume  VNDLST
SOODODS DATASET.THREE
```

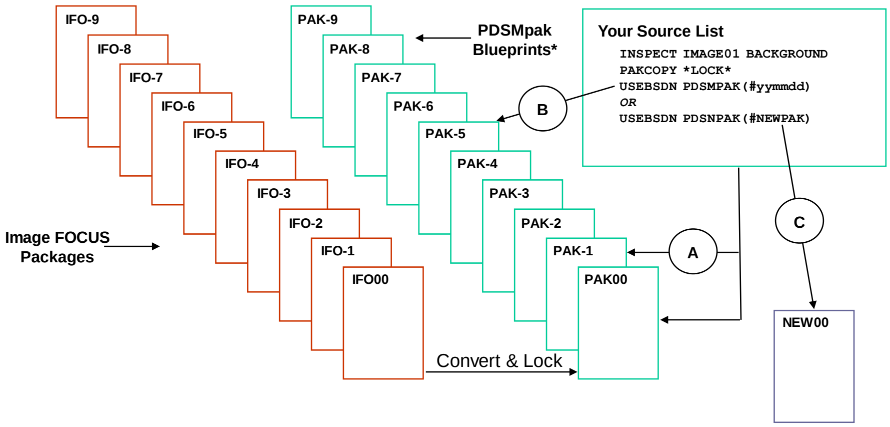
PDSMsod Blueprints & Change Reports are built using the SOODODS DS Name and *NOT* the real DS name.

Pack Copy Compare

NewEra Software Product Family

ISNMBRS 3.0 - PAKCOPY

ISNMBRS - PAKCOPY



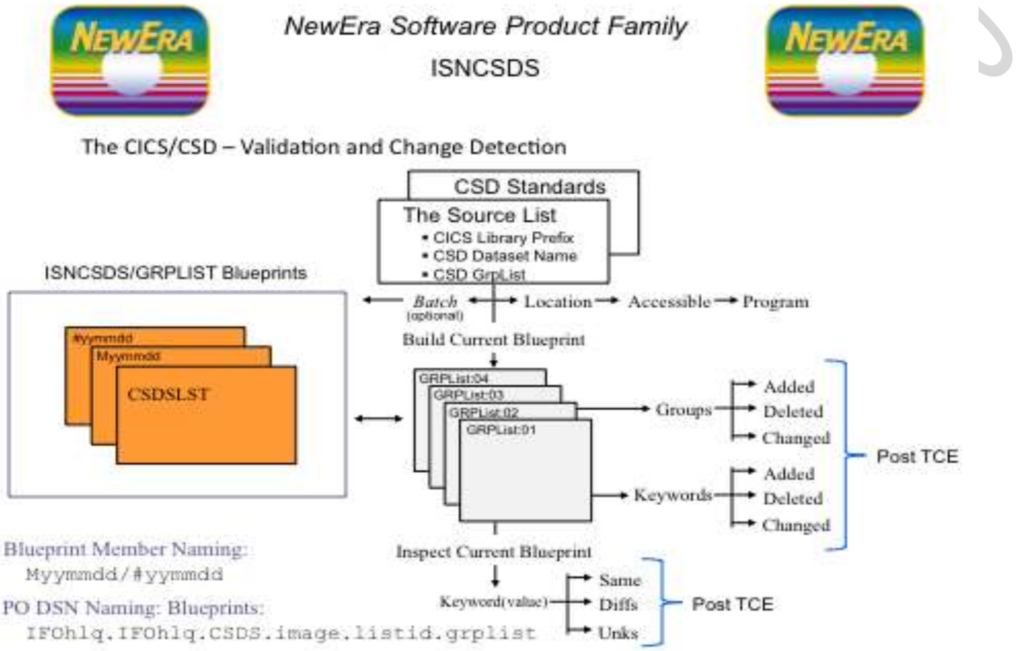
```
INSPECT IMAGE01 BACKGROUND
PAKCOPY *LOCK*
USEBSDN PDSMPAK(#yymmdd)
OR
USEBSDN PDSNPAK(#NEWPAK)
```

`IFOhlq.IFOhlq.PSDMpak.image.listid`

8 Working with ISNCSDS

8.1 Application Overview

The CICS System Configuration Dataset (CSD) contains the configuration parameters used in the initialization of one or more CICS Regions. The purpose of this Supplemental Inspector is to extract *All or Named GROUPs* from the CSD, evaluate their configuration parameters against a set of user defined standards, blueprint the GROUPs, detect changes, and report findings as directed.

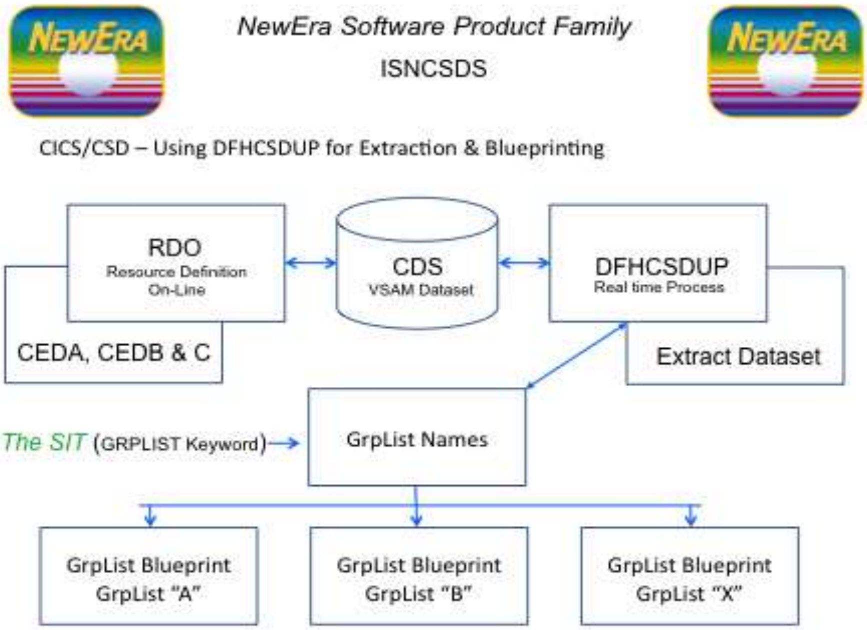


8.2 Controlled Datasets and Source List

A Controlled Dataset candidate is any dataset that is of valued importance to your understanding of the integrity and fitness of the z/OS system environment. Candidate datasets become Controlled Datasets when they are defined to the Integrity Controls Environment and/or its Applications using a Dataset Control/Source List. In this application, the Source List will point the inspector to the CICS System Definition (CSD) Dataset.

8.3 Extracting CSD Source

Using the information provided in the Source List, the inspector will attempt to indentify the target CSD. If the target is valid, it will immediately begin extracting its contents and build a Blueprint for each named CSD Group. The extractor used is the trusted IBM module DFHCSDUP.


NewEra Software Product Family
ISNCSDS
CICS/CSD – Using DFHCSDUP for Extraction & Blueprinting
RDO
Resource Definition On-Line
CEDA, CEDB & C
CDS
VSAM Dataset
DFHCSDUP
Real time Process
Extract Dataset
The SIT (GRPLIST Keyword)
GrpList Names
GrpList Blueprint GrpList "A"
GrpList Blueprint GrpList "B"
GrpList Blueprint GrpList "X"

8.4 Configuring the ISNCSDS application

ISNCSDS, like all the other Supplemental Inspectors, shares certain common shared configuration definitions. In addition to those common shared definitions, ISNCSDS responds to its own unique configuration definition set. In this release, no configuration modifications are required.

8.4.1 ISNCSDS Configuration Options

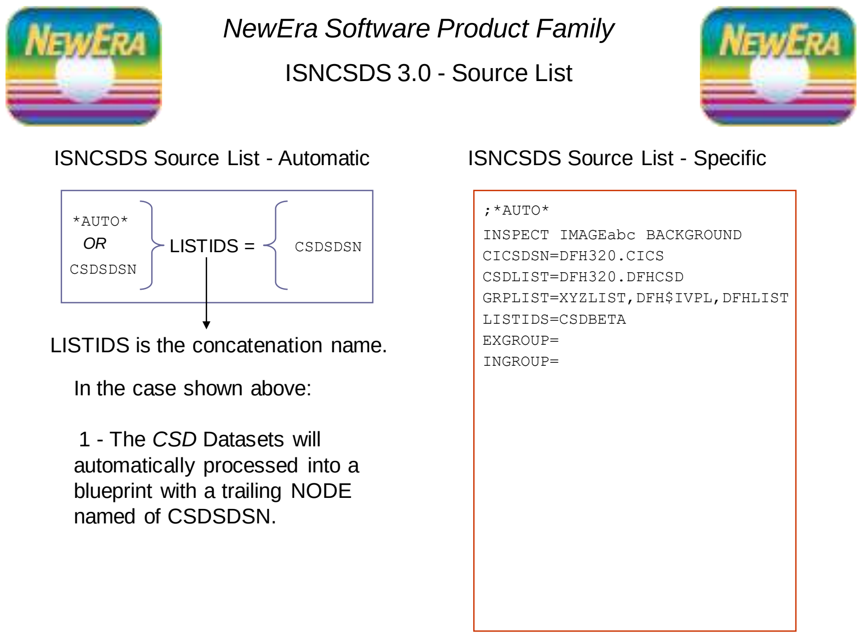
Keyword	Default/Optional	Functional Description
		No Specific Configuration Modifications are required for this Supplemental Inspector.

8.4.2 Creating an ISNCSDS Source List

When using ISNCSDS you have a choice – run “Out of the Box” and let Image FOCUS discover the CSD Dataset for each Image defined for inspection or customized ISNCSDS Source List as described below. The ISNCSDS Source List is member PLGCSDS and can be found (along with a sample SAMCSDS) in the Integrity Controls Environment USERLIB. PLGCSDS can be edited directly under TSO and/or Image FOCUS and/or via the optional Supplemental In-Line Interface.

8.4.3 ISNCSDS Source List Overview

The Source List tells the Inspector what to Inspect and Blueprint. Its content could be as simple as a single dataset or list of datasets that in some way are related. In a more complex form, the Source List contains special blueprinting instructions that specify the level of detail to be included in a Blueprint and/or what exactly is to be compared when detecting changes.



NewEra Software Product Family

ISNCSDS 3.0 - Source List

ISNCSDS Source List - Automatic

LISTIDS is the concatenation name.

In the case shown above:

1 - The *CSD* Datasets will automatically processed into a blueprint with a trailing NODE named of CSDSDSN.

ISNCSDS Source List - Specific

```
;*AUTO*
INSPECT IMAGEabc BACKGROUND
CICSDSN=DFH320.CICS
CSDLIST=DFH320.DFHCSD
GRPLIST=XYZLIST,DFH$IVPL,DFHLIST
LISTIDS=CSDBETA
EXGROUP=
INGROUP=
```

*Source List are found in IFOhlq1.IFOhlq1.USERLIB. SAMPCSDS is a sample, PLGCSDS is the real thing.

8.4.4 ISNCSDS Source List Options

The following table details the ISNMBRS Source List Keywords and their possible values.

Keyword	Default/Optional	Functional Description
AUTO		Planned for a future release
INSPECT	Image_name	Allows for multiple Image definitions in a single Source List
CICSDSN	Library Name	Names the CICS Dataset
CSDLIST	CSD Source Library	The full qualified name of the Source CICS System Definition Library.
GRPLIST	GROUP LIST Names	An optional list of comma separated GROUP List Names that are to be Included in inspection and blueprinting.
EXGROUP	Exclude GROUPs	An optional list of comma separated GROUP Names to be Excluded from inspection and blueprinting.
INGROUP	Include GROUPs	An optional list of comma separated GROUP Names to be Included for inspection and blueprinting.

8.5 Source List Examples

8.5.1 Default ISNCSDS Source List

```
****** ***************************** Top of Data ******************************
000001 -INC/EXC- -----------------DATASET NAME----------------- -VOLUME-LISTID-
```

8.5.2 Sample ISNCSDS Source List

```
****** ***************************** Top of Data ******************************
000001 -INC/EXC- -----------------DATASET NAME----------------- -VOLUME-LISTID-
000002 ;*AUTO* (RESERVED FOR A FUTURE RELEASE)
000003
000004 INSPECT IMAG0001 FOREGROUND
000005 CICSDSN=DFH320.CICS,CSDLIST=DFH320.DFHCSD,LISTIDS=TESTING
000006 GRPLIST=XYZLIST,DFH$IVPL,DFHLIST
000007
000008 ;CICSDSN=THE NODE PREFIX(S) TO .SDFHLOAD WHERE EXTRACTOR 'DFHCSDUP'
000009 ;CSDLIST=THE NAME OF THE TARGET CSD DATASET
000010 ;GRPLIST=THE NAME OF THE GROUP LIST TO BE PROCESSED
000011 ;LISTIDS=OPTIONAL BLUEPRINT DATASET NODE, DEFAULT IS INSPECTOR NAME
000012 ;EXGROUP=NAME,NAME,NAME (EXCLUDE THESE GROUP IN PROCESS)
000013 ;INGROUP=NAME,NAME,NAME (INCLUDE THESE GROUP IN PROCESS)
****** **************************** Bottom of Data ****************************
```

8.6 ISNCSDS Inspection

During inspection processing, ISNCSDS will check the value of the CSD Resource Keywords against user defined Validation Standards.

8.7 CSD Validation Standards List

The ISNCSDS Validation Standards List is member CSDSSTD and can be found in the Integrity Controls Environment USERLIB. CSDSSTD can be edited directly under TSO and/or Image FOCUS and/or via the optional Supplemental In-Line Interface.

8.7.1 Validation Standards List Example

```
---Resource Name(range)---   =key,keyword(value)
--------------------------   =------------------
RESOURCE=CONnection(group)   =nop,keyword(value)
RESOURCE=CORbaserver(group)  =nop,keyword(value)
RESOURCE=DB2Conn(group)      =nop,keyword(value)
RESOURCE=DB2Entry(group)     =nop,keyword(value)
RESOURCE=DB2Tran(group)      =nop,keyword(value)
RESOURCE=DJar(group)         =nop,keyword(value)
RESOURCE=DOctemplate(group)  =nop,keyword(value)
RESOURCE=Enqmodel(group)     =nop,keyword(value)
RESOURCE=File(group)         =nop,keyword(value)
RESOURCE=Ipconn(group)       =nop,keyword(value)
RESOURCE=Journalmodel(group) =nop,keyword(value)
RESOURCE=LIbrary(group)      =nop,keyword(value)
RESOURCE=LSRpool(group)      =nop,keyword(value)
RESOURCE=Mapset(group)       =YES,STATUS(ENABLED)
RESOURCE=PARTItionset(group) =nop,keyword(value)
RESOURCE=PARTNer(group)      =nop,keyword(value)
RESOURCE=PIpeline(group)     =nop,keyword(value)
RESOURCE=PROCesstype(group)  =nop,keyword(value)
RESOURCE=PROFile(group)      =nop,keyword(value)
RESOURCE=PROGram(group)      =YES,CEDF(NO)
RESOURCE=Requestmodel(group) =nop,keyword(value)
RESOURCE=Sessions(group)     =nop,keyword(value)
RESOURCE=TCpipservice(group) =nop,keyword(value)
RESOURCE=TDqueue(group)      =nop,keyword(value)
RESOURCE=TErminal(group)     =nop,keyword(value)
RESOURCE=TRANClass(group)    =nop,keyword(value)
RESOURCE=TRANSaction(group)  =YES,RESSEC(YES)
RESOURCE=TSmodel(group)      =nop,keyword(value)
RESOURCE=TYpeterm(group)     =nop,keyword(value)
RESOURCE=Urimap(group)       =nop,keyword(value)
RESOURCE=Webservice(group)   =nop,keyword(value)
```

8.8 Setting a Validation Standard

The Validation Standards file is read with each execution of ISNCSDS and each individual Resource Standard is evaluated. (You may repeat a Resource Definition as many times as needed.) If the first value following the “=” is “YES” the standard is considered active and will be applied within the scope of the value that appears with “()” immediately following the Resource Name. If the value is “group” the standard will be applied against *ALL GROUPS* that utilize the named Resource. If the value is other than “group” it will be considered to be a Group Name and therefore will only be applied against matching Group Names. The actual standard to be applied is derived from a keyword and its matching (value). All or Named Groups using active Resources that have matching values for keyword are logged and reported as “passed” while those that do not use matching keywords are logged and reported as “failed”. The total use of a Keyword within Group across all Groups is reported by setting the keyword value as “()”.

8.8.1 Sample CSD Inspection Report

```
CSD1140I BEGINNING INSPECTION OF CURRENT CICS CSD/GRPLST BLUEPRINT.
|
CSD1120I <>ACTIVE CSD STANDARDS VALIDATION - SUMMARY:
|
CSD1120I   -Groups- Resource Type ----Keyword---- --Values-- Hits Pass Fail
CSD1120I   -------- ------------- --------------- ---------- ---- ---- ----
CSD1121A   ALLGROUP        MAPSET          STATUS     ENABLE   29    0   29
CSD1121I   ALLGROUP       PROGRAM            CEDF    --n/s-- 1513    0    0
CSD1121A   ALLGROUP   TRANSACTION          RESSEC        YES  138   22  116
CSD1120I   ======== ============= =============== ========== ==== ==== ====
|
CSD1120I <>ACTIVE CSD STANDARDS VALIDATION - FAILURE DETAIL:
|
CSD1120I   ALLGROUP->MAPSET->STATUS->ENABLE - FAILED:29
|
CSD1120I   -Groups- Own --Name-- -----Attribute----- ----Keyword(Value)----
CSD1120I   -------- --- -------- ------------------- ----------------------
CSD1120I   DFHDBCTL IBM  DFHDBDE         START_GROUP STATUS(ENABLED)
CSD1120I   DFHDBCTL IBM  DFHDBIE         START_GROUP STATUS(ENABLED)
…
CSD1120I   ALLGROUP->TRANSACTION->RESSEC->YES - FAILED:116
|
CSD1120I   -Groups- Own --Name-- -----Attribute----- ----Keyword(Value)----
CSD1120I   -------- --- -------- ------------------- ----------------------
CSD1120I     DFHBMS IBM     CSPG            SECURITY RESSEC(NO)
CSD1120I     DFHBMS IBM     CSPQ            SECURITY RESSEC(NO)
CSD1120I     DFHBMS IBM     CSPS            SECURITY RESSEC(NO)
CSD1120I    DFHCONS IBM     CWTO            SECURITY RESSEC(NO)
```

8.9 ISNCSDS Blueprinting

ISNCSDS blueprinting is a process of identifying certain specific attributes of CSD Groups and their Resource, making an organized record of those attributes and subsequently using those records as the basis for detecting changes in Resource Keywords and their Paired Value.

8.9.1 Blueprints Types

In order to accommodate the need to detect changes at different levels and under different circumstances, ISNCSDS can create different types of Blueprints. Each serves a specific purpose and some may be optionally used to fulfill a specific need.

The Default Blueprint Format

The Default ISNCSDS Blueprint is created using information derived from an extraction of each individual GROUP List named in a CSD. A sample default format is shown below. Note that the Blueprint is broken into two major sections. The first section is used to record the GROUP Names found within the List. The second section is used to record the characteristics of each individual Group, i.e. Group Name, owner, resource type, resource name, resource attribute, date and time of last update, keyword and keyword value.

8.9.2 Sample Default CSDS Blueprint

```
/******************************************************************************/
/* OPERATION= INSPECTION EXECUTION IS IN THE FOREGROUND.                      */
/* BLUEPRINT= IFO.IFOP.CSDSLST.TESTING.DFHLIST(M090505)                       */
/* CSDSOURCE= DFH320.DFHCSD                                                   */
/* GROUPLIST= DFHLIST                                                         */
/* DATEBUILD= Tuesday, 17 Sep 2021 at 16:46:32                                */
/*** USR Group Names **********************************************************/
/* GROUPNAME=  DFHDCTG  DFHCBTS DFHIPECI      CEE                             */
/*** IBM Group Names **********************************************************/
/* GROUPNAME=   DFHBMS  DFHCONS DFHDBCTL   DFHDB2   DFHEDF   DFHEDP    DFHFE  */
/* GROUPNAME= DFHHARDC DFHINQUI DFHINTER   DFHISC  DFHMISC DFHMSWIT DFHOPCLS  */
/* GROUPNAME=  DFHOPER DFHPGAIP   DFHRMI DFHRSEND  DFHSIGN   DFHSPI DFHSTAND  */
/* GROUPNAME=  DFHVTAM DFHVTAMP  DFHTYPE  DFHTERM  DFHFEPI   DFHTCL  DFHINDT  */
/* GROUPNAME=  DFHLGQC  DFHSDAP  DFHCLNT DFHLGMOD   DFHWEB    DFHMQ  DFHPIPE  */
/* GROUPNAME=  DFHIIOP DFHISCIP  DFHJAVA   DFHOTS   DFHRQS   DFHCFC   DFHDOC  */
/* GROUPNAME=    DFHBR DFHPSSGN  DFHADST  DFHEJBU    DFHDP  DFHDPWB    DFHSO  */
/******************************************************************************/
DFHDCTG,USR,TDQUEUE,CADL,START_GROUP,08/296,09/22,DESCRIPTION(CEDA_VTAM_RESOURCE
DFHDCTG,USR,TDQUEUE,CADL,START_GROUP,08/296,09/22,TYPE(INDIRECT)
DFHDCTG,USR,TDQUEUE,CADL,EXTRA-PARTITION-PARAMETERS,08/296,09/22,DATABUFFERS()
DFHDCTG,USR,TDQUEUE,CADL,EXTRA-PARTITION-PARAMETERS,08/296,09/22,DDNAME()
DFHDCTG,USR,TDQUEUE,CADL,EXTRA-PARTITION-PARAMETERS,08/296,09/22,DSNAME()
DFHDCTG,USR,TDQUEUE,CADL,EXTRA-PARTITION-PARAMETERS,08/296,09/22,SYSOUTCLASS()
```

Other Blueprint Formats

Other Blueprints will be added as the need for them materializes.

8.9.3 Sample of CSD Detected Change Report

```
CSD1160I BEGINNING THE CICS CSD/GRPLST BLUEPRINT COMPARISON - XYZLIST.
|
CSD2110I <>CURRENT CSD/GRPLIST BLUEPRINT FOR COMPARISON OF "XYZLIST" BUILT.
CSD2111I   DSN(MOD):IFO.IFOP.CSDSLST.TESTING.XYZLIST.
CSD2112I    >CURRENT CSD/GRPLST BLUEPRINT IS DATED:14 AUG 2021 AT 19:44:49.
|
CSD1200I <>NO CHANGES FOUND IN CSD GROUP COMPOSITION.
|
CSD1200I   NO CSD GROUPS ADDED TO THIS GRPLIST.
|
CSD1200I   NO CSD GROUPS DELETED FROM THIS GRPLIST.
|
CSD1200I   54 COMMON CSD GROUPS IN THIS GRPLIST.
|
CSD1200C <> 2 COMMON CSD GROUP STRUCTURES HAVE CHANGED.
|
CSD1200I     --Name-- --Name-- --Name-- --Name-- --Name-- --Name-- --Name--
CSD1200I      DFHDCTG DFHTERMC
CSD1200I     ======== ======== ======== ======== ======== ======== ========
|
CSD1200I   STRUCTURAL CHANGES WITHIN GROUP - DFHDCTG - MAINTAINED BY - USR.
|
CSD1200I    > 2 GROUP KEYWORD OR STRUCTURAL ADDITIONS.
|
CSD1200I     Cng Resource Type --Name-- UpDate UpTime ---Keyword(Values)---
CSD1200I     --- ------------- -------- ------ ------ ---------------------
CSD1200I     Add       TDQUEUE     CADL 08.296  09:22 OPENTIME()
CSD1200I     Add       TDQUEUE     CADL 08.296  09:22 REWIND()
CSD1200I     === ============= ======== ====== ====== ======================
|
CSD1200I    >NO KEYWORD OR STRUCTURAL DELETIONS.
|
CSD1200I   STRUCTURAL CHANGES WITHIN GROUP - DFHTERMC - MAINTAINED BY - IBM.
|
CSD1200I    >NO KEYWORD OR STRUCTURAL ADDITIONS.
|
CSD1200I    >NO KEYWORD OR STRUCTURAL DELETIONS.
|
CSD1200I    > 1 GROUP KEYWORD OR STRUCTURAL CHANGES.
|
CSD1200I     Cng Resource Type --Name-- UpDate UpTime ---Keyword(Values)---
CSD1200I     --- ------------- -------- ------ ------ ---------------------
CSD1200I     New      TERMINAL     AUTC 08.296  09:22 TERMPRIORITY(4)
CSD1200I     Old      TERMINAL     AUTC 08.296  09:22 TERMPRIORITY(0)
CSD1200I     === ============= ======== ====== ====== ======================
|
CSD1200I <>CHANGE LOG FOR THIS GRPLIST UPDATED - XYZLIST.
CSD1200I   GRPLIST LOG DATASET(MBR):IFO.IFOP.CSDSLST.TESTING.XYZLIST($CNGLOG)
|
CSD1200I <>OPTIONAL LOG ENTRY POSTING TO CONTROL JOURNAL IS SET "OFF".
```

9 Appendix A

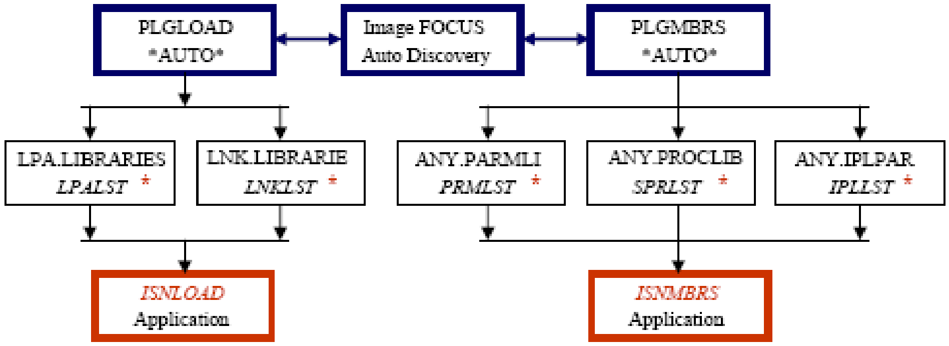
9.1 Control Dataset Automation

The Control Datasets to be evaluated by ISNLOAD and ISNMBRS may be automatically determined by using the *AUTO* Keyword within their Source List.

• When used with ISNLOAD, the resulting Source List will include all LPALST and LNKLST Libraries that are in use by the Image under evaluation by Image FOCUS.

• When used with ISNMBRS, the resulting Source List will include all ParmLib, ProcLib and IPLParm Datasets that are in use by the Image under evaluation by Image FOCUS.

The *AUTO* Keyword



**Source List Identification Name - LISTID*

9.2 Common Configuration Elements

9.2.1 NSEPLG00 – The Configuration File

The “Default” Supplemental Inspector Configuration File is automatically built when a Supplemental Inspector is first accessed. The configuration member named is NSEPLG00 and can be found in the following Image SENTRY dataset:

```
    IFOhlq1.IFOhlq2.$ISENTRY.$CONFIGS
```

Where the NODEs IFOhlq1 and IFOhlq2 are dataset qualifiers assigned to Image FOCUS during the Image FOCUS installation.

9.2.2 Common Configuration Keywords and Values

The Configuration Files contain a number of Configuration Keywords. Each keyword uses the format KEYWORD= VALUE. The pairing of these Keywords and their values determines and controls the operation of an Inspector. Each of the Keywords common to all Supplemental Inspectors, its function and possible values is explained below.

Keyword	Default/Optional	Functional Description
_URFILE=	DO NOT CHANGE	NODE qualifiers assigned to Image FOCUS during its installation. See also PLGBASE configuration option below.
_PREFIX=	DO NOT CHANGE	NODE qualifiers assigned to Image FOCUS during its installation. See also PLGBASE configuration option below.
_TSOUSER=	DO NOT CHANGE	NODE qualifiers assigned to Image FOCUS during its installation. See also PLGBASE configuration option below.
_USERHLQ	DO NOT CHANGE	NODE qualifiers assigned to Image FOCUS during its installation. See also PLGBASE configuration option below.
RPTMENU=	XXX AMAZING	Each “X” represents an inspector as described below. The default value for each is “A”. Setting the value to “N” will turn off the In-

		Line Interface for that Inspector. The second value is the password that is needed to reach the Source List and Configuration file VIA the In-Line Interface. The default is AMAZING.
SETUPDS=	XXX	Each "X" represents an inspector as described below. The default value for each is "A". Setting the default value to "N" will automatically prevent configuration file updates via the In-Line Interface.
PANSETS=	XXX	Reserved as a future enhancement
SRCLIST=	XXX	Each "X" represents an inspector as described below. The default value for each is "A". Setting the value to "N" will cause the inspector to disregard the Source List passed to it by Image FOCUS. It will look instead for the list as the value of MODLIST, MBRLIST, CSDLIST.
IFOSBKG=	XXXX	Each "X" represents an inspector as described below. The default value for each is "A". Setting the value to "N" will cause the inspector to function only in the background.
RETAINS=	10	Sets the upper limit of the number of Blueprints, by type, that will be retained for each inspector.
FILTERS=	AUTO/NONE	Controls overall message filtering. "NONE" turns message filtering OFF.
MFILTER=	AUTO/NONE	Reserved as a future enhancement
MLEVELS=	AUTO/NONE	Controls the changing of message levels when using the MLEVSxx configuration keywords as described below.
NOTICES=	AUTO/NONE	Controls the overall event notification process. "NONE" turns notification OFF.
BKGONLY=	AUTO/NONE	If set to "AUTO" notices of any type are sent only from the background. "NONE" allows notification in Foreground and Background.

VIAMAIL=	AUTO/NONE	If set to "AUTO" turns "ON" the Email notification category as described below.
MSERVER=	EMAIL SERVER NAME	The internet address/domain name of the Email Server.
SVRPORT=	EMAIL SERVER PORT	Port address that the mail server uses.
ESENDER=	EMAIL OF SENDER	The universal sender email address.
SUBJECT=	EMAIL SUBJECT	The universal email subject.
COPYEML=	EMAIL COPY RECIPIENT	The universal email copy address.

9.2.3 Sending Email

When sending email you have two options:

First, send one email to the universal recipient using the following option.

```
TSOULST= TSOUSERID/NONE
```

Second, send to a group of individuals using the following option.

```
EMLLS01-10 Reserved for a future release
```

9.2.4 Controlling message levels

To change a message, first identify the full inspection message you wish to change. An example inspection message is shown below:

```
MLEVS01= MOD0001A
```

Next, decide how you want to change the message level. Possible levels are: Error (E), Warning (W), Notice (N), Change (C), Audit (A) and Informational (I).

In this case, the Audit message will be changed to Informational by the following entry:

```
MLEVS01= MOD00001A(I)
```

Note that several changes may be made on the same line and that each change must be separated by at least one blank.

9.2.5 Inspector Batch Job Card Specifications & Rules

Use these keywords to specify the job name (replace “xxx” with alphanumeric characters):

```
LOADJOB= LOADxxx JOB
MBRSJOB= MBRSxxx JOB
CSDSJOB= CSDSxxx JOB
```

Use this keyword to specify the way batch jobs are submitted:

```
BATRULE= AUTO
```

AUTO dynamically creates and stores the JCL in USERLIB. If the JCL already exists, it will be overwritten. Once created, the JCL is automatically submitted.

NONE creates and stores the JCL, but does not automatically submit (controlled by user and/or job scheduler). This allows the user to modify the JCL before submitting. If the JCL already exists, it will not be overwritten. JCL is stored in the following dataset:

```
IFOhlq1.IFOhlq2.$ISENTRY.$BATLOAD.imagename
```

IFOhlq1 and IFOhlq2 are the High Level Qualifiers assigned to Image FOCUS during its installation.

10 Appendix B

10.1 Installation

10.1.1 Downloading

All Integrity Controls Environment Applications are distributed via the worldwide web and downloaded from www.newera.com directly to your desktop. The download contains application components that have been zipped into a single distribution file. The application components are:

```
1. ISNLOAD - The Load Module Inspector
2. ISNMBRS - The Member Inspector
3. ISNCSDS - The CICS CSD Inspector
4. Documentation
```

Unzip and Upload

Once the download is on your desktop, you will need to unzip it and upload the application MODULES to a pre-allocated dataset on your z/OS Host or upload them directly into the Integrity Controls Environment (ICE) "USERLIB" using the member names noted above. To maintain product integrity, certain application MODULES are distributed in a proprietary encoded format; others are not. DO NOT attempt to alter the encoded MODULES in any way as doing so will result in unpredictable product failures.

The product documentation is distributed as a PDF file. To view the documentation, you will need the ADOBE Acrobat Reader. If you do not currently have the Reader, a copy can be downloaded from www.adobe.com.

The License Key

Whether you are an existing Integrity Controls Environment user, or using the product on a trial basis, you will need to add the License Key(s) to your Integrity Controls Environment ParmLib dataset NSEPRM00. Once the License Key is inserted, ALL Integrity Controls Environment Application functions will be unlocked the next time you logon to Integrity Controls Environment.

If you downloaded Integrity Controls Environment using the 'Pre-Authorize' link, you are not required to insert License Key control cards into NSEPRM00. NewEra does it for you during the download!

10.1.2 Installation and Setup

Once you have downloaded and moved the application components described above into the ICE "USERLIB", the Supplemental Inspectors are ready for use; no additional installation is required.

10.1.3 Product Updates

As product updates and new releases become available, you will be automatically notified. When you want to update your product installation, reuse your original download link or request a new one from NewEra Technical Support.

New Application MODULES can be moved directly into the "ICE USERLIB"; no additional installation is needed.

10.1.4 Checklist

Use the following checklist to ensure a successful installation of the Supplemental Inspectors.

Action	Status
Download the .NEZ file that was shipped to you via an email link from NewEra Software Inc. Rename the file from an .NEZ to an .EXE as this is a self extracting file.	
Unzip the downloaded file.	
Confirm that all components (ISNLOAD, ISNMBRS, & ISNCSDS) are present.	
Upload components (ISNLOAD, ISNMBRS, & ISNCSDS) into the Image FOCUS USERLIB or pre-allocated datasets.	
Copy components into Image FOCUS USERLIB (if not originally uploaded there).	
Add License Key(s) to your Image FOCUS ParmLib dataset NSEPRM00.	
Set up configuration members PLGLOAD, PLGMBRS, and PLGCSDS with site specific variables.	
Logon to Integrity Controls Environment.	

11 Index

Contact us for additional information:

NewEra Software Technical Support

800-421-5035 or 408-520-7100
Or text support requests to 669-888-5061

support@newera.com

www.newera.com