

z/OS  
3.2

*z/OS Workload Interaction Correlator*



**Note**

Before using this information and the product it supports, read the information in [“Notices” on page 175](#).

This edition applies to IBM® z/OS® 3.2 (5655-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2025-09-30

© **Copyright International Business Machines Corporation 2020, 2025.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

**Figures..... V**

**Tables..... vii**

**About the IBM z/OS Workload Interaction Correlator ..... ix**

**How to provide feedback to IBM.....xi**

**Chapter 1. What is IBM z/OS Workload Interaction Correlator? ..... 1**

**Chapter 2. Enabling Workload Interaction Correlator.....7**

**Chapter 3. Writing programs that make use of Workload Interaction Correlator.....23**

**Chapter 4. Reference for system programmers..... 41**

**Chapter 5. Reference for application programmers..... 79**

**Appendix A. Accessibility.....173**

**Notices.....175**

**Index..... 179**



---

# Figures

1. Sample assembler program to call a WIC exit routine service..... 26



---

# Tables

1. Base elements and optional features in z/OS.....	3
2. Summary of SMF type 98 subtypes.....	13
3. Summary of SMF type 100 subtypes.....	22
4. Overview of specific (A), general (B), and overall (C) aggregate bucket instances.....	33
5. Aggregate bucket (supplied by exploiter; exploiter area mapped by WicWR_AggBucket).....	34
6. SMF type 98 subtype x aggregate bucket record (mapped by SMF98AggBucket in IHAHR098).....	35
7. Exception bucket (supplied by exploiter; exploiter area mapped by WicWR_ExcBucket).....	37
8. Example of SMF type 98 subtype x job index data record (mapped by SMF98JobIdx in IHAHR098)....	39
9. Example of SMF type 98 subtype x job list data record.....	39
10. Return and reason codes for the IFAWIC service.....	85
11. Structure CVTFIX.....	97
12. Structure CVTVSTGX.....	122
13. Structure CVTXTNT1.....	123
14. Structure CVTXTNT2.....	123
15. Cross Reference for CVT.....	125
16. Structure WICPARM.....	144
17. Structure WIC_WARNING_RSN.....	152
18. Structure WICENFQUAL.....	153
19. Structure WICENF.....	153
20. Structure WICNB_PARMLIST.....	154
21. Structure WICCB_PARMLIST.....	155
22. Structure WICWR_PARMLIST.....	156
23. Structure WICWR_AGGBUCKETPARMLIST.....	160

24. Structure WICWR_AGGBUCKET.....	160
25. Structure WICWR_EXCbucket.....	161
26. Cross Reference for IFAWICCB.....	165



# About the IBM z/OS Workload Interaction Correlator

---

**Purpose of this information** This is a collection of the information needed to understand and use IBM z/OS Workload Interaction Correlator. Some of the information contained in this collection also exists elsewhere in the z/OS library.

**Who should read this information** This information is intended for system programmers who are responsible for installing, configuring, and maintaining the IBM z/OS Workload Interaction Correlator. There is also information for people writing programs or subsystems that monitor performance.

## Related information

To find the complete z/OS library, go to [IBM Documentation \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).



## How to provide feedback to IBM

---

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. For more information, see [How to send feedback to IBM](#).



---

# Chapter 1. What is IBM z/OS Workload Interaction Correlator?

IBM z/OS Workload Interaction Correlator is a priced feature that provides infrastructure for z/OS software to generate synchronized, standardized, concise, content-rich data with common context for automated analysis by an analytics engine like the IBM z/OS Workload Interaction Navigator. IBM z/OS Workload Interaction Correlator requires IBM z14™ or z15™ hardware. Support will be included in the PTF for APAR OA57165 for z/OS in V2R3 and V2R4.

This collection includes updates to the following publications:

- *z/OS Planning for Installation*
- *z/OS MVS Initialization and Tuning Reference*
- *z/OS MVS Programming: Authorized Assembler Services Guide*
- *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*
- *z/OS MVS System Commands*
- *z/OS MVS System Management Facilities (SMF)*
- *z/OS MVS System Messages, Vol 6 (GOS-IEA)*
- *z/OS MVS System Messages, Vol 8 (IEF-IGD)*
- *z/OS MVS Data Areas Volume 1 (ABE - IAR)*
- *z/OS MVS Data Areas Volume 2 (IAX - ISG)*

You might also wish to view the following introduction and demonstration videos that are available in IBM MediaCenter:

- [IBM z/OS Workload Interaction Correlator and IBM z/OS Workload Interaction Navigator Introduction \(mediacenter.ibm.com/media/IBM%20z\\_OS%20Workload%20Interaction%20Correlator%20and%20Navigator%20Introduction/1\\_ksx31z0z\)](https://mediacenter.ibm.com/media/IBM%20z_OS%20Workload%20Interaction%20Correlator%20and%20Navigator%20Introduction/1_ksx31z0z)
- [IBM z/OS Workload Interaction Navigator Single Interval Demo \(mediacenter.ibm.com/media/IBM%20z\\_OS%20Workload%20Interaction%20Navigator%20Single%20Interval%20Demo/1\\_4fgi4vxs\)](https://mediacenter.ibm.com/media/IBM%20z_OS%20Workload%20Interaction%20Navigator%20Single%20Interval%20Demo/1_4fgi4vxs)

---

## List of base elements and optional features

Table 1 on page 3 is a partial listing of the table of base elements and optional features from *z/OS Planning for Installation* and describes the optional features related to Workload Interaction Correlator. The following table headings are used:

### **Name**

The name of the element or feature.

### **Last time changed (and equivalent product if nonexclusive)**

The most recent release in which the element or feature changed. ("Change" means that one or more of the element or feature FMID [function modification identifiers] was changed, or that the element or feature was added to the system. A new function added through a program temporary fix (PTF) is not considered a change. Also, for nonexclusive elements and features, the equivalent level of the separate product is listed in parentheses.

The last release in which an element or feature was changed is considered its *function level*. Do not confuse the function level with the product level. All elements and features are at the product level, but they are at various function levels. For example, the product level of Metal C Runtime Library is z/OS 3.2, but its function level is z/OS V1R13 because z/OS V1R13 was the last release in which it changed.

**Type and description**

Indicates the following attributes:

- Whether it is a base element or an optional feature.
- Whether the base element or optional feature is *exclusive* (existing only within z/OS) or *nonexclusive* (also available as a separate product).
- For an optional feature, whether the feature is priced or unpriced.
- For an optional feature, whether the feature supports dynamic enablement. All priced features support dynamic enablement.

**Description**

A brief description of the element or feature.

Table 1. Base elements and optional features in z/OS			
Name	Last time changed (and equivalent product if nonexclusive)	Type	Description
⋮			

Table 1. Base elements and optional features in z/OS (continued)

Name	Last time changed (and equivalent product if nonexclusive)	Type	Description
<b>IBM z/OS Workload Interaction Correlator</b>	z/OS 3.2	Optional feature, exclusive, priced, can be dynamically enabled.	<p>IBM z/OS Workload Interaction Correlator enables a z/OS component, middleware, or application that runs in multiple instances distributed across multiple address spaces to generate a high-frequency, summarized, system-wide view. Products can use the IBM z/OS Workload Interaction Correlator services to generate an SMF type 98 record for a product subtype every 5 seconds. The record contains data about the product's activities in a standardized, synchronized, contextualized format. For more information, see <a href="#">z/OS Workload Interaction Correlator in z/OS MVS Programming: Authorized Assembler Services Guide</a>.</p> <p>IBM z/OS Workload Interaction Correlator generated data can be used as input to an analytics engine, such as the IBM z/OS Workload Interaction Navigator.</p> <p>To use IBM z/OS Workload Interaction Correlator, your installation must enable the following:</p> <ul style="list-style-type: none"> <li>• Correlator feature through the product enablement policy by using the IFAPRDxx parmlib member.</li> <li>• Collection of type 98 records by using the SMFPRMxx parmlib member or SETSMF command.</li> </ul> <p>For more information, see IBM z/OS Workload Interaction Correlator in <a href="#">z/OS MVS System Management Facilities (SMF)</a>.</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. The entitlement and ordering of RMF or z/OS Advanced Data Gatherer allows usage of IBM z/OS Workload Interaction Correlator. You do not need to order the IBM z/OS Workload Interaction Correlator feature to use it when either z/OS Advanced Data Gatherer or RMF is ordered. The IFAPRDxx enablement of either z/OS Advanced Data Gatherer or RMF covers usage of IBM z/OS Workload Interaction Correlator, when either z/OS Advanced Data Gatherer or RMF is ordered and enabled in IFAPRDxx.</li> <li>2. IBM Threat Detection for z/OS (IBM TDz) is a separately priced product, which entitles you to the z/OS Workload Interaction Correlator feature. When you are entitled to IBM Threat Detection for z/OS, you can update your IFAPRDxx parmlib member to indicate that the z/OS Workload Interaction Correlator feature is enabled. Follow the instructions in the IBM Threat Detection for z/OS Program Directory.</li> </ol>



Table 1. Base elements and optional features in z/OS (continued)			
Name	Last time changed (and equivalent product if nonexclusive)	Type	Description
⋮			
<b>z/OS Advanced Data Gatherer</b>	z/OS 3.2	Optional feature, exclusive, priced, can be dynamically enabled.	<p>The z/OS Advanced Data Gatherer feature allows:</p> <ul style="list-style-type: none"> <li>• System administrators to configure to write SMF 70-78 record data into the SMF buffer and to SMF data sets or to a log stream.</li> <li>• Application programs to access collected Monitor II, Monitor III, and SMF 70-78 record data by using Sysplex Data Server APIs and z/OS Data Gatherer APIs.</li> </ul> <p><b>Note:</b> The entitlement and ordering of RMF or IBM z/OS Advanced Data Gatherer allows usage of the IBM z/OS Workload Interaction Correlator. You do not need to order the IBM z/OS Workload Interaction Correlator feature to use it when either z/OS Advanced Data Gatherer or RMF is ordered. The IFAPRDxx enablement of either z/OS Advanced Data Gatherer or RMF covers usage of the IBM z/OS Workload Interaction Correlator, when either z/OS Advanced Data Gatherer or RMF is ordered and enabled in IFAPRDxx.</p>

For the complete list of z/OS base elements and optional features, see [List of base elements and optional features](#) in *z/OS Planning for Installation*.



---

## Chapter 2. Enabling Workload Interaction Correlator

The following topics describe how to enable IBM z/OS Workload Interaction Correlator.

- IBM z/OS Workload Interaction Correlator adds or updates the following parameters in the SMFPRMxx member of parmlib:
  - HFTSINTVL (updated)
  - WIC (new)

These are further described later in "HFTSINTVL — Specifying the high-frequency throughput statistics interval" and "WIC — Specifying the generation of type 98 records for IBM z/OS Workload Interaction Correlator."

For complete information about the SMFPRMxx parameters, see [Statements and parameters for SMFPRMxx in z/OS MVS Initialization and Tuning Reference](#) and [Customizing SMF in z/OS MVS System Management Facilities \(SMF\)](#).

- To enable IBM z/OS Workload Interaction Correlator, complete the tasks in [“IBM z/OS Workload Interaction Correlator”](#) on page 9.
- For information about the changes to the applicable SMF records, see [“Correlator SMF records”](#) on page 12.

### IBM z/OS Workload Interaction Correlator

You can configure SMF to collect IBM z/OS Workload Interaction Correlator data. IBM z/OS Workload Interaction Correlator enables a product, such as a z/OS component, middleware, or application, that executes in multiple instances distributed across multiple address spaces to generate high-frequency, summarized, system-wide and compartmental data. Products can use the IBM z/OS Workload Interaction Correlator services to generate an SMF type 98 record for a product's subtype every 5 seconds containing data about the product's activities in a standardized, synchronized, contextualized format.

Records from the workload interaction correlator produce a summary view of the component's activities across address spaces of similar priority and CPU usage and highlight data from exceptional address spaces. These records are generated on a synchronized, high-frequency 5-second interval across multiple products.

Analytics tools, such as IBM z/OS Workload Interaction Navigator, can analyze these records across z/OS components, middleware, and applications to detect and correlate anomalies. Presenting an organized progression of interacting anomalies reduces the time and expertise needed to diagnose a performance problem. When an action is taken to solve or mitigate the performance problem, the system programmer can verify that the fix is behaving as expected by comparing the results before and after the implementation of the action.

---

### HFTSINTVL — Specifying the high-frequency throughput statistics interval

The HFTSINTVL parameter specifies the time interval, in seconds, for writing SMF type 98 records, which record high-frequency throughput statistics (HFTS). The value specified is ignored when the WIC parameter is specified. IBM z/OS Workload Interaction Correlator uses a 5-second interval, starting at the top of the minute and for each 5-second multiple that follows, for SMF type 98 records.

The supported HFTSINTVL values are 5, 10, 15, 20, 30, and 60 seconds.

When you specify a HFTSINTVL value, SMF type 98 records are written every 5 seconds, starting at the top of the minute and for each 5-second multiple that follows, for one minute at 0, 15, 30, and 45 minutes past each hour. For all other minutes during each hour, SMF type 98 records are written starting at the top of the minute and for each HFTSINTVL multiple.

For instance, HFTSINTVL(20) writes 216 records every hour in the following manner:

- **Minutes 0, 15, 30, and 45:** Write 1 record at the top of the minute and for each 5-second multiple that follows, for a total of 48 records.
- **The other 56 minutes:** Write 1 record every 20 seconds at the top of the minute and for each 20-second multiple that follows, for a total of 168 records.

IBM recommends a HFTSINTVL value of 5 seconds.

The NOHFTSINTVL parameter disables the HFTS interval and prevents the collection of type SMF type 98 records. This is the default value.

SMF collects type 98 records only when both the HFTSINTVL parameter and TYPE(98) parameter are specified. Specifying NOHFTSINTVL along with TYPE(98) allows you to disable the collection of type 98 records without changing the TYPE setting.

IBM benchmarks did not detect additional measurable CPU overhead due to collecting HFTS data.

IBM recommends configuring Hardware Instrumentation Services (HIS) to collect hardware counters as described in [Collecting CPU MF \(Counters\) on z/OS \(www.ibm.com/support/pages/system/files/inline-files/Detailed\\_CPU\\_MF\\_Counters\\_Enablement\\_Instructions\\_September\\_2019.pdf\)](http://www.ibm.com/support/pages/system/files/inline-files/Detailed_CPU_MF_Counters_Enablement_Instructions_September_2019.pdf). Collecting hardware counters enriches the data collected in SMF type 98 subtype 1 records.

## WIC – Specifying the generation of type 98 records for IBM z/OS Workload Interaction Correlator

---

When you specify the WIC parameter, the IBM z/OS Workload Interaction Correlator (WIC) is enabled on the system. z/OS components, middleware, and applications can use the IFAWIC service to register for instrumentation, request an instrumentation buffer for the job, and provide an exit routine.

The exit routine will be called to summarize instrumentation data and produce SMF type 98 records for subtype 2 or greater. This exit will be called every 5 seconds starting at the top of the minute and for each 5-second multiple that follows when all of the following conditions are true:

- The WorkloadIntCorr feature is enabled in the product enablement policy (IFAPRDxx).
- SETSMF or SMFPRMxx has specified WIC.
- SMF is collecting type 98 subtype 2 or greater records.

Correlator data generation controls are all contained in the SMFPRMxx member of parmlib. When an SMF type 98 subtype record is not collected, the exploiter does not instrument correlator data and its exit is not called.

SMF type 98 subtype 1 (supervisor) records are not generated through IBM z/OS Workload Interaction Correlator and will be generated when all of the following conditions are true:

- SETSMF or SMFPRMxx has specified WIC or HFTSINTVL(5) or greater.
- SMF is collecting type 98 subtype 1 records.

IBM recommends configuring Hardware Instrumentation Services (HIS) to collect hardware counters as described in [Collecting CPU MF \(Counters\) on z/OS \(www.ibm.com/support/pages/system/files/inline-files/Detailed\\_CPU\\_MF\\_Counters\\_Enablement\\_Instructions\\_September\\_2019.pdf\)](http://www.ibm.com/support/pages/system/files/inline-files/Detailed_CPU_MF_Counters_Enablement_Instructions_September_2019.pdf). Collecting hardware counters enriches the data collected in SMF type 98 subtype 1 records.

IBM benchmarks did not detect additional measurable CPU overhead due to collecting HFTS data.

When you specify the WIC parameter, the HFTSINTVL parameter is ignored, and SMF type 98 subtype 1 records are generated every 5 seconds starting at the top of the minute and for each 5-second multiple that follows.

When you specify the NOWIC parameter, IBM z/OS Workload Interaction Correlator (WIC) is disabled on the system. The IFAWIC REQUEST=REGISTER service will fail for applications that call it. IFAWIC exploiters that are already registered for instrumentation will no longer have their exit routines called. When NOWIC is specified, HFTSINTVL is honored.

# IBM z/OS Workload Interaction Correlator

IBM z/OS Workload Interaction Correlator generates synchronized, standardized summary and exceptional data every 5 seconds, starting at the top of the minute and each 5-second multiple that follows. Correlator records include SMF type 98 and type 100 records collected with a 5-second interval.

Complete the following tasks to enable IBM z/OS Workload Interaction Correlator:

1. Enable the WorkLoadIntCorr feature of z/OS.
2. Collect workload interaction correlator records.

Optionally, complete the following tasks to write workload interaction correlator records to the z/OS file system (zFS) for analysis by analytics tools:

1. Record workload interaction correlator records to an in-memory resource.
2. Configure hardware instrumentation services (HIS) to write workload interaction correlator data to a z/OS UNIX file.

For more information about workload interaction correlator, see [z/OS Workload Interaction Correlator](#) in *z/OS MVS Programming: Authorized Assembler Services Guide*.

## Enabling the z/OS Workload Interaction Correlator feature

Complete this task to enable the IBM z/OS Workload Interaction Correlator feature.

### Procedure

1. If the product registration policy (IFAPRDxx) enables the IBM z/OS Advanced Data Gatherer feature as shown here, the z/OS Workload Interaction Correlator feature is also enabled. Continue with step “3” on page 9.

```
PRODUCT OWNER('IBM CORP')
NAME('z/OS')
ID(5655-ZOS)
FEATURENAME('ADV DG')
STATE(ENABLED)
```

**Note:** The ID value may vary over time with new versions (if any) of z/OS. See [Identifying modules, components, and products](#) in *z/OS MVS Diagnosis: Reference* for the product ID for the current version of z/OS.

2. If the product registration policy (IFAPRDxx) does not enable the z/OS Advanced Data Gatherer feature, follow the steps described in [Using dynamic enablement](#) in *z/OS Planning for Installation* to enable the z/OS Workload Interaction Correlator feature.

The following example shows the IFAPRDxx entry to enable the z/OS Workload Interaction Correlator feature:

```
PRODUCT OWNER('IBM CORP')
NAME('z/OS')
ID(5655-ZOS)
FEATURENAME(WorkloadIntCorr)
STATE(ENABLED)
```

**Note:** The ID value may vary over time with new versions (if any) of z/OS. See [Identifying modules, components, and products](#) in *z/OS MVS Diagnosis: Reference* for the product ID for the current version of z/OS.

3. If dynamically activating a new product registration policy (IFAPRDxx), issue the SET PROD=xx command, where xx identifies the updated IFAPRDxx parmlib member.

If no further action is taken, SMF will check the product registration policy at midnight to verify whether the z/OS Workload Interaction Correlator feature is enabled.

To activate the product registration immediately, reconfigure SMF to activate WIC, as described in "Collecting workload interaction correlator records".

When you activate WIC (see [SMFPRMxx \(system management facilities \(SMF\) parameters\)](#) in *z/OS MVS Initialization and Tuning Reference*) via a SET SMF=xx or SETSMF command, SMF also checks the product registration policy for the z/OS Workload Interaction Correlator feature.

4. Issue the DISPLAY SMF,WIC command to check the state of the z/OS Workload Interaction Correlator feature.

The following example shows the DISPLAY SMF,WIC command output:

```
SY1 IFA714I 15.01.51 SMF STATUS 885
SMF WIC STATUS
SPECIFIED SMF PARAMETER: WIC
WorkloadIntCorr PRODUCT FEATURE: ENABLED
ST   K PG # AS  ROUTINE  VERSION  R A E LAST ROUTINE CALL TIME
-----
00001          IEAHFXSV          Y    11/20/2021 15:01:50.192
KEY:
ST      - SUBTYPE NUMBER
K       - BUFFERKEY VALUE
PG      - NUM4KPAGES VALUE
# AS    - NUMBER OF ADDRESS SPACES REGISTERED
ROUTINE - EXIT ROUTINE NAME (CURRENT / PENDING)
VERSION - EXIT ROUTINE VERSION (IN HEXADECIMAL)
R       - SMF PARAMETERS REQUEST SUBTYPE
A       - EXIT ROUTINE TO BE CALLED NEXT INTERVAL
E       - ERROR ADDING EXIT ROUTINE
LAST... - TIME EXIT ROUTINE LAST CALLED
```

## What to do next

Complete the steps in "Collecting workload interaction correlator records."

## Collecting workload interaction correlator records

Complete this task to enable the collection of workload interaction correlator records.

### Before you begin

Complete the steps in "Enabling the z/OS Workload Interaction Correlator feature."

### Procedure

1. Configure SMF to activate WIC and enable the generation of all available SMF type 98 record subtypes with a 5-second interval, as described in "WIC – Specifying the generation of type 98 records for IBM z/OS Workload Interaction Correlator".
2. When Db2 12 has APAR PH18658 applied, activate the ZPARM STATIME\_MAIN=5 subsystem parameter to generate SMF type 100 records with a 5-second interval, as described in [MAIN STATS TIME field \(STATIME\\_MAIN subsystem parameter\) \(www.ibm.com/docs/en/db2-for-zos/12?topic=dtp-main-stats-time-field-statime-main-subsystem-parameter\)](#).
3. Specify the TYPE parameter to indicate that the system is to collect all correlator records.  
For details, see "TYPE and NOTYPE – Selecting and directing SMF records" and "SYS and SUBSYS with TYPE and NOTYPE – Selecting subtypes for SMF recording" in *z/OS MVS System Management Facilities (SMF)*.

For example, add the following statements to your SMFPRMxx configuration:

```
SYS(TYPE(98,100))
WIC
```

4. Issue the SET SMF=xx command, pointing to an SMFPRMxx parmlib member that contains these statements, to activate the configuration.
5. Optionally, issue the following commands to verify your configuration:
  - a) Issue the DISPLAY SMF,O command to see which SMF options are in effect.
  - b) Issue the DISPLAY SMF,WIC command to see the status of the WIC parameter and whether subtypes are being collected for registered IFAWIC exploiters and their subtypes.

## What to do next

Optionally, complete the steps in "Recording workload interaction correlator records to an in-memory resource."

## Recording workload interaction correlator records to an in-memory resource

Workload interaction correlator records can be recorded to an in-memory resource. Optionally, complete this task to configure the recording of workload interaction correlator records to an in-memory resource.

### Before you begin

Complete the steps in "Enabling the z/OS Workload Interaction Correlator feature" and "Collecting workload interaction correlator records".

### Procedure

1. Enable log stream recording, as described in [Setting up and managing SMF recording to logstreams in z/OS MVS System Management Facilities \(SMF\)](#).
2. Define an in-memory resource, as described in [Defining in-memory resources in z/OS MVS System Management Facilities \(SMF\)](#), and write workload interaction correlator records to the in-memory resource.

For example, to create an in-memory resource named IFASMF.WIC and write SMF type 98 and type 100 correlator records to it, add the following statement to the SMFPRMxx parmlib member:

```
INMEM(IFASMF.WIC,RESSIZMAX(2G),TYPE(98,100))
```

Then, issue the SET SMF=xx command to activate it.

3. To see existing in-memory resources, issue the DISPLAY SMF,M command.

## What to do next

Complete the steps in "Configuring HIS to write workload interaction correlator records to a z/OS UNIX file".

## Configuring HIS to write workload interaction correlator records to a z/OS UNIX file

Optionally, complete this task to configure hardware instrumentation services (HIS) to write workload interaction correlator records to a z/OS UNIX file.

### Before you begin

Complete the steps in "Recording workload interaction correlator records to an in-memory resource".

### About this task

Hardware instrumentation services (HIS) has a subcomponent that reads workload interaction correlator SMF records from an in-memory resource and writes them to a file on the z/OS UNIX file system. This function makes workload interaction correlator data available for tools that need to read the data from files, such as IBM z/OS Workload Insights Navigator.

### Procedure

1. Allocate and mount one or more zFS file systems so that they are writable by the *hisproc* address space.

- When allocating a zFS, determine its size based on 1 system with HIS keeping 14 days of workload interaction correlator records. See the `MODIFY hisproc,WIC,WICPATH` command in [Starting, configuring, and stopping hardware event collection in z/OS MVS System Commands](#).
- For a sysplex, IBM recommends creating a separate zFS for each member and mounting each as a read/write, sysplex-aware zFS with high availability and AUTOMOVE at `/global/wic/sysname`, where *sysname* represents the system name of each sysplex member. With this recommendation, each sysplex member's HIS procedure (*hisproc*) will regularly write its workload interaction correlator data to its member-specific zFS, which results in each member owning its zFS. Each zFS is recommended to be read/write, sysplex-aware with high availability and AUTOMOVE so that analysis products, such as IBM z/OS Workload Interaction Navigator, running on one sysplex member can analyze workload interaction correlator records and create analysis files for other active and inactive sysplex members.

For example, if each sysplex member has a zFS allocated in data set `OMVS.WIC.sysname.ZFS`, when each member joins the sysplex, use `SYS1.PARMLIB(BPXPRMxx)` or the `MOUNT` command to mount the zFS, as follows:

```
MOUNT
FILESYSTEM('OMVS.WIC.sysname.ZFS')
TYPE(ZFS) MODE(RDWR) AUTOMOVE(INCLUDE,*) PARM('RWSHARE,HA')
MOUNTPPOINT('/global/wic/sysname')
```

- For systems that are not in a sysplex, IBM recommends mounting a non-sysplex aware read/write zFS at `/global/wic/sysname`, where *sysname* represents the system name.

For example, if the system has a zFS allocated in data set `OMVS.WIC.sysname.ZFS`, use `SYS1.PARMLIB(BPXPRMxx)` or the `MOUNT` command to mount the zFS, as follows:

```
MOUNT
FILESYSTEM('OMVS.WIC.sysname.ZFS')
TYPE(ZFS) MODE(RDWR) UNMOUNT PARM('NORWSHARE')
MOUNTPPOINT('/global/wic/sysname')
```

Consider using the `&SYSNAME.` system symbol in place of *sysname* in the preceding examples.

For more information, see:

- [Statements and parameters for BPXPRMxx in z/OS MVS Initialization and Tuning Reference](#)
  - [Specifying zFS file systems as sysplex-aware and Specifying the high availability option for read/write sysplex-aware file systems in z/OS File System Administration](#)
  - [mount - Logically mount a file system in z/OS UNIX System Services Command Reference](#)
  - [What are system symbols?](#) and [Static system symbols in z/OS MVS Initialization and Tuning Reference](#)
2. Issue the `MODIFY hisproc,WIC` command to set an in-memory resource from which to read workload interaction correlator records, designated by the `INMEM` keyword, and the pre-defined z/OS UNIX path to which to write the records, designated by `WICPATH`.

For example, the following command reads from an SMF in-memory resource, `IFASMF.WIC`, and writes to the zFS at `/global/wic/sysname`:

```
MODIFY hisproc,WIC,INMEM=IFASMF.WIC,WICPATH='/global/wic/sysname'
```

where *hisproc* is the name of the procedure that started HIS.

Consider using the `&SYSNAME.` system symbol in place of *sysname* in the preceding example. For more information, see [What are system symbols?](#) and [Static system symbols in z/OS MVS Initialization and Tuning Reference](#).

3. Issue the `DISPLAY HIS,WIC` command to see the HIS state for the workload interaction correlator.

## Correlator SMF records

Correlator records include the following SMF record types:



- Record type 98 (X'62') — Workload interaction correlator and high-frequency throughput statistics
- Record type 100 (X'64') — Db2 statistics

## Record type 98 (X'62') — Workload interaction correlator and high-frequency throughput statistics

z/OS provides the SMF type 98 record for High-Frequency Throughput Statistics (HFTS) and IBM z/OS Workload Interaction Correlator data generation. An analytics application, such as IBM z/OS Workload Interaction Navigator, uses type 98 records as input for data analysis.

The SMF type 98 record contains multiple subtypes that are assigned when a z/OS component or middleware product exploits HFTS or z/OS Workload Interaction Correlator data generation. Type 98 subtype records are synchronized to the top of the minute for an interval, measured in seconds, as follows:

- **HFTS:** Based on the active SMFPRMxx member of parmlib and the value specified for the HFTSINTVL keyword. See "HFTSINTVL — Specifying the high-frequency throughput statistics interval", and see [SMFPRMxx \(system management facilities \(SMF\) parameters\)](#) in *z/OS MVS Initialization and Tuning Reference*.
- **z/OS Workload Interaction Correlator:** Based on the active SMFPRMxx member of parmlib and the WIC keyword. The specification of the WIC keyword forces HFTS data collection to a 5-second interval. See "WIC — Specifying the generation of type 98 records for IBM z/OS Workload Interaction Correlator", and see [SMFPRMxx \(system management facilities \(SMF\) parameters\)](#) in *z/OS MVS Initialization and Tuning Reference*.

Each type 98 subtype that is being collected produces 1 record per interval. With type 98 subtype records being collected every 5 seconds, each subtype will generate 17,280 records per day.

The following table summarizes the SMF type 98 subtypes:

Table 2. Summary of SMF type 98 subtypes

Subtype	Record owner	Available with HFTS	Available with Correlator	Average record size per interval	Total average record data per day <sup>1</sup>
1	z/OS supervisor	Yes	Yes	32 KB	550 MB
3	z/OS I/O supervisor	No	Yes	2 KB	35 MB
4	z/OS I/O supervisor	No	Yes	2 KB	35 MB
5	DFSMS	No	Yes <sup>2</sup>	32 KB	550 MB
6	DFSMS	No	Yes <sup>2</sup>	32 KB	550 MB
7	DFSMS	No	Yes <sup>2</sup>	32 KB	550 MB
8	DFSMS	No	Yes <sup>2</sup>	32 KB	550 MB
1024	CICS®	No	Yes	2 KB	35 MB
1025	IMS	No	Yes	2 KB	35 MB

### Notes:

1. The "Total average record data per day" value is based on the "Average record size per interval" estimate and 5-second intervals.
2. Requires that the active DEVSUPxx member contain the keyword SMF98\_DATASET\_ACTIVITY=YES.

IBM benchmarks did not detect additional measurable CPU overhead due to collecting SMF type 98 subtype records.

All SMF type 98 subtype records have a similar structure, as described in "Record mapping". "Subtype data section" contains different content for each subtype and is generally defined in "Subtypes for IBM z/OS Workload Interaction Correlator".

- SMF type 98 subtype records that are owned by z/OS components have their data sections defined in "Record type 98 (X'62') – Workload interaction correlator and high-frequency throughput statistics" in *z/OS MVS System Management Facilities (SMF)*.
- SMF type 98 subtype records that are owned by middleware products have their data sections defined in the documentation for the middleware product.

## Record environment

The following conditions exist for the generation of SMF type 98 records:

### Subtype 1

#### Macro

SMFEWTM, BRANCH=YES (record exit: IEFU84)

#### Storage residency

31-bit

## Record mapping

### Record header

This section contains the common SMF record type 98 header fields.

Offsets	Name	Length	Format	Description
0	0 SMF98LEN	2	binary	Record length. This field along with the next form the record descriptor word (RDW). For a details, see <a href="#">Standard and extended SMF record headers in z/OS MVS System Management Facilities (SMF)</a> .
2	2 SMF98SEG	2	binary	Segment descriptor. (See SMF98LEN field.)
4	4 SMF98FLG	1	binary	Header flags:  <div> <b>Bit</b>  <b>Meaning when set</b>  <b>0</b>  Subsystem identification follows system identification.  <b>1</b>  Subtypes are used.  <b>2-7</b>  Reserved. </div>
5	5 SMF98RTY	1	binary	Record type 98 (X' 62 ')
6	6 SMF98TME	4	binary	Time since midnight, in hundredths of a second, that the record was moved into the SMF buffer.
10	A SMF98DTE	4	packed	Date that the record was moved into the SMF buffer, in the form <i>0ccyydddF</i> . For a details, see <a href="#">Standard and extended SMF record headers in z/OS MVS System Management Facilities (SMF)</a> .
14	E SMF98SID	4	EBCDIC	System identification (from the SID parameter in the SMFPRMxx member of parm.lib).
18	12 SMF98SSI	4	EBCDIC	Subsystem identifier for the SMF address space ('STC' for started task).

Offsets	Name	Length	Format	Description
22	16 SMF98STY	2	binary	Record subtype: <b>Subtype</b> <b>Description</b> <b>1</b> Supervisor performance <b>3</b> I/O supervisor - FICON performance <b>4</b> I/O supervisor - zHyperLink performance <b>1024</b> IBM CICS <b>1025</b> IBM IMS
24	18 SMF98IND	1	binary	Additional record flags: <b>Bit</b> <b>Meaning when set</b> <b>0</b> This SMF record has multiple parts. There are more parts to come. <ul style="list-style-type: none"> <li>For a single part record, this bit is OFF.</li> <li>On the first part of a multiple-part record, bit 0 is ON and bit 1 is OFF.</li> <li>On subsequent parts for the same record, both bit 0 and bit 1 are ON.</li> <li>On the last record part, bit 0 is OFF and bit 1 is ON.</li> </ul> <b>1</b> This record is the continuation of the multiple-part record. This bit must be OFF on the first part of the multiple-part record. <b>2</b> Error: Storage was not available to generate more data in this SMF record. <b>3-7</b> Reserved.
25	19 SMF98PartSeqNo	1	binary	Record part sequence number, which identifies the order of the record part in a multiple-part SMF type 98 record. This value is meaningful only when a record has multiple parts (that is, SMF98IND bit 0 is ON for the first part of the record). The value is 0 for the first part, 1 for the next part, and so on.
26	1A SMF98SDSLen	2	binary	Length of the self-defining section.
28	1C SMF98SDSTripletsNum	2	binary	Number of triplets in the self-defining section.
30	1E *	18	binary	Reserved.

## Self-defining section

This section contains the triplet fields (offset, length, and number) that locate the other sections on the record. This triplet information should be checked prior to accessing a section of the record. The number triplet field is the primary indication of the existence of the field. This section is an extension of the header and physically follows it in the record. The offsets listed are from the start of the SMF record. The length of the self-defining section is described by SMF98SDSLen and the number of triplets is described by SMF98SDSTripletsNum.

Offsets	Name	Length	Format	Description
0	0 SMF98IOF	4	binary	Offset to the identification section

Offsets	Name	Length	Format	Description
4	4 SMF98ILN	2	binary	Length of the identification section
6	6 SMF98ION	2	binary	Number of identification sections
8	8 SMF98CSOF	4	binary	Offset to the context summary section
12	C SMF98CSLN	2	binary	Length of the context summary section
14	E SMF98CSON	2	binary	Number of context summary sections
16	10 SMF98DOF	4	binary	Offset to the subtype data section
20	14 SMF98DLN	2	binary	Length of the subtype data section
22	16 SMF98DON	2	binary	Number of subtype data sections

### Identification section

This section provides information to identify the source of the SMF type 98 records, including job name, step name, and start and stop times in SMF and TOD formats.

#### Triplet information:

You can locate this section in the record by using the following triplet fields which are found in the self-defining section:

##### Offset

SMF98IOF

##### Length

SMF98ILN

##### Number

SMF98ION

Offsets	Name	Length	Format	Description
0	0 SMF98JBN	8	EBCDIC	Job name.
8	8 SMF98RST	4	binary	Reporting interval start time (local time in hundredths of a second from midnight).
12	C SMF98RSD	4	packed	Reporting interval start date, in the form 0cyydddF.
16	10 SMF98STP	8	EBCDIC	Step name.
24	18 SMF98IntervalStart	8	TOD	Interval start time (local time in TOD format). You can convert to GMT by subtracting the value in the SMF98_CVTLDTO field.
32	20 SMF98IntervalEnd	8	TOD	Interval end time (local time in TOD format). You can convert to GMT by subtracting the value in the SMF98_CVTLDTO field.
40	28 SMF98SysName	8	EBCDIC	System Name when first byte not x'00'.
48	30 SMF98IntervalStart_ETOD	16	ETOD	The interval start time—local time in ETOD format. Can be converted to GMT by subtracting SMF98_ECVTLDT0 from this value.
64	40 SMF98IntervalEnd_ETOD	16	ETOD	The interval end time—local time in ETOD format. Can be converted to GMT by subtracting SMF98_ECVTLDT0 from this value.

### Context summary section

This section contains fields that describe the source of the subtype data records, including the level of the subtype exit and when the exit was run.

Use the SMF98\_ReleaseIndex, SMF98\_WithinReleaseIndex, and SMF98\_PrototypeIndex fields to determine the level of the macro you are using when traversing SMF type 98 records.

- SMF98\_ReleaseIndex is incremented when a subtype record in a product is significantly changed (such as adding new sections and fields for a new release of z/OS).
- SMF98\_WithinReleaseIndex is incremented when small changes are made to a subtype (such as adding a new field via an APAR).
- SMF98\_PrototypeIndex is incremented for any temporary changes for a given SMF98\_ReleaseIndex value and SMF98\_WithinReleaseIndex value, such as to denote changes to the SMF record for different versions of a temporary APAR fix (++APAR). SMF98\_PrototypeIndex is set to 0 for GA-level code.

#### Triplet information:

You can locate this section in the record by using the following triplet fields which are found in the self-defining section:

#### Offset

SMF98CSOF

#### Length

SMF98CSLN

#### Number

SMF98CSON

Offsets	Name	Length	Format	Description
0	0 SMF98_HftsInfo	8	binary	An 8-byte token that is equivalent across SMF 98 subtypes for the same interval.
8	8 SMF98_SubtypeInfo	24	binary	Subtype information.
8	8 SMF98_ReleaseIndex	2	binary	Release index. See the description earlier in this topic.
10	A SMF98_WithinReleaseIndex	2	binary	Within-release index. See the description earlier in this topic.
12	C SMF98_PrototypeIndex	2	binary	Prototype index. See the description earlier in this topic.
14	E *	2	binary	Reserved.
16	10 SMF98_Prodlevel	16	EBCDIC	Product level information. (z/OS components use CVTPROD.)
32	20 SMF98_ExitSerialTOD	8	binary	Time, in TOD units, used by the exit routine holding serialization. May be 0 if information is not available. This is the CPU time that serialization is held. It is obtained by calculating the TimeUsed delta before obtaining and after releasing the serialization.
40	28 SMF98_ExitTimeUsed	8	binary	Time, in TOD units, used by the exit routine up to the point when the SMF record is written. This is the CPU time for the exit (including the time spent holding serialization). It is obtained by calculating the TimeUsed delta from when the exit is entered until the exit writes the record. For continuation records, the last record will have the total time used by the exit.
48	30 SMF98_CVTLDTO	8	TOD	Offset value needed to adjust the TOD value to the local date and time of day. Add this offset to a GMT value to get the local date/time value. Subtract this value from a local TOD value to get the GMT value.
56	38 SMF98_CVTLSO	8	TOD	Leap second offset value needed to adjust TOD values to and from a system clock time. Times in the SMF98 record are already incremented with leap seconds; however, other system times might not be adjusted. Add or subtract this offset to allow for time comparisons with TOD values that are not adjusted for leap seconds.
64	40 SMF98_ECVTLDTO	16	ETOD	Offset value needed to adjust the ETOD value to the local date and time of day. Add this offset to a GMT value to get the local date and time. Subtract this offset from a local ETOD value to get a GMT date and time.

Offsets	Name	Length	Format	Description
80 50	SMF98_ECVTISO	16	ETOD	Leap second offset value needed to adjust ETOD values to and from a system clock time. Times in the SMF type 98 record are already incremented with leap seconds; however, other system times may not be adjusted. Add or subtract this offset to allow for time comparisons with ETOD values not adjusted for leap seconds.

### ***Subtype data section***

This section provides the data for each subtype and is mapped according by subtype. (Refer to the mappings for each subtype.)

#### **Triplet information:**

You can locate this section in the record by using the following triplet fields which are found in the self-defining section:

#### **Offset**

SMF98DOF

#### **Length**

SMF98DLN

#### **Number**

SMF98DON

## **Subtypes for IBM z/OS Workload Interaction Correlator**

IBM z/OS Workload Interaction Correlator produces SMF type 98 records for subtypes that are registered with the IFAWIC service. These mappings are incomplete and depend on the particular data to be recorded for each exploiter. Use the mapping produced by the subtype holder to complete each record section.

### ***Data section***

The SMF type 98 data section for WIC subtypes begins with the number of triplets (SMF98WicData\_TripletsNum) and the length of the data triplets area (SMF98WicData\_TripletsLen). Check this triplet information prior to accessing a section of the record. The SMF98WicData\_TripletsNum field is the primary indication of the existence of a section.

All offset values are from the start of the record.

Offsets	Name	Length	Format	Description
0 0	SMF98WicData_TripletsNum	4	binary	Number of data triplets
4 4	SMF98WicData_TripletsLen	4	binary	Length of the data triplets area (SMF98WicData_TripletsArea)
8 8	SMF98WicData_TripletsArea	32	binary	Data triplet area
8 8	SMF98WicData_AggBucket1	8	binary	First aggregate bucket triplet
8 8	SMF98WicData_AggBucketOF1	4	binary	Offset to the first aggregate bucket section from the beginning of the record
12 C	SMF98WicData_AggBucketLN1	2	binary	Length of the first aggregate bucket section
14 E	SMF98WicData_AggBucketON1	2	binary	Number of first aggregate bucket sections
16 10	SMF98WicData_AggBucket2	8	binary	Second aggregate bucket triplet
16 10	SMF98WicData_AggBucketOF2	4	binary	Offset to the second aggregate bucket section from the beginning of the record
20 14	SMF98WicData_AggBucketLN2	2	binary	Length of the second aggregate bucket section
22 16	SMF98WicData_AggBucketON2	2	binary	Number of second aggregate bucket sections

Offsets	Name	Length	Format	Description
24 18	SMF98WicData_JobIndex	8	binary	Job index triplet: contains indexes that reference jobs within the job list data
24 18	SMF98WicData_JobIndexOF	4	binary	Offset to the first job index section from the beginning of the record
28 1C	SMF98WicData_JobIndexLN	2	binary	Length of each job index section
30 1E	SMF98WicData_JobIndexON	2	binary	Number of job index sections
32 20	SMF98WicData_JobList	8	binary	Job list triplet: list of unique ASIDs and job names
32 20	SMF98WicData_JobListOF	4	binary	Offset to the first job list section from the beginning of the record
36 24	SMF98WicData_JobListLN	2	binary	Length of each job list section
38 26	SMF98WicData_JobListON	2	binary	Number of job list sections

### Aggregate bucket sections

The aggregate buckets contain summary data about an IFAWIC exploiter's use of shared resources across multiple address spaces. The aggregate bucket is created from data that an IFAWIC exploiter's exit routine passes to an SMF service to generate SMF type 98 records.

Every IFAWIC exploiter instruments activities in its address space specific buffer. At the end of every 5 second interval, each IFAWIC exploiter's exit routine is called and aggregates job activities into aggregate buckets by CPU type, job priority, and job size.

**Note:** The SMF 98 records contain buckets that have more than one contributor (SMF98AggBucketEA\_Contributors > 0).

You can locate this section in the record by using the following triplet fields which are found in the SMF98WicData section:

#### Offset

SMF98WicData\_AggBucketOF1 or SMF98WicData\_AggBucketOF2

#### Length

SMF98WicData\_AggBucketLN1 or SMF98WicData\_AggBucketLN2

#### Number

SMF98WicData\_AggBucketON1 or SMF98WicData\_AggBucketON2

Note that there are two possible aggregate bucket sections per WIC subtype.

Offsets	Name	Length	Format	Description
0 0	SMF98AggBucket	*	binary	Aggregate bucket.
0 0	SMF98AggBucketHdr	6	binary	Aggregate bucket header.
0 0	SMF98AggBucketHdr_CPUType	2	binary	CPU type for this bucket. Valid values are:  <div> <b>Value (Constant)</b>  <b>Meaning</b>  <b>0 (PSAProcClass_CP)</b>  CP  <b>4 (PSAProcClass_zIIP)</b>  zIIP </div>

Offsets	Name	Length	Format	Description
2	2 SMF98AggBucketHdr_JobPriority	2	binary	Job priority. Valid values are: <b>Value (Constant)</b> <b>Meaning</b> <b>FFFF (SMF98_kPriorityBucket_All)</b> All <b>1 (SMF98_kPriorityBucket_1)</b> Critical <b>2 (SMF98_kPriorityBucket_2)</b> High <b>3 (SMF98_kPriorityBucket_3)</b> Low <b>4 (SMF98_kPriorityBucket_4)</b> Discretionary
4	4 SMF98AggBucketHdr_JobSize	2	binary	Job size relative to other jobs in this system for this job priority, for this CPU type. Valid values are: <b>Value (Constant)</b> <b>Meaning</b> <b>FFFF (SMF98_kConsume_SubBucket_All)</b> All <b>1 (SMF98_kConsume_SubBucket_1)</b> Large <b>2 (SMF98_kConsume_SubBucket_2)</b> Medium <b>3 (SMF98_kConsume_SubBucket_3)</b> Small <b>4 (SMF98_kConsume_SubBucket_4)</b> Tiny
4	4 SMF98AggBucketHdr_CustomJobSize	2	binary	Custom job size chosen by the exploiter. See SMF type 98 subtype documentation for values, meanings, and constants.
6	6 SMF98AggBucketExpArea	*	binary	Aggregate bucket exploiter area.
6	6 SMF98AggBucketEA_Contributors	2	binary	Number of address spaces contributing to this bucket entry.
8	8 SMF98AggBucketEA_Data	*	binary	Exploiter data. See the exploiter of the IFAWIC service for their data mapping.

### Exceptional job sections

For each aggregate bucket activity, an IFAWIC exploiter may provide an exceptional job with its corresponding activities and job name.

### Exceptional job index section

The SMF 98 exceptional job index section consists of buckets (by CPU type, job priority, and job size) with an activity-specific exceptional job index that locates the most exceptional job in the job list.

The SMF98JobIdxActivityArea field is a list of 2-byte fields that represent activity-specific indices into the SMF 98 job list section. These indices start at 1. The offset for this exceptional job is calculated as follows, where  $n$  is the index number:

$$\text{SMF98WicData\_JobList0F} + (\text{SMF98WicData\_JobListLN} * (n - 1))$$

Offsets	Name	Length	Format	Description
0	0 SMF98JobIdx	*	binary	Job index data area.
0	0 SMF98JobIdxHdr	6	binary	Job index header.



Offsets	Name	Length	Format	Description
0	0 SMF98JobIdxHdr_CPUType	2	binary	CPU type for this bucket. Valid values are: <b>Value (Constant)</b> <b>Meaning</b> <b>0 (PSAProcClass_CP)</b> CP <b>4 (PSAProcClass_zIIP)</b> zIIP
2	2 SMF98JobIdxHdr_JobPriority	2	binary	Job priority. Valid values are: <b>Value (Constant)</b> <b>Meaning</b> <b>FFFF (SMF98_kPriorityBucket_All)</b> All <b>1 (SMF98_kPriorityBucket_1)</b> Critical <b>2 (SMF98_kPriorityBucket_2)</b> High <b>3 (SMF98_kPriorityBucket_3)</b> Low <b>4 (SMF98_kPriorityBucket_4)</b> Discretionary
4	4 SMF98JobIdxHdr_JobSize	2	binary	Job size relative to other jobs in this system for this job priority, for this CPU type. Valid values are: <b>Value (Constant)</b> <b>Meaning</b> <b>FFFF (SMF98_kConsume_SubBucket_All)</b> All <b>1 (SMF98_kConsume_SubBucket_1)</b> Large <b>2 (SMF98_kConsume_SubBucket_2)</b> Medium <b>3 (SMF98_kConsume_SubBucket_3)</b> Small <b>4 (SMF98_kConsume_SubBucket_4)</b> Tiny
4	4 SMF98JobIdxHdr_CustomJobSize	2	binary	Custom job size chosen by the exploiter. See SMF type 98 subtype documentation for values, meanings, and constants.
6	6 SMF98JobIdxActivityArea	*	binary	Exploiter activity index area. Each exploiter activity is assigned a 2-byte index. This area is an array of 2-byte indices. These indices start at 1 and describe the entry number into the SMF 98 job list section (SMF98JobList) with which to find the exceptional job corresponding to the exploiter-defined activity. See the exploiter's subtype definition for details.

## Exceptional job list section

The SMF 98 exceptional job list section, mapped by SMF98JobList, contains the job name, address space ID number, and activity data for exceptional jobs.

Offsets	Name	Length	Format	Description
0	0 SMF98JobList	*	binary	Job list data area.
0	0 SMF98JobListHdr	10	binary	Job list header.
0	0 SMF98JobListHdr_ASID	2	binary	Address space ID (ASID).

Offsets	Name	Length	Format	Description
2	2 SMF98JobListHdr_JobName	8	EBCDIC	Job name.
10	A SMF98JobListActivityArea	*	binary	Exploiter job activity area. See the exploiter of the IFAWIC service for their data mapping.

## Record type 100 (X'64') – Db2 statistics

IBM Db2® writes record type 100 to record transaction data collected at event monitoring points. For more information about record type 100, see the appropriate edition of *IBM DB2® for z/OS Managing Performance* for your version of Db2.

### Generating correlator records

Db2 12 with APAR PH18658 can activate the ZPARM STATIME\_MAIN=5 subsystem parameter to generate SMF type 100 correlator records with a 5-second interval, as described in MAIN STATS TIME field (STATIME\_MAIN subsystem parameter) ([www.ibm.com/docs/en/db2-for-zos/12?topic=dtpp-main-stats-time-field-statime-main-subsystem-parameter](http://www.ibm.com/docs/en/db2-for-zos/12?topic=dtpp-main-stats-time-field-statime-main-subsystem-parameter)). Correlator records can be provided to an analytics application, such as IBM z/OS Workload Interaction Navigator, with these APARs: PH39140 (Db2 12), PH39141 (Db2 12), OA61924 (z/OS), and OA61618 (Navigator).

When generating correlator records, each SMF type 100 record subtype produces 1 record per Db2 instance per 5-second interval. So, with 1 Db2 instance generating SMF type 100 records every 5 seconds, each subtype would produce 1 record every 5 seconds, which is equivalent to 17,280 records per day. SMF type 100 subtype sizes are dependent upon configuration and workload.

Table 3 on page 22 summarizes the SMF type 100 subtype sizes.

Table 3. Summary of SMF type 100 subtypes		
Subtype	Average record size per interval	Total average record data per day <sup>1</sup>
0	4 KB	70 MB
1	16 KB	280 MB
2	1 KB	18 MB
3	1 KB	18 MB
4	2 KB	35 MB
<b>Notes:</b> 1. The "Total average record data per day" value is based on the "Average record size per interval" estimate and 5-second intervals.		

---

## Chapter 3. Writing programs that make use of Workload Interaction Correlator

The following topic presents information about the operation of IBM z/OS Workload Interaction Correlator. This information is useful in writing programs that make use of workload interaction correlator services.

For related information, see *z/OS MVS Authorized Assembler Services Guide*.

---

### IBM z/OS Workload Interaction Correlator

IBM z/OS Workload Interaction Correlator (WIC) enables a product, such as a z/OS component, middleware, or application that executes in multiple instances distributed across multiple address spaces, to generate high-frequency, summarized, system-wide and compartmental data. The product can use the IBM z/OS Workload Interaction Correlator services to generate an SMF type 98 record for the product's subtype every 5 seconds containing data about the product's activities in a standardized, synchronized, contextualized format.

To exploit the IBM z/OS Workload Interaction Correlator services, a product issues the IFAWIC service. The IFAWIC service allows an exploiter to register a product's subtype for instrumentation, receive an instrumentation buffer, and provide an exit routine. The IFAWIC exploiter provides an IBM-assigned subtype number to distinguish the system function. The subtype number is used by the exploiter's exit routine when writing SMF type 98 records. If your product wishes to exploit workload interaction correlator services, contact IBM to be assigned a subtype number. Subtype numbers in the range 8192 – 8199 are available for test purposes. Production environments must use an official IBM-assigned subtype number.

When registering for instrumentation, the exploiter specifies the desired characteristics for the workload interaction correlator buffer. The exploiter can use the BUFFERKEY parameter to specify that the buffer is in key 0 or key 8 shared storage. The exploiter can use the BUFFER4KPAGES parameter to specify the size of the shared buffer. The system makes the instrumentation storage available to the exploiter's primary address space to instrument into the buffer, and to the SMF address space for use by the exploiter's exit routine.

The IFAWIC REGISTER request returns a buffer to the caller to use for instrumenting exploiter resource use activities for the exploiter's primary address space. The REGISTER request must be used in each address space in which an exploiter runs. This supplies each address space with a distinct buffer for instrumenting address space activities.

The IFAWIC caller provides an exit routine that SMF calls during the WIC processing. This exit routine is responsible for aggregating and summarizing activities from all address spaces in which an exploiter runs. The exit routine is also responsible for preparing and writing an SMF type 98 record for an exploiter's subtype.

#### Workload interaction correlator exploiter exit routine

When the IFAWIC REQUEST=REGISTER service is invoked, the exploiter provides a WIC exit routine that SMF will call every five seconds during WIC processing. The exit routine aggregates and summarizes activities from all address spaces in which the exploiter is registered. The results are prepared into aggregate and exceptional buckets and then written to an SMF type 98 record for an exploiter-specific subtype number.

## Environment

During processing of the IFAWIC service, the caller's exit routine is copied into storage that SMF can later use to call the exit routine. The exit routine is then called from the SMF address space with the following attributes:

<b>Mode:</b>	Enabled for I/O and external interrupts
<b>State:</b>	Supervisor
<b>Key:</b>	0
<b>Addressing mode:</b>	31- or 64-bit, depending on how the module was linked (64 is recommended)
<b>Locks held:</b>	None
<b>Address space control:</b>	ASC mode
<b>Memory mode:</b>	Primary = Secondary = Home
<b>Dispatch mode:</b>	Dispatchable unit mode
<b>Address space:</b>	SMF

## Input register information

Register contents at the time of entry to the WIC exit routine are as follows:

Register	Contents
GR0	Address of an 8 K dynamic work area in 31-bit storage Note that only bits 32-63 of GR0 are set; do not rely on bits 0-31 being 0.
GR1	Address of the WicParm, mapped by IFAWICCB
GR2 - GR12	Irrelevant
GR13	Address of a 144-byte save area
GR14	Return address
GR15	Entry point address

## Output register information

Register contents at the time of return from the WIC exit routine must be as follows:

Register	Contents
GR0	Irrelevant
GR1 - GR14	Unchanged from entry
GR15	Must be 0

## Parameters

The WIC exploiter exit routine is called every 5 seconds. Upon entry to the WIC exploiter exit routine, general purpose register 1 contains the address of the WicParm, mapped by IFAWICCB. WicParm is defined as follows:

Field	Length in bytes	Description
WicParm_Version	2	Version of the WIC parameter list
WicParm_Length	2	Length of the WIC parameter list
WicParm_Subtype	2	Record subtype being processed

Field	Length in bytes	Description
WicParm_Request	2	Request to either prime or write record  <b>WicParm_ReqPrime</b> This is a request for the WIC exploiter exit routine to prepare for writing a record on the next interval by clearing the WIC instrumentation buffer in each address space. The WIC exploiter exit routine must not produce a record for this interval.  <b>WicParm_ReqWriteRecord</b> This is a request for the WIC exploiter exit routine to write an SMF type 98 record for the given subtype.
WicParm_DynAreaPtr	8	The 64-bit address of the dynamic area for the WIC exploiter exit routine
WicParm_DynAreaLen	4	The length of the dynamic area, at least 1 MB in size
WicParm_WICSrv_GetNextBufferPtr	8	The 64-bit address of the get next buffer service of the WIC exit routine
WicParm_WICSrv_ClearBufferPtr	8	The 64-bit address of the clear buffer service of the WIC exit routine
WicParm_WICSrv_WriteSMF98Ptr	8	The 64-bit address of the build and write SMF98 record service of the WIC exit routine

## Workload interaction correlator exit routine services

Exit routine services aid a WIC exit routine in processing and preparing data for all the address spaces for which an exploiter issued an IFAWIC REQUEST=REGISTER request. The exit routine services are:

- Get next buffer — returns either the first or next WIC buffer for a given subtype.
- Clear buffer — clears a portion of a WIC buffer for a given subtype.
- Build and write SMF98 record — builds and writes an SMF type 98 record.

The addresses of these services can be found in WicParm, mapped in IFAWICCB.

## Environment

Callers of the WIC exit routine services must be running in 64-bit addressing mode (AMODE 64) and primary address space control mode at the time the WIC exit routine services are invoked.

Data to be passed by the exploiter exit routine to the exit routine services can reside above the 2G bar.

## Programming requirements

WIC exit routine services do not provide recovery. The invoker of the WIC exit routine services is responsible for providing recovery.

## Input register information

Register contents at the time of invocation must be as follows:

Register	Contents
GR0	Irrelevant
GR1	Address of the parameter list for the WIC exit routine service
GR2 - GR12	Irrelevant
GR13	Address of a 144-byte save area

Register	Contents
GR14	Return address
GR15	Irrelevant

## Output register information

Register contents at the time of return are as follows:

Register	Contents
GR0	Used by WIC exit routine services
GR1 - GR14	Unchanged
GR15	Used by WIC exit routine services

## WIC exit routine service parameter list

On entry to WIC exit routine services, the address of the WIC exit routine service parameter list must be placed in general purpose register 1 (GR1) and addresses three doublewords as follows:

### Doubleword 1

Address of specific WIC exit routine service parameter list (mapped by IFAWICCB)

### Doubleword 2

Address of a 31-bit area that will contain a return code from the WIC exit routine service

### Doubleword 3

Address of a 31-bit area that will contain a reason code from the WIC exit routine service

Assembler callers can invoke the WIC exit routine services by using the CALL macro, as shown in [Figure 1 on page 26](#):

```

        SYSSTATE AMODE64=YES
* code to get address of WicParm into register x
:
        USING WicParm,x
:
        LG 15,WicParm_WICSrv_GetNextBufferPtr
        CALL (15),(WicNBParms,WicSrvRet,WicSrvRsn),MF=(E,PLAREA)
:
WicNBParms DS CL(L'WicNB_ParmList)
WicSrvRet DS F
WicSrvRsn DS F
PLAREA DS 3D
:

```

Figure 1. Sample assembler program to call a WIC exit routine service

## WIC exit routine service — Get next buffer

The get next buffer service returns the first or next WIC buffer for a given subtype. Get next buffer is meant to be used iteratively to return and process each of the WIC buffers for a given subtype. On the initial call to this service, the caller should set WicNB\_inoutASID to 0. On subsequent calls to the service, the caller should not change WicNB\_inoutASID from what was returned by the previous call.

The get next buffer service does not return address spaces that are swapped out. A WIC exit routine must tolerate not receiving data from address spaces that are swapped out.

## Parameter list

The get next buffer service uses the WicNB\_ParmList (mapped by IFAWICCB). The WicNB\_ParmList has the following format:

### WicNB\_Version

A 2-byte field that identifies the version of the parameter list. This should be set to the WicNB\_#Ver equate.

### WicNB\_inoutASID

A 2-byte input and output field that contains the address space ID. The invoker of the get next buffer service must initialize this field to zero for the initial invocation. When the get next buffer service returns with a return code of WicSrv\_Rc\_Success (0), this field contains the address space ID for the WIC buffer returned in the WicNB\_outBuffer@ field. For subsequent invocations of the get next buffer service, the caller should not change the contents of this parameter in order to obtain the next WIC buffer to process. When no additional WIC buffers are found, the service returns a return code of WicSrv\_Rc\_Warning and a reason code of WicSrv\_Rsn\_NoMoreBuffers.

### WicNB\_Reserved1

A 20-byte reserved area; must be binary zeros.

### WicNB\_ASInfo

A 16-byte output area with contents as follows:

#### WicNB\_outJobName (bytes 0 - 7)

An 8-byte job name for the address space ID contained in WicNB\_inoutASID.

#### WicNB\_outJobPriorityArea (bytes 8 - 15)

An 8-byte area that describes the job priority bucket and the job size bucket, as follows:

##### WicNB\_outJobPriorityBucket (bytes 0 - 1)

A 2-byte area that describes the relative priority of the address space ID in WicNB\_inoutASID. Possible values are in the range 1 - 4. Refer to IHASVT for descriptions of the priority buckets, specifically the SVT\_kHiPriorityBucketIndex, SVT\_kMedPriorityBucketIndex, SVT\_kLowPriorityBucketIndex, and SVT\_kDiscPriorityBucketIndex fields.

##### WicNB\_outJobSizeBucket (bytes 2 - 7)

A 6-byte area that contains the job size sub-buckets associated with the given job priority bucket. This area is defined as three 2-byte entries in the following format:

##### Bytes 0 - 1

The job size for the CP CPU type.

##### Bytes 2 - 3

Reserved.

##### Bytes 4 - 5

The job size for the zIIP CPU type.

### WicNB\_outBufferAddr

An 8-byte output area that will contain the starting address of the WIC buffer for the address space ID in the WicNB\_inoutASID field.

## Return and reason codes

The following hexadecimal return and reason codes are associated with the get next buffer service:

Return code (hex)	Reason code (hex)	Meaning and action
0 WicSrv_Rc_Success	0 WicSrv_Rc_Success	<b>Meaning:</b> The get next buffer service has returned the first or next WIC buffer. <b>Action:</b> None.
4 WicSrv_Rc_Warning	401 WicSrv_Rsn_NoMoreBuffers	<b>Meaning:</b> The get next buffer service was unable to find the first or next WIC buffer. <b>Action:</b> Callers of the WIC get next buffer service should terminate current processing and move on to the next phase of exit routine processing.

Return code (hex)	Reason code (hex)	Meaning and action
8 WicSrv_Rc_UserError	801 WicSrv_Rsn_BadParmList	<b>Meaning:</b> An invalid parameter list was built and supplied to the get next buffer service. <b>Action:</b> Verify that the parameter list for the called service is properly built and initialized.

## WIC exit routine service — Clear buffer

The clear buffer service can be used to clear a portion of a given WIC buffer area.

### Parameter list

The clear buffer service uses the WicCB\_ParmList (mapped by IFAWICCB). The WicCB\_ParmList has the following format:

#### WicCB\_Version

A 2-byte field that identifies the version of the parameter list. This should be set to the constant, WicCB\_#Ver.

#### WicCB\_inASID

A 2-byte input field that contains the address space ID. The address space ID is obtained from the invocation of the get next buffer service.

#### WicCB\_Reserved1

A 4-byte reserved area.

#### WicCB\_inStartAddrBufToClear

An 8-byte input field that contains the address within the WIC buffer to clear for the subtype and given address space ID (WicCB\_inASID). The WIC buffer address is obtained by calling the get next buffer service.

#### WicCB\_inBufLenToClear

A 4-byte input field that contains the length of the WIC buffer to be cleared, from the starting address specified by WicCB\_inStartAddrBufToClear.

### Return and reason codes

The following hexadecimal return and reason codes are associated with the clear buffer service:

Return code (hex)	Reason code (hex)	Meaning and action
0 WicSrv_Rc_Success	0 WicSrv_Rc_Success	<b>Meaning:</b> The clear buffer service has cleared the WIC buffer starting at the specified address for the specified length. <b>Action:</b> None.
4 WicSrv_Rc_Warning	401 WicSrv_Rsn_ClearSTAddrBufLenOutOfRange	<b>Meaning:</b> The caller of the clear buffer service specified a buffer length to clear that, when combined with the starting address within the WIC buffer, extends beyond the buffer size requested when the IFAWIC REGISTER request was issued for the subtype. The WIC buffer area starting at the address to clear is cleared (set to binary zeros). <b>Action:</b> To avoid this warning, re-register the subtype and specify a larger buffer size.
8 WicSrv_Rc_UserError	801 WicSrv_Rsn_BadParmList	<b>Meaning:</b> An invalid parameter list was built and supplied to the clear buffer service. <b>Action:</b> Verify that the parameter list for the called service is properly built and initialized.



Return code (hex)	Reason code (hex)	Meaning and action
8 WicSrv_Rc_UserError	802 WicSrv_Rsn_ BadClearSTAddrBufLen	<p><b>Meaning:</b> The caller of the clear buffer service specified a starting WIC buffer address and buffer length to clear that extends beyond the maximum size of a WIC buffer. The buffer area is not cleared.</p> <p><b>Action:</b> Verify that the starting buffer address and buffer length to clear are correct.</p>

## WIC exit routine service — Build and write SMF98 record

The build and write SMF98 record service builds and writes an SMF type 98 record.

### Parameter list

The build and write SMF98 record service uses the WicWR\_ParmList (mapped by IFAWICCB). The WicWR\_ParmList has the following format:

#### WicWR\_Version

A 2-byte field that identifies the version of the parameter list. This should be set to the current version constant, WicWR\_kVer1.

#### WicWR\_Rsvd02

A 4-byte reserved field.

#### WicWR\_Flag

A 1-byte input field where:

- Bit 0, **WicWR\_FlagExcJobsUniqueByJobSize**, controls whether multiple instances of the same job are compressed into a single instance (off, default) or per-job-size instance (on).

#### WicWR\_CustomJobSizeEntries

A 1-byte input field that controls whether to use the default or exploiter-defined job size entries and attributes.

When this field is zero, the default job sizes are applied. When this field is non-zero, the required job size bucket entry (1) and attribute (All) are applied, and the remaining job size bucket entries (1–4) and attributes (Large, Medium, Small, Tiny) are replaced with exploiter-defined job size entries from 1 to WicWR\_CustomJobSizeEntries and attributes. WicWR\_kCustomJobSizeEntriesMax is the maximum value that can be specified.

#### WicWR\_SubtypeInfo

An 24-byte input field that contains the SMF type 98 subtype information in the following format:

##### WicWR\_ReleaseIndex (bytes 0 - 1)

A 2-byte input field that contains the current release version index value.

##### WicWR\_WithinReleaseIndex (bytes 2 - 3)

A 2-byte input field that contains the current within-release index value.

##### WicWR\_PrototypeIndex (bytes 4 - 5)

For IBM use only, a 2-byte input field that contains the prototype index value.

##### WicWR\_Rsvd0D (bytes 6 - 7)

A 2-byte reserved field; must be binary zeros.

##### WicWR\_ProdLevel (bytes 8 - 23)

A 16-byte input field that contains the program product level, in the following format:

###### Bytes 0 - 7

Product name

###### Bytes 8 - 15

FMID

**WicWR\_AggregateParmList1**

A 24-byte input field that contains the first aggregate bucket parameter list. See [“Aggregate bucket parameter list”](#) on page 30 for details.

**WicWR\_AggregateParmList2**

A 24-byte input field that contains the second aggregate bucket parameter list. See [“Aggregate bucket parameter list”](#) on page 30 for details.

**WicWR\_ExceptionBucketParmList**

A 24-byte input field that contains the exception bucket parameter list. See [“Exception bucket parameter list”](#) on page 30 for details.

**Aggregate bucket parameter list**

The aggregate bucket parameter list has the following format:

**WicWR\_AggBucketNumOfEntries**

A 4-byte input field that contains the number of aggregate bucket entries.

**WicWR\_AggBucketEntryLength**

A 4-byte input field that contains the total length of one entry or instance in the aggregate bucket exploiter area. For more information, see [“Aggregate buckets”](#) on page 32.

**WicWR\_AggBucketEntryWriteLength**

A 4-byte input field that contains the write length of one entry or instance in the aggregate bucket exploiter area. For more information, see [“Aggregate buckets”](#) on page 32.

**WicWR\_Rsvd2B**

A 4-byte reserved area; must be binary zeros.

**WicWR\_AggBucketStartAddr**

An 8-byte input field that contains the starting address of the aggregate bucket. For more information, see [“Aggregate buckets”](#) on page 32.

**Exception bucket parameter list**

The exception bucket parameter list has the following format:

**WicWR\_ExcBucketNumOfEntries**

A 4-byte input field that contains the number of entries or instances in the exception bucket.

**WicWR\_ExcBucketNumOfActivities**

A 4-byte input field that contains the number of activities for the exception bucket.

**WicWR\_ExcBucketEntryLength**

A 4-byte input field that contains the total length of one entry or instance in the exception bucket exploiter area. For more information, see [“Exception bucket”](#) on page 36.

**WicWR\_ExcBucketEntryWriteLength**

A 4-byte input field that contains the write length of one entry or instance in the exception bucket exploiter area. For more information, see [“Exception bucket”](#) on page 36.

**WicWR\_ExcBucketStartAddr**

An 8-byte input field that contains the starting address of the exception bucket. For more information, see [“Exception bucket”](#) on page 36.

**Return and reason codes**

The following hexadecimal return and reason codes are associated with the build and write SMF98 record service:

Return code (hex)	Reason code (hex)	Meaning and action
0 WicSrv_Rc_Success	0 WicSrv_Rc_Success	<b>Meaning:</b> The build and write SMF98 record service has successfully built and written an SMF type 98 record based on the supplied input. <b>Action:</b> None.
4 WicSrv_Rc_Warning	401 WicSrv_Rsn_ ApproachingMaxSMF98RecLen	<b>Meaning:</b> The build and write SMF98 record service has determined that the SMF98 record is approaching the 32756 record length limit. <b>Action:</b> Verify that the number of records for the aggregate buckets and the exception bucket and the write lengths for the buckets are correct. If the number of records and write lengths are correct, consider reducing the amount of data recorded for each aggregate bucket record and/or exception bucket record.
8 WicSrv_Rc_UserError	801 WicSrv_Rsn_BadParmList	<b>Meaning:</b> An invalid parameter list was built and supplied to the build and write SMF98 record service. <b>Action:</b> Verify that the parameter list for the called service is properly built and initialized.
8 WicSrv_Rc_UserError	803 WicSrv_Rsn_ AggBucketsNotSpecified	<b>Meaning:</b> The caller of the build and write SMF98 record service specified a zero address for both aggregate bucket addresses within the aggregate bucket portion of the parameter list. <b>Action:</b> Verify that at least one of the aggregate bucket addresses is specified and is not zero.
8 WicSrv_Rc_UserError	804 WicSrv_Rsn_ BadNumOfAggBucketEntries	<b>Meaning:</b> The caller of the build and write SMF98 record service specified an incorrect number of aggregate bucket entries. <b>Action:</b> Specify a correct number of aggregate bucket entries.
8 WicSrv_Rc_UserError	805 WicSrv_Rsn_ AggBucketWRLenGTOneBucLen	<b>Meaning:</b> The caller of the build and write SMF98 record service specified a write length for one aggregate bucket entry that is greater than one aggregate bucket entry length. <b>Action:</b> Verify that the write length of one aggregate bucket entry is defined to be less than the length of one aggregate bucket entry.
8 WicSrv_Rc_UserError	807 WicSrv_Rsn_ ExcBucketNotSpecified	<b>Meaning:</b> The caller of the build and write SMF98 record service specified a zero address for the exception bucket address, or specified a zero for the number of entries contained in the exception bucket, or specified a zero for the number of activities within the exception bucket portion of the parameter list. <b>Action:</b> Verify that the exception bucket address is specified and is not zero, that the number of entries is not zero, and that the number of activities is not zero.
8 WicSrv_Rc_UserError	808 WicSrv_Rsn_ BadNumOfExcBucketEntries	<b>Meaning:</b> The caller of the build and write SMF98 record service specified an incorrect number of exception bucket entries. <b>Action:</b> Specify a correct number of exception bucket entries.
8 WicSrv_Rc_UserError	809 WicSrv_Rsn_ ExcBucketWRLenGTOneBucLen	<b>Meaning:</b> The caller of the build and write SMF98 record service specified a write length for one exception bucket entry that is greater than one exception bucket entry length. <b>Action:</b> Verify that the write length of one exception bucket entry is defined to be less than the length of one exception bucket entry.
8 WicSrv_Rc_UserError	80A WicSrv_Rsn_ SMF98RecordTooLong	<b>Meaning:</b> The build and write SMF98 record service has determined that the SMF98 record exceeds 32756 bytes in length. <b>Action:</b> Verify that the number of records for the aggregate buckets and the exception bucket and the write lengths for the buckets are correct. If the number of records and write lengths are correct, consider reducing the amount of data recorded for each aggregate bucket record and/or exception bucket record.

Return code (hex)	Reason code (hex)	Meaning and action
8 WicSrv_Rc_UserError	80B WicSrv_Rsn_ BadCustomJobSizeEntries	<b>Meaning:</b> The caller of the build and write SMF98 record service specified WicWR_CustomJobSizeEntries greater than WicWR_#CustomJobSizeEntriesMax. The SMF 98.x record is not written.  <b>Action:</b> Set WicWR_CustomJobSizeEntries to a value less than or equal to WicWR_#CustomJobSizeEntriesMax.
8 WicSrv_Rc_UserError	80C WicSrv_Rsn_ ZiipAggBucExcUniqueByJobSize	<b>Meaning:</b> The caller of the build and write SMF98 record service provided zIIP aggregate bucket(s) when an exception job's activities with different job sizes are unique. The SMF 98.x record is not written.  <b>Action:</b> Remove zIIP aggregate job bucket(s) or set WicWR_FlagExcJobsUniqueByJobSize off.
8 WicSrv_Rc_UserError	80D WicSrv_Rsn_ ZiipExcBucExcUniqueByJobSize	<b>Meaning:</b> The caller of the build and write SMF98 record service provided zIIP exceptional bucket(s) when an exception job's activities with different job sizes are unique. The SMF 98.x record is not written.  <b>Action:</b> Remove zIIP exception job bucket(s) or set WicWR_FlagExcJobsUniqueByJobSize off.
C WicSrv_Rc_EnvError	C01 WicSrv_Rsn_ UnableToWriteSMF98Record	<b>Meaning:</b> The build and write SMF98 record service was unable to write the SMF type 98 record.  <b>Action:</b> This error is issued when the return code from the SMFEWMT macro invocation is non-zero. Refer to the high half (bits 0-15) of this reason code for the return code issued by the SMFEWMT macro. For information and corrective actions for the SMFEWMT return codes, see <a href="#">SMFEWMT — Writing SMF records in z/OS MVS System Management Facilities (SMF)</a> .

## Aggregate buckets

A WIC exploiter registers a WIC exit routine through the IFAWIC REQUEST=REGISTER service. The WIC exploiter receives an instrumentation buffer for each registered address space. Each address space instruments activities (for each CPU type or across all CPU types) into the buffer during each 5-second interval. For each address space buffer, the exploiter exit routine aggregates and summarizes activities from all address spaces that the exploiter registered.

For the purposes of WIC, a job represents an address space.

WIC uses *aggregate buckets* to summarize activities across jobs. The exploiter exit routine aggregates job activities into the aggregate buckets according to the following activity and job attributes:

### CPU type

An activity attribute that can be CP or zIIP. A CP aggregate activity can contain data for only CP CPUs (when a zIIP aggregate bucket exists) or all CPUs (when no zIIP aggregate bucket exists). A zIIP aggregate activity (in the zIIP aggregate bucket) contains data for only zIIP CPUs.

### Job priority

A job attribute that can be critical, high, low, or discretionary, based on the job's importance classification in WLM policy.

### Job size

Usually, a job attribute that can be large, medium, small, or tiny, based on the job's CPU use relative to all work by CPU type.

An exploiter can replace job size entries and attributes with custom exploiter-defined job entries and attributes (via WicWR\_CustomJobSizeEntries, mapped by IFAWICCB).

An aggregate bucket contains simple activities, such as counts, totals, and sizes, and compound activities, such as averages.

- A simple activity is instrumented during a five-second interval. Examples of simple activities include: number of transactions, total CPU time, and total response time.

- A compound activity is a combination of multiple simple activities. Compound activities are calculated at the end of the five-second interval. Examples of compound activities include: average CPU time per transaction and average response time per transaction.

An aggregate bucket entry is defined as follows:

#### Contributors

A required 2-byte area that contains the number of contributors or address spaces (jobs) that contribute to the given bucket entry.

#### Data

A WIC exploiter-defined area that contains recorded and internal activities.

- *Recorded activities* include simple and compound activities that will be written in the SMF type 98 record. By design, recorded activities must be grouped together before internal activities.
- *Internal activities* include simple and compound activities. The data contained in the internal activities will not be written in the SMF type 98 record. By design, internal activities must be grouped together after recorded activities. Internal activities are those that are collected to assist in building the recorded activities.

At least one aggregate bucket must be provided to the build and write SMF98 record service, but no more than two aggregate buckets can be provided.

The aggregate bucket is a table of entries where each entry represents an instance of an aggregate bucket. Aggregate bucket types include the following instances:

- *Specific instances (A)*, which summarize activity across jobs by CPU type, job priority, and job size. For example, CPU type: CP, job priority: high, and job size: small.
- *General instances (B)*, which summarize activity across multiple specific instances by CPU type and job priority. For example, CPU type: CP, job priority: high, job size: all.
- *Overall instances (C)*, which summarize activity across all the general instances by CPU type. There is only one overall instance per supported CPU type. For example, CPU type: CP, job priority: all, job size: all.

**Note:** All bucket combinations must exist in the data area and be provided to the build and write SMF 98 record service, but not all data will be written into the SMF 98 record output.

- Aggregate buckets with 0 contributors are not included in the SMF output.
- Aggregate buckets with a CPU type of CP or zIIP, a job priority of ALL, and a job size other than ALL (that is, large, medium, small, or tiny) are not included in the SMF output.

Table 4 on page 33 illustrates the specific, general, and overall bucket instances.

Table 4. Overview of specific (A), general (B), and overall (C) aggregate bucket instances					
Job size <sup>1</sup>	Job priority				
	All	Critical	High	Low	Discretionary
All	C	B	B	B	B
Large	-	A	A	A	A
Medium	-	A	A	A	A
Small	-	A	A	A	A
Tiny	-	A	A	A	A
<b>Notes:</b> 1. Job size entries (1–4) and attributes (Large–Tiny) can be replaced with custom, exploiter-defined job entries (1 to WicWR_CustomJobSizeEntries) and attributes.					

Bucket instances must be defined in ascending order, first by CPU type, then by job priority (starting with ALL), and by job size (starting with ALL). The build and write SMF98 record service requires the aggregate bucket instances to be ordered first by CPU type, then by job priority, and finally by job size, as shown in Table 5 on page 34. A job's priority and size are returned by the get next buffer service.

The aggregate bucket is mapped by WicWR\_AggBucket (in IFAWICCB). Table 5 on page 34 shows an example that depicts an array with three indices for CPU type, job priority and job size. In this example, the exploiter provides the data in the exploiter area for each CPU type, job priority, and job size combination including the number of contributors and the data, both recorded and internal.

Table 5. Aggregate bucket (supplied by exploiter; exploiter area mapped by WicWR_AggBucket)					
Indices (table entry)			Exploiter area		
			Contributors	Data	
CPU type	Job priority	Job size <sup>2</sup>		Recorded	Internal
CP (0)	ALL (0)	ALL (0)	7	Exploiter data	Exploiter data
CP (0)	ALL (0)	LARGE (1)	0	Null data <sup>1</sup>	Null data <sup>1</sup>
CP (0)	ALL (0)	MEDIUM (2)	0	Null data <sup>1</sup>	Null data <sup>1</sup>
CP (0)	ALL (0)	SMALL(3)	0	Null data <sup>1</sup>	Null data <sup>1</sup>
CP (0)	ALL (0)	TINY (4)	0	Null data <sup>1</sup>	Null data <sup>1</sup>
CP (0)	CRIT (1)	ALL (0)	3	Exploiter data	Exploiter data
CP (0)	CRIT (1)	LARGE (1)	2	Exploiter data	Exploiter data
CP (0)	CRIT (1)	MEDIUM (2)	1	Exploiter data	Exploiter data
CP (0)	CRIT (1)	SMALL (3)	0	Null data	Null data
CP (0)	CRIT (1)	TINY(4)	0	Null data	Null data
CP (0)	HIGH (2)	ALL(0)	2	Exploiter data	Exploiter data
CP (0)	HIGH (2)	LARGE (1)	2	Exploiter data	Exploiter data
CP (0)	HIGH (2)	MEDIUM (2)	0	Null data	Null data
CP (0)	HIGH (2)	SMALL (3)	0	Null data	Null data
CP (0)	HIGH (2)	TINY (4)	0	Null data	Null data
CP (0)	LOW (3)	ALL (0)	2	Exploiter data	Exploiter data
CP (0)	LOW (3)	LARGE (1)	0	Null data	Null data
CP (0)	LOW (3)	MEDIUM (2)	2	Exploiter data	Exploiter data
CP (0)	LOW (3)	SMALL (3)	0	Null data	Null data
CP (0)	LOW (3)	TINY (4)	0	Null data	Null data
CP (0)	DISC (4)	ALL (0)	0	Null data	Null data
CP (0)	DISC (4)	LARGE (1)	0	Null data	Null data
CP (0)	DISC (4)	MEDIUM (2)	0	Null data	Null data
CP (0)	DISC (4)	SMALL (3)	0	Null data	Null data
CP (0)	DISC (4)	TINY (4)	0	Null data	Null data
zIIP (1)	ALL (0)	ALL (0)	2	Exploiter data	Exploiter data
zIIP (1)	ALL (0)	LARGE (1)	0	Null data <sup>1</sup>	Null data <sup>1</sup>
zIIP (1)	ALL (0)	MEDIUM (2)	0	Null data <sup>1</sup>	Null data <sup>1</sup>
zIIP (1)	ALL (0)	SMALL(3)	0	Null data <sup>1</sup>	Null data <sup>1</sup>

Table 5. Aggregate bucket (supplied by exploiter; exploiter area mapped by WicWR\_AggBucket) (continued)

Indices (table entry)			Exploiter area		
			Contributors	Data	
CPU type	Job priority	Job size <sup>2</sup>		Recorded	Internal
zIIP (1)	ALL (0)	TINY (4)	0	Null data <sup>1</sup>	Null data <sup>1</sup>
zIIP (1)	CRIT (1)	ALL (0)	0	Null data	Null data
zIIP (1)	CRIT (1)	LARGE (1)	0	Null data	Null data
zIIP (1)	CRIT (1)	MEDIUM (2)	0	Null data	Null data
zIIP (1)	CRIT (1)	SMALL (3)	0	Null data	Null data
zIIP (1)	CRIT (1)	TINY(4)	0	Null data	Null data
zIIP (1)	HIGH (2)	ALL(0)	2	Exploiter data	Exploiter data
zIIP (1)	HIGH (2)	LARGE (1)	0	Null data	Null data
zIIP (1)	HIGH (2)	MEDIUM (2)	2	Exploiter data	Exploiter data
zIIP (1)	HIGH (2)	SMALL (3)	0	Null data	Null data
zIIP (1)	HIGH (2)	TINY (4)	0	Null data	Null data
zIIP (1)	LOW (3)	ALL (0)	0	Null data	Null data
zIIP (1)	LOW (3)	LARGE (1)	0	Null data	Null data
zIIP (1)	LOW (3)	MEDIUM (2)	0	Null data	Null data
zIIP (1)	LOW (3)	SMALL (3)	0	Null data	Null data
zIIP (1)	LOW (3)	TINY (4)	0	Null data	Null data
zIIP (1)	DISC (4)	ALL (0)	0	Null data	Null data
zIIP (1)	DISC (4)	LARGE (1)	0	Null data	Null data
zIIP (1)	DISC (4)	MEDIUM (2)	0	Null data	Null data
zIIP (1)	DISC (4)	SMALL (3)	0	Null data	Null data
zIIP (1)	DISC (4)	TINY (4)	0	Null data	Null data

**Notes:**

1. WIC does not produce records for job priority ALL, job size-specific indices.
2. Job size entries (1–4) and attributes (Large–Tiny) can be replaced with custom, exploiter-defined job entries (1 to WicWR\_CustomJobSizeEntries) and attributes.

The build and write SMF98 record service builds the aggregate data in the SMF type 98 record, as shown in Table 6 on page 35. Notice that only data where the contributor value is non-zero is written to the SMF type 98 record.

Table 6. SMF type 98 subtype x aggregate bucket record (mapped by SMF98AggBucket in IHAHR098)

Populated by build and write SMF98 record service			As supplied by the exploiter	
CPU type	Job priority	Job size <sup>1</sup>	Contributors	Recorded data
CP (0)	ALL (-1)	ALL (-1)	7	Exploiter data
CP (0)	CRIT (1)	ALL (-1)	3	Exploiter data
CP (0)	CRIT (1)	LARGE (1)	2	Exploiter data
CP (0)	CRIT (1)	MEDIUM (2)	1	Exploiter data

Table 6. SMF type 98 subtype *x* aggregate bucket record (mapped by SMF98AggBucket in IHAHR098) (continued)

Populated by build and write SMF98 record service			As supplied by the exploiter	
CPU type	Job priority	Job size <sup>1</sup>	Contributors	Recorded data
CP (0)	HIGH (2)	ALL (-1)	2	Exploiter data
CP (0)	HIGH (2)	LARGE (1)	2	Exploiter data
CP (0)	LOW (3)	ALL (-1)	2	Exploiter data
CP (0)	LOW (3)	MEDIUM (2)	2	Exploiter data
zIIP (1)	ALL (-1)	ALL (-1)	2	Exploiter data
zIIP (1)	HIGH (2)	ALL (-1)	2	Exploiter data
zIIP (1)	HIGH (2)	MEDIUM (2)	2	Exploiter data

**Notes:**

1. Job size entries (1–4) and attributes (Large–Tiny) can be replaced with custom, exploiter-defined job entries (1 to WicWR\_CustomJobSizeEntries) and attributes.

## Exception bucket

The exception bucket contains data about jobs with exceptional activity. Each bucket instance (every CPU type, job priority, and job size combination) activity can have a most exceptional job name and its corresponding job activity. The exploiter can determine what characteristics make a job exceptional such as the maximum, minimum, or top average for an activity. The exploiter can also choose how many exceptional activities to include and what to provide for each job.

An exception bucket entry is mapped by WicWR\_ExcBucket (in IFAWICCB) and is defined as follows:

### Header area

A 10-byte area that contains the address space identifier (2 bytes) and the job name space identification (8 bytes).

### Data

A WIC exploiter-defined area that contains recorded and internal activities.

- *Recorded activities* include simple and compound activities that will be written to the SMF type 98 record. By design, recorded activities must be grouped together before internal activities.
- *Internal activities* include simple and compound activities. The data contained in the internal activities will not be written to the SMF type 98 record. By design, internal activities must be grouped together after recorded activities.

Only one exception bucket must be supplied to the build and write SMF 98 record service. The exception bucket can be thought of as a table of entries, where each entry represents an instance of an exception bucket.

The exception bucket is organized like the aggregate bucket, but contains an extra dimension—the activity index. Each exception bucket entry is ordered by CPU type, job priority, job size, and activity index. The activity index is determined by the exploiter and starts at 1. An activity index identifies the activity for the given exception bucket instance.

Exception bucket entries must be defined in ascending order, first by CPU type, then by job priority starting with ALL, job size starting with ALL, and activity index. [Table 7 on page 37](#) shows an exception bucket with *n* activities.

The build and write SMF98 record service requires the exploiter area to be supplied along with the number of activities on which the exploiter is reporting. The service expects that the exploiter area data is in the order shown in [Table 7 on page 37](#).



Table 7. Exception bucket (supplied by exploiter; exploiter area mapped by WicWR\_ExcBucket)

Indices (table entry)				Exploiter area			
				ASID	Job name	Data	
CPU type (0 - 1)	Job priority (0 - 4)	Job size <sup>2</sup> (0 - 4)	Activity index (1 - n)			Recorded	Internal
CP (0)	ALL (0)	ALL (0)	1	X'40'	JOB1	Exploiter data	Exploiter data
CP (0)	ALL (0)	ALL (0)	:	:	:	:	:
CP (0)	ALL (0)	ALL (0)	n	X'42'	JOB3	Exploiter data	Exploiter data
CP (0)	ALL (0)	LARGE (1)	1	X'00'	-	Null data <sup>1</sup>	Null data <sup>1</sup>
:	:	:	:	:	:	:	:
CP (0)	ALL (0)	TINY (4)	n	X'00'	-	Null data <sup>1</sup>	Null data <sup>1</sup>
CP (0)	CRIT (1)	ALL (0)	1	X'40'	JOB1	Exploiter data	Exploiter data
CP (0)	CRIT (1)	ALL (0)	:	:	:	:	:
CP (0)	CRIT (1)	ALL (0)	n	X'42'	JOB3	Exploiter data	Exploiter data
CP (0)	CRIT (1)	LARGE (1)	1	X'40'	JOB1	Exploiter data	Exploiter data
CP (0)	CRIT (1)	LARGE (1)	:	:	:	:	:
CP (0)	CRIT (1)	LARGE (1)	n	X'41'	JOB2	Exploiter data	Exploiter data
CP (0)	CRIT (1)	MEDIUM (2)	1	X'42'	JOB3	Exploiter data	Exploiter data
CP (0)	CRIT (1)	MEDIUM (2)	:	:	:	:	:
CP (0)	CRIT (1)	MEDIUM (2)	n	X'42'	JOB3	Exploiter data	Exploiter data
CP (0)	CRIT (1)	SMALL (3)	1	X'00'	-	Null data	Null data
:	:	:	:	:	:	:	:
CP (0)	CRIT (1)	TINY (4)	n	X'00'	-	Null data	Null data
CP (0)	HIGH (2)	ALL (0)	1	X'43'	JOB4	Exploiter data	Exploiter data
CP (0)	HIGH (2)	ALL (0)	:	:	:	:	:
CP (0)	HIGH (2)	ALL (0)	n	X'43'	JOB4	Exploiter data	Exploiter data
CP (0)	HIGH (2)	LARGE (1)	1	X'43'	JOB4	Exploiter data	Exploiter data
CP (0)	HIGH (2)	LARGE (1)	:	:	:	:	:
CP (0)	HIGH (2)	LARGE (1)	n	X'43'	JOB4	Exploiter data	Exploiter data
CP (0)	HIGH (2)	MEDIUM (2)	1	X'00'	-	Null data	Null data
:	:	:	:	:	:	:	:
CP (0)	HIGH (2)	TINY (4)	n	X'00'	-	Null data	Null data
CP (0)	LOW (3)	ALL (0)	1	X'46'	JOB7	Exploiter data	Exploiter data
CP (0)	LOW (3)	ALL (0)	:	:	:	:	:
CP (0)	LOW (3)	ALL (0)	n	X'45'	JOB6	Exploiter data	Exploiter data
CP (0)	LOW (3)	LARGE (1)	1	X'00'	-	Null data	Null data
CP (0)	LOW (3)	LARGE (1)	:	:	:	:	:
CP (0)	LOW (3)	LARGE (1)	n	X'00'	-	Null data	Null data

Table 7. Exception bucket (supplied by exploiter; exploiter area mapped by WicWR\_ExcBucket) (continued)

Indices (table entry)				Exploiter area			
				ASID	Job name	Data	
CPU type (0 - 1)	Job priority (0 - 4)	Job size <sup>2</sup> (0 - 4)	Activity index (1 - n)			Recorded	Internal
CP (0)	LOW (3)	MEDIUM (2)	1	X'46'	JOB7	Exploiter data	Exploiter data
CP (0)	LOW (3)	MEDIUM (2)	:	:	:	:	:
CP (0)	LOW (3)	MEDIUM (2)	n	X'45'	JOB6	Exploiter data	Exploiter data
CP (0)	LOW (3)	SMALL (3)	1	X'00'	-	Null data	Null data
:	:	:	:	:	:	:	:
CP (0)	DISC (4)	TINY (4)	n	X'00'	-	Null data	Null data
zIIP (1)	ALL (0)	ALL (0)	1	X'47'	JOB8	Exploiter data	Exploiter data
zIIP (1)	ALL (0)	ALL (0)	:	:	:	:	:
zIIP (1)	ALL (0)	ALL (0)	n	X'48'	JOB9	Exploiter data	Exploiter data
zIIP (1)	ALL (0)	LARGE (1)	1	X'00'	-	Null data <sup>1</sup>	Null data <sup>1</sup>
:	:	:	:	:	:	:	:
zIIP (1)	ALL (0)	TINY (4)	n	X'00'	-	Null data <sup>1</sup>	Null data <sup>1</sup>
zIIP (1)	CRIT (1)	ALL (0)	1	X'00'	-	Null data	Null data
:	:	:	:	:	:	:	:
zIIP (1)	CRIT (1)	TINY (4)	n	X'00'	-	Null data	Null data
zIIP (1)	HIGH (2)	ALL (0)	1	X'47'	JOB8	Exploiter data	Exploiter data
zIIP (1)	HIGH (2)	ALL (0)	:	:	:	:	:
zIIP (1)	HIGH (2)	ALL (0)	n	X'48'	JOB9	Exploiter data	Exploiter data
zIIP (1)	HIGH (2)	LARGE (1)	1	X'00'	-	Null data	Null data
zIIP (1)	HIGH (2)	LARGE (1)	:	:	:	:	:
zIIP (1)	HIGH (2)	LARGE (1)	n	X'00'	-	Null data	Null data
zIIP (1)	HIGH (2)	MEDIUM (2)	1	X'47'	JOB8	Exploiter data	Exploiter data
zIIP (1)	HIGH (2)	MEDIUM (2)	:	:	:	:	:
zIIP (1)	HIGH (2)	MEDIUM (2)	n	X'48'	JOB9	Exploiter data	Exploiter data
zIIP (1)	HIGH (2)	SMALL (3)	1	X'00'	-	Null data	Null data
:	:	:	:	:	:	:	:
zIIP (1)	DISC (4)	TINY (4)	n	X'00'	-	Null data	Null data

**Notes:**

1. WIC does not produce records for job priority ALL, job size-specific indices.
2. Job size entries (1–4) and attributes (Large–Tiny) can be replaced with custom, exploiter-defined job entries (1 to WicWR\_CustomJobSizeEntries) and attributes.

The build and write SMF98 record service adds the exception data to the SMF type 98 record as two data areas, a job index data record and a job list data record (mapped by SMF98JobIdx and SMF98JobList in the IHAHR098 data area).

Job index data identifies the most exceptional job for defined activities for a given CPU type, job priority, and job size. The most exceptional job activity is identified by an index value that can be used to calculate an offset within the job list data area that identifies the address space identification (ASID) and job name of the maximum exploiter for the given activity. [Table 8 on page 39](#) shows an example of the job index data record.

<i>Table 8. Example of SMF type 98 subtype x job index data record (mapped by SMF98JobIdx in IHAHR098)</i>						
Populated by the build and write SMF98 record service			Maximum activity 1	Maximum activity 2	...	Maximum activity n
CPU type	Job priority	Job size <sup>1</sup>				
CP (0)	ALL (-1)	ALL (-1)	1	2	...	3
CP (0)	CRIT (1)	ALL (-1)	1	2	...	3
CP (0)	CRIT (1)	LARGE (1)	1	2	...	1
CP (0)	CRIT (1)	MEDIUM (2)	3	3	...	3
CP (0)	HIGH (2)	ALL (-1)	4	5	...	4
CP (0)	HIGH (2)	LARGE (1)	4	5	...	4
CP (0)	LOW (3)	ALL (-1)	6	7	...	7
CP (0)	LOW (3)	MEDIUM (2)	6	7		7
zIIP (1)	ALL (-1)	ALL (-1)	8	8	...	9
zIIP (1)	HIGH (2)	ALL (-1)	8	8	...	9
zIIP (1)	HIGH (2)	MEDIUM (2)	8	8		9
<b>Notes:</b> 1. Job size entries (1–4) and attributes (Large–Tiny) can be replaced with custom, exploiter-defined job entries (1 to WicWR_CustomJobSizeEntries) and attributes.						

The job list data area is a list of entries with unique ASIDs and job names with corresponding job activities. [Table 9 on page 39](#) shows an example of the job list data record.

WicWR\_FlagExcJobsUniqueByJobSize, mapped by IFAWICCB, controls whether multiple instances of the same job are compressed into a single instance always (off, default) or per job size (on).

<i>Table 9. Example of SMF type 98 subtype x job list data record</i>						
Job list index	SMF98.x job list data record					
	ASID	Job name	Exploiter record area			
			Activity 1	Activity 2	...	Activity n
1	X'40'	JOB1	Exploiter record area			
2	X'41'	JOB2	Exploiter record area			
3	X'42'	JOB3	Exploiter record area			
4	X'43'	JOB4	Exploiter record area			
5	X'44'	JOB5	Exploiter record area			
6	X'45'	JOB6	Exploiter record area			
7	X'46'	JOB7	Exploiter record area			
8	X'47'	JOB8	Exploiter record area			
9	X'48'	JOB9	Exploiter record area			



---

## Chapter 4. Reference for system programmers

The following topics are of particular interest to system programmers:

- [“SMFPRMxx \(system management facilities \(SMF\) parameters\)” on page 41](#)
- [“Displaying hardware event data collection status \(HIS\)” on page 45](#)
- [“Displaying SMF data \(SMF\)” on page 47](#)
- [“Starting, configuring, and stopping HIS data collection” on page 48](#)
- [“HIS and IFA system messages” on page 60](#)

---

### SMFPRMxx (system management facilities (SMF) parameters)

The SMFPRMxx member allows you to control how system management facilities (SMF) works at your installation. You can use SMFPRMxx parameters to accomplish the following tasks:

- Identify the system on which SMF is active.
- Specify global values for interval recording and synchronization that SMF, RMF, and other requesters can use to schedule the execution of their interval functions.
- Specify the data sets or log streams to be used for SMF recording.
- Allow compression of SMF data before recording to log streams.
- Specify the system identifier to be used in all SMF records.
- Select the SMF record types and subtypes SMF is to generate.
- Allow the operator to change the SMF parameters that are established at IPL.
- Specify the job wait time limit.
- Specify whether SMF is to invoke installation-supplied SMF exit routines.
- Specify whether the SMF dump program is to attempt to recover from abends.
- Specify the system response when SMF has used all of the buffered storage in its address space and is recording on data sets.
- Specify the system response when the last SMF data set is filled and no other data sets are available for use.
- Specify the amount of real time that SMF allows data to remain in an SMF buffer before it is written to a recording data set or a log stream.
- Specify the installation default MEMLIMIT.
- Specify whether only registration data and not usage data is to be recorded when using the IFAUSAGE macro.
- Specify valid user exists for the IFASMFDP and IFASMFDL programs.
- Specify whether to suppress empty execute channel program (EXCP) using the EMPTYEXCPSEC parameter.
- Specify whether SMF record flood support is active and the filter for the SMF record flood.
- Specify whether SMF is to generate digital signatures for records recorded to log streams.
- Specify whether SMF is to record records to in-memory resources for real-time analysis.
- Specify whether SMF is to record high-frequency throughput statistics.
- Specify whether to enable IBM z/OS Workload Interaction Correlator.

**Note:** If you specify that SMF is to record on data sets, the SMF data sets must be cataloged on DASD. If there are no data sets for SMF to use, SMF buffers data until you specify a data set for SMF to use. If SMF runs out of buffers, there might be a loss of data.

You can specify SMF parameters in several ways:

- Before the first IPL of a newly generated system by creating an SMFPRMxx parmlib member.
- At each initialization of SMF by entering the parameters at the console.
- During SMF execution, by using the SET SMF command to specify a different SMFPRMxx parmlib member or by using the SETSMF command to replace one or more previously defined SMF parameters.

See *z/OS MVS System Commands* for more information about the commands that are used with SMF. For information about setting up and using SMF, see [z/OS MVS System Management Facilities \(SMF\)](#).

## Syntax format of SMFPRMxx

INTVAL(*mm*)

SYNVAL(*mm*)

ACTIVE | NOACTIVE

RECORDING(DATASET | LOGSTREAM)

```
DEFAULTLSNAME(logstreamname
               [,NOBUFFS(HALT|MSG)]
               [,BUFUSEWARN(nn)]
               [,DSPSIZMAX(nnnnM|nG)]
               [,NOCOMPRESS|COMPRESS[(PERMFIX(nnnnM))]]
               [,NORECSIGN|RECSIGN(HASH(SHA1|SHA256|SHA384|SHA512),
                                   SIGNATURE(RSA|ECDSA),
                                   TOKENNAME(tokenname))])
               [,NOARECSIGN|ARECSIGN(HASH(SHA512),
                                   SIGNATURE(LI2),
                                   TOKENNAME(tokenname))])
```

```
LSNAME(log streamname,TYPE{(aa,bb)|(aa,bb:zz)|(aa,bb:zz,...)}
        [,NOBUFFS(HALT|MSG)]
        [,BUFUSEWARN(nn)]
        [,DSPSIZMAX(nnnnM|nG)]
        [,NOCOMPRESS|COMPRESS[(PERMFIX(nnnnM))]]
        [,NORECSIGN|RECSIGN(HASH(SHA1|SHA256|SHA384|SHA512),
                            SIGNATURE(RSA|ECDSA),
                            TOKENNAME(tokenname))])
        [,NOARECSIGN|ARECSIGN(HASH(SHA512),
                            SIGNATURE(LI2),
                            TOKENNAME(tokenname))])
```

```
INMEM(rname,RESSIZMAX({nnnnM|nG}),{TYPE({aa,bb|aa,bb:zz|aa,bb:zz,...})|
                                NOTYPE({aa,bb|aa,bb:zz|aa,bb:zz,...})}
```

DSNAME(dataset)

LISTDSN | NOLISTDSN

```
SID { (xxxx) }
    { (xxxx,SYSNAME(sysname)) }
    { (xxxx,ser#[,ser#...]) }
    { (xxxx,COMBIN(ser#[,ser#...])) }
```

REC(PERM|ALL)

```
NORECSIGN | RECSIGN(HASH(SHA1|SHA256|SHA384|SHA512),
                     SIGNATURE(RSA|ECDSA),
                     TOKENNAME(tokenname))
NOARECSIGN | ARECSIGN(HASH(SHA512),
                     SIGNATURE(LI2),
                     TOKENNAME(tokenname))
```

MAXDORM(mmss) | NOMAXDORM

```
MEMLIMIT({NOLIMIT})
         {nnnnnM}
         {nnnnnG}
         {nnnnnT}
         {nnnnnP}
```

EMPTYEXCPSEC(NOSUPPRESS|SUPPRESS)

HFTSINTVL(*ss*)|NOHFTSINTVL

WIC | NOWIC

STATUS([*hhmmss*] ) | NOSTATUS  
[SMF[,SYNC|NOSYNC]]

JWT(*hhmm*)

SWT(*hhmm*)

TWT(*hhmm*)

DDCONS(YES|NO)

PROMPT(ALL,IPLR,LIST) | NOPROMPT

AUTHSETSMF | NOAUTHSETSMF

SYS([TYPE  
[,INTERVAL]  
[,EXITS  
[,DETAIL ]

TYPE {*aa,bb(cc)* }  
NOTYPE {*aa,bb:zz* }  
{*aa,dd(cc:yy),...*}  
{*aa,bb(cc,...)* }

NOINTERVAL | INTERVAL({*hhmmss* }  
{SMF[,SYNC|NOSYNC]} )

EXITS(*exit-name,exit-name,...*) | NOEXITS

NODETAIL | DETAIL

SUBPARM(*name(parameter)*)

SUBSYS(*name*,[TYPE  
[,INTERVAL]  
[,EXITS  
[,DETAIL ]

DUMPABND(RETRY|NORETRY)

NOBUFFS(MSG|HALT)

LASTDS(MSG|HALT)

MULCFUNC | NOMULCFUNC

BUFSIZMAX(*nnnnM*|1G)

BUFUSEWARN(*nn*)

SMFDLEXIT( {USER1(*exit-name[,exit-name,...]*) | NOUSER1},  
{USER2(IRRADU00[,*exit-name,...]*) | NOUSER2},  
{USER3(IRRADU86[,*exit-name,...]*) | NOUSER3},  
{USER4(*exit-name[,exit-name,...]*) | NOUSER4},  
{USER5(*exit-name[,exit-name,...]*) | NOUSER5} )

SMFDPEXIT( {USER1(*exit-name[,exit-name,...]*) | NOUSER1},  
{USER2(IRRADU00[,*exit-name,...]*) | NOUSER2},  
{USER3(IRRADU86[,*exit-name,...]*) | NOUSER3},  
{USER4(*exit-name[,exit-name,...]*) | NOUSER4},  
{USER5(*exit-name[,exit-name,...]*) | NOUSER5} )

FLOOD(ON|OFF)

FLOODPOL(TYPE({*aa,bb* }  
{*aa,bb:zz* }  
{*aa,bb:zz,...*}  
RECTHRESH(*xxxx*),  
INTVLTIME(*xxxx*),  
MAXHIGHINTS(*xxxx*),  
ENDINTVL(*xxxx*),  
ACTION(MSG|DROP)

```
)
MAXEVENTINTRECS(nn)
DSPSIZMAX(nnnnM|nG))
PERMFIX(nnnnM)
SMF30COUNT | NOSMF30COUNT
```

## Statements and parameters for SMFPRMxx

The following SMFPRMxx parameters apply to IBM z/OS Workload Interaction Correlator:

### **HFTSINTVL(ss) | NOHFTSINTVL**

When the NOWIC parameter is specified or allowed to default, the HFTSINTVL parameter specifies the time interval, in seconds, for writing SMF type 98 records, which record high-frequency throughput statistics (HFTS).

The supported values are 5, 10, 15, 20, 30, and 60 seconds. For instance, HFTSINTVL(20) indicates that SMF type 98 records are to be written every 20 seconds, starting at the top of the minute and each 20-second multiple that follows.. IBM recommends setting HFTSINTVL(5) to write SMF type 98 records every 5 seconds, starting at the top of the minute and each 5-second multiple that follows.

When you specify any HFTSINTVL value, SMF type 98 records are collected every 5 seconds starting at the top of the minute and each 5-second multiple for one minute at 0, 15, 30, and 45 minutes past each hour. For all other minutes during each hour, SMF type 98 records are written at the interval specified by the HFTSINTVL parameter.

The NOHFTSINTVL parameter disables the HFTS interval and prevents the collection of SMF type 98 records. This is the default value.

SMF collects type 98 records only when both the HFTSINTVL parameter and the TYPE(98) parameter are specified. Specifying NOHFTSINTVL along with TYPE(98) allows you to disable the collection of type 98 records without changing the TYPE setting.

When the WIC parameter is specified, the HFTSINTVL value is ignored.

**Default:** NOHFTSINTVL

**Values:** 5, 10, 15, 20, 30, or 60 seconds

### **WIC | NOWIC**

When you specify the WIC parameter, the IBM z/OS Workload Interaction Correlator (WIC) feature is enabled on the system. z/OS components, middleware, and applications can use the IFAWIC service to register for instrumentation, request an instrumentation buffer for the job, and provide an exit routine.

The exit routine will be called to summarize instrumentation data and produce SMF type 98 records for subtype 2 or greater. This exit will be called every 5 seconds, starting at the top of the minute and each 5-second multiple that follows, when the following conditions are true:

- The WorkloadIntCorr feature is enabled in the product enablement policy (IFAPRDxx).
- SETSMF or SMFPRMxx has specified WIC.
- SMF is collecting type 98 subtype 2 or greater records.

SMF type 98 subtype 1 (supervisor) records are not generated through IBM z/OS Workload Interaction Correlator and will be generated when the following conditions are true:

- SETSMF or SMFPRMxx has specified WIC or HFTSINTVL(5) or greater.
- SMF is collecting type 98 subtype 1 records.

When you specify the WIC parameter, the HFTSINTVL parameter is ignored, and SMF type 98 subtype 1 records are generated every 5 seconds.



When you specify the NOWIC parameter, IBM z/OS Workload Interaction Correlator (WIC) is disabled on the system. The IFAWIC REQUEST=REGISTER service will fail for applications that call it. IFAWIC exploiters that are already registered for instrumentation will no longer have their exit routines called. When NOWIC is specified, HFTSINTVL is honored.

**Default:** NOWIC

## Displaying hardware event data collection status (HIS)

Use the DISPLAY HIS command to display the results of the latest HIS Profiler run initiated by an F *hisproc*, BEGIN command. Hardware event data collection is performed by hardware instrumentation services (HIS), which collects hardware event data for processors in SMF records type 113 and some of the z/OS UNIX System Services output files.

The DISPLAY HIS command is also used to display statistics on the exploiters of the HISSERV service. The HISSERV service allows authorized programs to collect hardware instrumentation data. The amount of time each exploiter takes to collect the data is recorded by the service, and displayed here to aid in problem determination should a performance degradation occur.

Note that the results displayed by the D HIS command in system message HIS015I message will only be as current as the last time you initiated a hardware event data collection run with the F *hisproc*, BEGIN command. The displaying of statistical information, as well as any service parameters, are displayed as long as the HISSERV service is active.

The DISPLAY HIS command is also used to display HIS WIC collection status. WIC collection is altered by the F *hisproc*, WIC command. D HIS, WIC can be used to display just the WIC collection information.

```
D HIS[,WIC][,L=a|name|name-a]
```

The parameters are:

### HIS

The system displays HIS information in message HIS015I. For more information about HIS015I, see [z/OS MVS System Messages, Vol 6 \(GOS-IEA\)](#).

### WIC

This option limits the displayed information to just the information about HIS use of IBM z/OS Workload Interaction Correlator.

### L=a|name|name-a

This operand specifies the display area (a), console name (name), or both (name-a) where the output is to be displayed.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter command.

You can start a run of hardware event data collection using the MODIFY *hisproc*, BEGIN command. See "Starting, configuring, and stopping HIS data collection."

## Examples

**Example 1:** If you issue the D HIS command to display the results of the latest hardware event data collection run while the collection is still in progress, the data displayed might be incomplete. The following example shows output from a D HIS command issued while data collection is still running. SYSHIS20071108.184129 is the file name prefix for all the output files for the data collection run. The PATH= /user is the USS directory where all the output files are created. The COMPLETION STATUS indicates if the instrumentation run was successful or not. LOST SAMPLES shows the total count of lost samples on all the processors if sampling is active.

The HIS profiler (the collection started by F *hisproc*, BEGIN) is the only HISSERV exploiter. It queries for counters data every 15 minutes, and takes almost .2 seconds to process each sampling data event callback.

The data collection was not complete when the D HIS command was issued, so the system had not yet converted the MAPJOB job names to MAPASID values at the time when DISPLAY command was issued, which results in the MAPASID values not being displayed in the output:

```
HIS015I 10.19.12 DISPLAY HIS 241
HIS      001E ACTIVE
COMMAND: MODIFY HIS,B,TT='Sampling',BUF=25,PATH='/user',MJOB=(OMVS,J*,GRS)
START TIME: 2014/10/27 10:19:08
END TIME:   ----/--/--  --:--:--
COMPLETION STATUS: -----
FILE PREFIX: SYSHIS20141027.101908.
LOST SAMPLES: 0
COUNTER VERSION NUMBER 1: 1   COUNTER VERSION NUMBER 2: 3
COMMAND PARAMETER VALUES USED:
  TITLE= Sampling
  PATH=  /user
  DURATION= 10 (MINUTES)
  DATALOSS= IGNORE
  STATECHANGE= SAVE
  SMP= YES
  BUFCNT= 25 (PAGES/PROCESSOR)
  SAMPTYPE= DEFAULT
  SAMPFREQ= DEFAULT
  CNT= YES
  COUNTER SET= BASIC,PROBLEM-STATE
  CNTINTVAL= 15 (MINUTES)
  CNTFILE= YES
  MAP= YES
  MAPJOB= OMVS      J*      GRS
HISSERV STATUS: ACTIVE
EVENT
  AUTHORIZED= BASIC,PROBLEM-STATE,CRYPTO-ACTIVITY,EXTENDED,ZOS
  ENABLED= BASIC,PROBLEM-STATE
SAMPLE
  AUTHORIZED= BASIC
  ENABLED= BASIC
  BUFCNT= 25 (PAGES/PROCESSOR)
  SAMPFREQ= 800000 (SAMPLES/MINUTE)
PROFILER
  NAME      START      QUERY      SAMPLE  S  F
  HISPROF   2014/10/27 10:19:08  00:00:00.000017  -----  S
HIS WIC STATUS: INACTIVE
INMEM= *NONE*
WICPATH= *NONE*
```

**Example 2:** In the second example, the DISPLAY command was issued after a hardware event data collection run that specified MAPONLY as well as MAPASID and MAPJOBS. There are no HISSERV exploiters at the time the DISPLAY command was issued.

```
HIS015I 10.40.43 DISPLAY HIS 381
HIS      0030 IDLE
COMMAND: MODIFY HIS,B,TT='map-only',MAPONLY,MAS=(1,EC),MJOB=(J*,GRS)
START TIME: 2014/10/27 10:40:38
END TIME:   ----/--/--  --:--:--
COMPLETION STATUS: -----
FILE PREFIX: SYSHIS20141027.104038.
COMMAND PARAMETER VALUES USED:
  TITLE= map-only
  PATH=  .
  DURATION= NOLIMIT
  DATALOSS= IGNORE
  STATECHANGE= SAVE
  SMP= NO
  CNT= NO
  MAP= YES
  MAPASID= 1   EC   12   14   1D   2A   7
  MAPJOB= J*      GRS
HISSERV STATUS: INACTIVE
EVENT
  AUTHORIZED= BASIC,PROBLEM-STATE,CRYPTO-ACTIVITY,EXTENDED,ZOS
  ENABLED= NONE
SAMPLE
  AUTHORIZED= BASIC
  ENABLED= NONE
  BUFCNT= 9 (PAGES/PROCESSOR)
  SAMPFREQ= 800000 (SAMPLES/MINUTE)
HIS WIC STATUS: INACTIVE
```

```
INMEM= *NONE*
WICPATH= *NONE*
```

**Example 3:** The following D HIS command output shows a counters-only run, to collect only event counter set data, specifying all the counter sets. Note that the DISPLAY command was issued while the data collection was still ongoing.

The HIS Profiler (the collection started by F hisproc,BEGIN) and BADEXP are currently exploiting the HISSERV service. Notice how the BADEXP exploiter queries for counter data every .5 seconds, and handles sample data in an untimely manner. If hardware event data collection is degrading system performance, BADEXP might be the reason.

```
HIS015I 10.42.54 DISPLAY HIS 394
HIS      0030 ACTIVE
COMMAND: MODIFY HIS,B,CNTSET=ALL,TITLE='Counters only',CNTONLY
START TIME: 2014/10/27 10:42:50
END TIME:   ----/--/--  --:--:--
COMPLETION STATUS: -----
FILE PREFIX: SYSHIS20141027.104250.
COUNTER VERSION NUMBER 1: 1  COUNTER VERSION NUMBER 2: 3
COMMAND PARAMETER VALUES USED:
  TITLE= Counters only
  PATH= .
  DURATION= NOLIMIT
  DATALOSS= IGNORE
  STATECHANGE= SAVE
  SMP= NO
  CNT= YES
  COUNTER SET= BASIC,PROBLEM-STATE,CRYPTO-ACTIVITY,EXTENDED
  CNTINTVAL= 15 (MINUTES)
  CNTFILE= YES
  MAP= NO
HISSERV STATUS: ACTIVE
EVENT
  AUTHORIZED= BASIC,PROBLEM-STATE,CRYPTO-ACTIVITY,EXTENDED,ZOS
  ENABLED= BASIC,PROBLEM-STATE,CRYPTO-ACTIVITY,EXTENDED
SAMPLE
  AUTHORIZED= BASIC
  ENABLED= NONE
  BUFCNT= 9 (PAGES/PROCESSOR)
  SAMPFREQ= 800000 (SAMPLES/MINUTE)
PROFILER
  NAME      START      QUERY      SAMPLE  S F
  HISPROF   2014/10/27 10:42:50  00:00:00.000016  -----
  BADEXP    2014/10/27 06:00:00  00:00:00:538521  +999999 S
HIS WIC STATUS: INACTIVE
INMEM= *NONE*
WICPATH= *NONE*
```

**Example 4:** The following D HIS,WIC command output shows that a HIS is reading from an INMEM and writing to a WICPATH following a F hisproc,WIC command.

```
HIS015I 12.19.40 DISPLAY HIS 929
HIS WIC STATUS: ACTIVE
INMEM= IFASMF.WIC
WICPATH= /wic
```

For more information about HIS015I, see [z/OS MVS System Messages, Vol 6 \(GOS-IEA\)](#).

## Displaying SMF data (SMF)

Use the DISPLAY SMF command to display System Management Facilities (SMF) data.

### D SMF

```
D SMF[, {S|O|M|WIC}] [, L={a|name|name-a}]
```

**SMF**

Indicates that the status of SMF data sets or the SMF options in effect are to be displayed (message IEE967I).

**S**

Directs the system to display the names and status of the SMF data sets.

**O**

Directs the system to display the current SMF options.

**M**

Directs the system to display information about in-memory resources and connections to them. In response, the system issues the IFA714I message.

**WIC**

Direct the system to display SMF information about IBM z/OS Workload Interaction Correlator.

**L=*a*, *name*, or *name-a***

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

## Starting, configuring, and stopping HIS data collection

---

The MODIFY hisproc command can be used to:

- BEGIN and END hardware event data collection including hardware counts, hardware samples, and MAP files with address space information.
- Change the characteristics of an HIS profiler (exploiter of HISSERV service) with the SERVICE subcommand.
- Read SMF records from a SMF in-memory resource and write them to file system for the IBM z/OS Workload Interaction Correlator data generation using the WIC subcommand.

Use the MODIFY *hisproc* command to manage collection of hardware event data for System z10 or later machines. Use F *hisproc*, BEGIN to configure and start a run of data collection, and F *hisproc*, END to stop the run. You may start each run of hardware data collection either through explicit issuance of HIS commands, or through use of system automation.

During a run of hardware data collection, the system writes the data to z/OS UNIX System Services output files and to SMF record type 113. The system writes the raw data to SMF record type 113 at the start, the end, and defined intervals during the data collection run, and writes different types of data to the z/OS UNIX System Services output files at the end of the run. For more information about the different z/OS UNIX System Services output files, see [Accessing the output from a hardware event data collection run in z/OS MVS System Commands](#).

If there is a significant state change within the system, such as a CPU capability change during a collection run, the STATECHANGE parameter from the MODIFY *hisproc* command will determine the appropriate action. When the action is SAVE (the default), there will be two SMF 113 records at every state change, one for the end of the previous state and one for the start of the new state. There will also be an additional set of z/OS UNIX output files for each detected state change. Each z/OS UNIX output file only represents an interval of time between the start and the end of the same system state. Each interval of a collection run has a unique set of z/OS UNIX output files, determined by the sequence number as part of the filename (the xxx section of the filename). If a .MAP file is to be generated for the collection run, one is not generated at a every state change. A .MAP file is only generated at the end of a collection run with a sequence number representing the last collection run interval. The group of files relating to the same run can be identified through the file prefix (SYSHISyyyymmdd.hhmmss).

Before you issue the F *hisproc*, BEGIN command to configure and start hardware event gathering on a system, you must do some set up steps. See [Setting up hardware event data collection in z/OS MVS System Commands](#).

**Note:**

1. It is important to assign a sufficiently high dispatch priority to the instrumentation started task *hisproc*, so that the task can write the sampling data to the .SMP output files in a timely manner.
2. The sampling frequency is the number of samples per second regardless of processor model. The relative impact of sampling at the default rate is greater for slower processors. For slower sub-capacity models, a lower sampling frequency might be appropriate. For more information, see the SAMPFREQ parameter description.

See [Accessing the output from a hardware event data collection run](#) in *z/OS MVS System Commands* for information on the different files the HIS Profiler generates, depending on the F *hisproc*, BEGIN parameters you specify.

If you configure a new processor online in a system **after** you've already issued the F *hisproc*, BEGIN command to start a data collection run for that system, HIS might not collect data for that processor. To ensure that data is collected for all the processors on a system, bring the processors online before beginning a hardware data event collection run. The system does not collect data on a processor that is configured offline.

Note that z/OS IRD processor management can configure processors offline or online automatically. A processor is online at the start of the instrumentation run, but it might be configured offline (and sometimes online again) during the run. The system does not collect data on the offline processor.

```
F hisproc,{BEGIN|B}
    [, {TITLE|TT}='textdata']
    [, PATH='pathname']
    [, {DDNAME|DD}=ddname]
    [, {DURATION|DUR}=duration_value]
    [, {STATECHANGE|SC}={SAVE|STOP|IGNORE}]
    [, {DATALOSS|DL}={IGNORE|STOP}]

    [, SMP={YES|NO}]
        [, {SAMPTYPE|ST}=samptype|PERSIST|NONE]
        [, {BUFCNT|BUF}={bufcnt|PERSIST}]
        [, {SAMPFREQ|SF}=freq|PERSIST]

    [, CNT={YES|NO}]
        [, {CNTSET|CTRSET|CTR}={COMPLETE
            |SOFTWARE
            |HARDWARE
            |(ctr1,ctr2,...ctrn)}]
        [, {CNTINTVAL|CI|SMFINTVAL|SI}={SYNC|intv}]
        [, CNTFILE={YES|NO}]

    [, MAP={YES|NO}]
        [, {MAPASID|MAS}={ALL|(asid1,asid2,...asidn)}]
        [, {MAPJOB|MJOB}={job1,job2,...jobn}]
        [, {MAPVERBOSE|MAPV}]

    [, {CNTONLY|CTRONLY|MAPONLY}]

{END|E}

{SERVICE|S}
    [, {DDNAME|DD}=ddname]
    [, {BUFCNT|BUF}={bufcnt|PERSIST}]
    [, {SAMPFREQ|SF}=freq|PERSIST]
    [, {SAMPTYPE|ST}=samptype|PERSIST|NONE]
    [, {REFRESH|REFR}]

{WIC}
    [, INMEM=inmem]
    [, WICPATH=wicpath]
```

The parameters are:

### ***hisproc***

The name of the hardware instrumentation services (HIS) catalogued startup procedure.

### **{BEGIN|B} or {END|E} or {SERVICE|S} or WIC**

You must specify either BEGIN or END or SERVICE on the F *hisproc* command to begin or end a run of hardware event data collection, or update the hardware event data collection service parameters, for a system. You must specify WIC to update WIC parameters for a system.

**BEGIN|B**

Specifies that the system begin a HIS Profiler collection run. Note that you must first start the HIS address space with the `START hisproc` command before you can start the HIS Profiler. Note that any service parameters specified will be accepted and will be used as the new service parameter values.

**END|E**

Specifies that the system end a HIS Profiler collection run. As part of the end processing, the system writes the hardware data to your z/OS UNIX System Services output files and writes the last SMF record type 113 to the SMF data set.

Note that while both `F hisproc,END` and `STOP hisproc` end the data collection run, the two commands are different:

- Using `F hisproc,END` ends the HIS Profiler collection. You must issue `F hisproc,BEGIN` command to restart a HIS Profiler collection.
- Using `STOP hisproc` both ends the HIS Profiler collection and stops the HIS address space, which also disables the HISSERV service. You must reissue the `START hisproc` command before starting the HIS Profiler with the `F hisproc,BEGIN` command, as well as any HISSERV exploiter.

**SERVICE|S**

Specifies that the system should update the service parameters for all exploiters of the HISSERV service. If the HIS Profiler run is currently collecting instrumentation data, it will be notified of the change and a state change event will occur.

**WIC**

Specify that the system should read SMF records specified by the INMEM and write those records to files at the designated WICPATH for IBM z/OS Workload Interaction Correlator.

**TITLE|TT='textdata'**

Optional parameter specifying up to 32 characters of text data meaningful to the user. This data will be displayed in the z/OS UNIX System Services .CNT output file. For example, you might use this field to create an eye catcher to identify the reason for a hardware data collection run. The text data must be enclosed in single quotation marks.

**PATH='pathname'**

Specifies the z/OS UNIX System Services path (in a local file system) where you want the system to write the collected hardware event data for one run. The system creates MAP, CNT, and SMP output files (the .MAP, .CNT and .SMP files) and writes the collected data to the output files at the end of a run. This parameter is required, unless you have already set up the file path using the HOME keyword in the OMVS segment of an `ADDUSER hisproc` or `ALTUSER hisproc` command. See [Setting up hardware event data collection in z/OS MVS System Commands](#) for more information.

PATH does not affect WIC files. Use WICPATH to affect WIC files.

The *pathname* must be enclosed in single quotation marks and can be up to 64 characters. For example, you could specify the following for *pathname*:

- `PATH= ' . '`, which means to use the current working directory.
- `PATH= ' /u/john '`, which means to use the absolute directory `/u/john`.
- `PATH= ' user/mary '`, which means to use the relative directory `user/mary`, which is relative to the current working directory.

If instrumentation is to be run concurrently on multiple LPARs with a shared file system, a unique path that is specified by the PATH parameter must be created for each LPAR sharing the file system.

See [Accessing the output from a hardware event data collection run in z/OS MVS System Commands](#) for information on the files the HIS Profiler generates at the z/OS UNIX System Services path.

**WICPATH={NONE|PERSIST|'value'}**

Optional keyword specifying the USS file path value for WIC in quotes, the keyword NONE, or the keyword PERSIST.

**WICPATH=NONE**

Indicates that the system should not write data to a WICPATH. HIS will stop writing to a WICPATH if it was previously writing to another WICPATH.

**WICPATH=PERSIST**

Indicates that the system should continue unchanged, using the previously specified option.

**WICPATH='value'**

'value' is a path in the local file system where the system should write the collected WIC data. The system creates a file and writes the collected data to this path. At the top of every hour, HIS will switch and write to a new file.

Up to 64 characters can be specified for the WICPATH.

Pathnames can begin with or without a slash:

- A pathname that begins with a slash is an absolute pathname. The slash refers to the root directory where the search for the file starts.
- A pathname that does not begin with a slash is a relative pathname. The search for the file starts at the working directory.

The WICPATH value must be enclosed by single quotes while the WICPATH proper cannot contain a single quote, otherwise a syntax error is received. The file path value can be in mixed case. You can identify the current working directory (CWD) to be used by specifying WICPATH= ' . ' on the MODIFY HIS command.

Consider the following example:

```
WICPATH='.'          <--- use current working directory
WICPATH='/u/john'    <--- use this absolute directory
WICPATH='user/mary'  <--- path starting form the current working directory
```

**Note:** USS files with the following names will be generated within the specified path: <>  
 SYSWICyyyymmdd.hh.xxx.sysname.

Where:

SYSWIC = EBCDIC 'SYSWIC'

yyyymmdd = year/month/day

hh = hour

xxx = file sequence number

sysname = the name of the system that generated the file

The installation should choose a WICPATH different from the PATH keyword, although there is no enforcement.

**CAUTION:** HIS will delete all the WIC files that the system generates in the old path, when either changing from an existing WICPATH to WICPATH=NONE or when changing to a new WICPATH.

If the WICPATH keyword is omitted, WICPATH=PERSIST is assumed. HIS starts with WICPATH=NONE before any HIS parameters are accepted.

**CAUTION:** HIS will delete WIC files that the system generates, in this path, that are at least 14 days old.

WICPATH will not affect CNT, SMP, or MAP files. Use PATH to affect these files.

**INMEM={NONE|PERSIST|value}**

**INMEM**

An optional keyword specifying the SMF in memory name to connect and read SMF records from, the keyword NONE, or the keyword PERSIST.

INMEM can be have one of the following values:

**NONE**

Indicates for the system not to read data from an INMEM. HIS will stop reading from the INMEM and disconnect from it if HIS was previously reading from an INMEM.

**PERSIST**

Indicates for the system to continue, unchanged, using the previously specified option.

**value**

Indicates for the system to connect to the indicated SMF INMEM resource. HIS will copy any newly written SMF records into a file in the WICPATH. Every minute HIS will check for, and write, new records into the file. At the top of every hour, HIS will complete writing to an existing file and a new file will be created.

The INMEM value must match the z/OS log stream naming conventions. The INMEM must be at least 7 but no more than 26 characters long.

If the INMEM keyword is omitted, INMEM=PERSIST is assumed. HIS starts with INMEM=NONE before any HIS parameters are accepted.

**{DDNAME|DD}=ddname**

Optional parameter specifying the 1- to 8-character name identifying the job control language (JCL) data definition (DD) statement that defines a command file for HIS MODIFY *hisproc* parameters. The command file referenced contains parameters for HIS Profiler collection runs or for a service update (depending on whether it is a BEGIN or SERVICE request) which are set up the same way they would appear in the MODIFY command - the same rules and formatting apply to the command file that you would use in the console command. The command file gives you an alternative to specifying data collection options in a MODIFY *hisproc* command, which can be useful if you have difficulty fitting all the desired parameters on the command.

Note that duplicate parameters on the MODIFY *hisproc* command and in the command file are not allowed and will be flagged as errors.

When the system processes the DDNAME parameter, the system takes the character string on each line and concatenates them into one command string for parsing. It then merges the command file contents with the parameters specified on the MODIFY *hisproc* command used to process the command. Thus, you can specify some of your data collection options in the command file, and some in the MODIFY *hisproc* command. For example, you can put all the constant parameters in the command file and dynamic parameters on the command line. For example, TITLE can be used to document the reason for the run on the command line.

The statements in the command file are normal MODIFY *hisproc* parameters, without the MODIFY *hisproc*, BEGIN or MODIFY *hisproc*, SERVICE heading.

The following shows an example of valid command file contents assuming a MODIFY *hisproc*, BEGIN, DDNAME=ddname1 heading:

```
PATH= '/user/john',CTRSET=(BASIC,PROB),
BUFCNT=PERSIST, SAMPTYPE=BASIC,
SAMPFREQ=850000,DATALOSS=IGNORE,
MAPASID=(7,0E,E1D),
MAPJOB=(PROG*,DB*,GRS,JE??)
```

These commands would be valid in a command file for a MODIFY *hisproc*, SERVICE, DDNAME=ddname2 heading :

```
SAMPTYPE=BASIC,
SAMPFREQ=130000
```



Use the following syntax rules for the command file:

- The parameters specified can reside on multiple lines of the command file.
- You must separate keywords with commas.
- The system treats a blank character between any two non-blank characters on the same line (unless the blank is quoted within quotation marks) as the end of the command.
- You can use columns 1-72 of each line.
- A quoted string (a title or path name, for example) can not span more than one line.
- Parameters and values in the command file must be in UPPER case, unless the parameter value is a quoted string.

In order to use a command file for MODIFY *hisproc*, you must specify the *ddname* in the HIS started cataloged start up procedure *hisproc*. The following example shows how to use a command file named CMDFILE for the MODIFY *hisproc* command. In the example, DDNAME1 specifies a command file CMDFILE, which is a member of SAMPLING.PROCLIB and contains some or all of the parameters for the MODIFY *hisproc* command. Likewise, DDNAME2 specifies command file CMDFILE2 which would contain a different set of parameters.

```
//HISPROC PROC
//HISPROC EXEC PGM=HISINIT,REGION=0K,TIME=NOLIMIT
//DDNAME1 DD DSN=SAMPLING.PROCLIB(CMDFILE),DISP=SHR
//DDNAME2 DD DSN=SAMPLING.PROCLIB(CMDFILE2),DISP=SHR
//SYSPRINT DD SYSOUT=*
```

The command file must have a RECFM format of LRECL=80 fixed length record.

Note that the same syntax rules apply to the parameters in the command file as to a MODIFY *hisproc* command. The system flags duplicate parameters or mutually exclusive parameters entered in the MODIFY *hisproc* command and the command file as errors and you will receive an error message.

There is no default for the DDNAME.

#### **{DURATION|DUR}=duration\_value**

Optional parameter specifying the duration, in minutes, that you want the hardware event data collection run to last. At the end of this period, data collection stops automatically.

The default DURATION for instruction address sampling is 10 minutes. If request is for event counters only (CTONLY), the default DURATION is unlimited. To stop an unlimited data collection run, you must explicitly specify one of the following commands:

- F *hisproc*,END
- STOP *hisproc*

**Value range:** 1–1440 (minutes)

#### **{STATECHANGE|SC}={SAVE|STOP|IGNORE}**

Optional parameter specifying the action you want the system to take when a significant change to the system is detected. An example of a significant state change would be if the capability of a CPU has changed. You can specify the following for STATECHANGE:

- **SAVE**, the current run of hardware event data collection will save all data collected as if the duration of the run has expired. It will then resume collecting hardware event data in the new state for the remaining duration of the current collection run.
- **STOP**, the current run of hardware event data collection for a system is ended as if the duration of the run has expired.
- **IGNORE**, the current run of hardware event data collection continues as if no state change had been detected.

Default: SAVE

**{DATALOSS|DL}={IGNORE|STOP}**

Optional keyword specifying the action you want the system to take if data loss occurs during the instrumentation run. Data loss can occur from any of the following events:

- Sampling buffer overflow condition
- Loss of Sample Data Measurement Alert (raised by hardware)
- Loss of Counter Data Measurement Alert (raised by hardware)

You can specify the following values for the DATALOSS parameter:

**IGNORE |I**

The system continues with the instrumentation run if data loss occurs.

**STOP |S**

The system stops the instrumentation run if data loss occurs.

You can reduce the chances of losing data in the event of a buffer overflow by either allocating more buffers for data collection or increasing the priority of the HIS started task.

A Loss of Sample Data Measurement Alert and a Loss of Counter Data Measurement Alert are determined by the hardware and cannot be predicted nor prevented.

Default: IGNORE

**SMP={YES|NO}**

Specifies that you want to collect sample data. The default is YES.

**{SAMPTYPE|ST}={samptype|PERSIST|NONE}**

Optional parameter that specifies the sampling functions to be performed. This parameter is only valid when SMP=YES is in effect.

Whether this option is specified or omitted under the MODIFY *hisproc*, BEGIN or the MODIFY *hisproc*, SERVICE heading, the change is global to all HISSERV exploiters.

***samptype***

The sampling functions (*samptype*) supported include:

- **BASIC or B:** for basic sampling functions.
- **DIAG or D:** for basic and diagnostic sampling. Diagnostic sampling provides additional information over what is available in basic sampling. When diagnostic sampling is requested, basic sampling is also automatically selected. To use the diagnostic sampling function, you must first authorize to the diagnostic sampling facilities on the SE console. Unless instructed by IBM, use ST=Basic.

For information about how to set up the authorization of the sampling facilities through the support element (SE) console, see *Support Element Operations Guide* for System z10 machine on the [Resource Link](http://www.ibm.com/servers/resourcelink) home page ([www.ibm.com/servers/resourcelink](http://www.ibm.com/servers/resourcelink)).

**PERSIST**

The sampling functions to be performed are not changed from its prior value.

**NONE**

No sampling functions will be performed, and no HISSERV exploiters will be authorized to collect sampling data.

Example: SAMPTYPE=DIAG

Default: Basic. If you do not specify SAMPTYPE, the default is used.

**{BUFCNT|BUF}={bufcnt|PERSIST}**

Optional parameter that specifies the number of sampling buffers. This parameter is only valid when SMP=YES is in effect.

Whether this option is specified or omitted under the MODIFY *hisproc*, BEGIN or the MODIFY *hisproc*, SERVICE heading, the change is global to all HISSERV exploiters.

**bufcnt**

The number of sampling buffers (in 4K pages) per processor for the system. A range of values between 4 - 1024 (pages) is supported.

The total number of sampling buffers the system uses is calculated from the BUFCNT specified, as follows:

$$(\text{BUFCNT} * \text{Number of active processors in the configuration})$$

If you specify too small a value for BUFCNT and the system runs out of buffer space, you might lose some sample data. If you specify a high sampling frequency on the SAMPFREQ parameter, HIS will consume more sampling buffer space and you might need a higher value for BUFCNT to prevent loss of samples.

**PERSIST**

The number of sampling buffers is not changed from its prior value.

Default: If you do not specify BUFCNT, the system calculates the number of buffers needed using the number of processors in the configuration and the sampling frequency in effect (SAMPFREQ).

**{SAMPFREQ|SF}=freq|PERSIST}**

Optional parameter that specifies the frequency for the sampling functions. This parameter is only valid when SMP=YES is in effect.

Whether this option is specified or omitted under the MODIFY *hisproc*, BEGIN or the MODIFY *hisproc*, SERVICE heading, the change is global to all HISSERV exploiters.

**freq**

is the total number of samples to be taken in a minute on all active processors in the configuration. For example, a *freq* value of 500000 specifies a sampling frequency of 500,000 samples per minute.

Note that the effective sampling rate is usually smaller than the specified SAMPFREQ for LPARs that share the processors. Samples are captured only on logical processors that are actively in use. (A waiting shared logical processor does not produce samples.)

**PERSIST**

The sampling frequency is not changed from its prior value.

Default: 800000, which is equivalent to 8 million samples in 10 minutes. This default may be aggressive for slower sub-capacity models. A less aggressive value of 130 000 should be used if there is an unacceptable performance degradation when the default value is used. The DURATION can be extended as needed if more samples are required. For example, to achieve 8 million samples with a sample frequency of 130 000, the duration of the sampling run would be 60 minutes.

If you do not specify SAMPFREQ, the default is used.

**CNT={YES|NO}**

Specifies that you want to collect event counter set data. The default is YES.

**{CNTSET|CTRSET|CTR}={COMPLETE|SOFTWARE|HARDWARE|(*ctr1*,*ctr2*,...*ctrn*)}**

Optional parameter specifying the set of counters you want to collect. **CTRSET** and **CTR** are aliases of **CNTSET**. This parameter is only valid when CNT=YES is in effect.

**COMPLETE**

When you specify COMPLETE the system collects event counters for all available counter sets. For example, if you have installed and authorized only the basic and problem-state counter sets on your system, a specification of CNTSET=COMPLETE, results in basic and problem-state counter set events being collected. Be aware that specifying CNTSET=COMPLETE also means collecting event counters that are not currently defined, but may be defined in the future. It is possible that event counter sets in the future may have unforeseen performance impacts on the system. Note that COMPLETE, is not enclosed in parentheses.

## SOFTWARE

When you specify SOFTWARE, the system collects all software related event counter sets. Any future counter set defined as a software counter set may have unforeseen performance impact on the system. Note that SOFTWARE is not enclosed in parentheses. CNTSET=SOFTWARE is currently equivalent to CNTSET=(Z).

## HARDWARE

When you specify HARDWARE, the system collects all hardware related event counter sets. Any future counter set defined as a hardware counter set will have little unforeseen performance impact on the system. Note that HARDWARE is not enclosed in parentheses. CNTSET=ALL is an alias to CNTSET=HARDWARE. CNTSET=HARDWARE is currently equivalent to CNTSET=(B,P,C,E).

### (ctr1,ctr2,...ctrn)

You can also specify a list of counter sets to collect. This comma separated list can be one or more of the following counter sets:

- **B or BASIC for basic:** This is a hardware CPUcounter set and includes architected system activities, such as cycle count, instruction count, level 1 cache misses, for example, for a CPU in either the problem or supervisor state.
- **P or PROB for problem state:** This is a CPU hardware counter set and includes the architected system activities only when the CPU is in the problem state.
- **C or Crypto for crypto:** This is a hardware CPU counter set and includes the architected crypto activities, such as function count, cycle count, blocked function count, and blocked cycle count for each of the PRNG, SHA, DEA, and AES functions.
- **E or EXT for extended:** This is a hardware CPUcounter set and includes model dependent counters described in model dependent system library publications. For more information about the extended counters for the supported counter version number for each model, see the extended counter definition in "The CPU-Measurement Facility" available on the Resource Link home page ([www.ibm.com/servers/resourcelink](http://www.ibm.com/servers/resourcelink)).
- **Z or ZOS for z/OS:** This is a software CPUcounter set and includes counters for z/OS related activities. Enabling the z/OS counter set incurs a slightly higher performance overhead than the hardware counter sets. The z/OS counter set uses the Enhanced Monitor Facility. Note the z/OS counters are not defined and are primarily used for IBM diagnostic purposes.
- **MTD or MTDIAG for MT-diagnostic:** This is a hardware core counter set and includes model dependent counters described in model dependent system library publications. For more information about the MT-diagnostic counters for the supported counter version number for each model, see the MT-diagnostic counter definition in "The CPU-Measurement Facility" available on the Resource Link home page ([www.ibm.com/servers/resourcelink](http://www.ibm.com/servers/resourcelink)).

**Default:** If you do not specify CNTSET, HIS uses the B (basic) and P (problem state) counter sets. For example, use CNTSET=(B,P,E) to specify three counter sets (Basic, Problem state, and extended), and use CNTSET=(BASIC,E) to specify two counter sets (BASIC and extended).

### {CNTINTVAL|CI|SMFINTVAL|SI}={SYNC|intv}

Optional parameter that specifies the interval at which SMF record type 113 records are recorded. **CI**, **SMFINTVAL**, and **SI** are aliases of **CNTINTVAL**. This parameter is only valid when CNT=YES is in effect. The following values are accepted:

#### SYNC

Specifies that the SMF record type 113 should be recorded in sync with the SMF global recording interval. The global recording interval is set by parameters **INTVAL** and **SYNCVAL** in parmlib member SMFPRMxx. Note that if SMF is not active, global recording interval processing is disabled or there is some other error that prevents HIS to sync with the SMF global recording interval, no interval SMF record type 113 will be recorded. Receiving messages IEE500I or IEE068A is an example of when no SMF record type 113 will be recorded. For more information on the SMF global recording interval, see [INTVAL and SYNCVAL - Performing interval accounting](#) in *z/OS MVS System Management Facilities (SMF)*.

**intv**

An interval, in minutes, that you want to record interval SMF record type 113 records. The acceptable value range is 1 - 60.

If **CNTINTVAL** is not specified, the default is 15.

**CNTFILE={YES|NO}**

Optional parameter that specifies whether event counter set data is to be written to z/OS UNIX System Services .CNT output files. This parameter is only valid when CNT=YES is in effect.

**Default:** YES

**MAP={YES|NO}**

Specifies that you want to collect load module mapping information. The default value is NO.

**{MAPASID|MAS}={ALL|(asid1,asid2,...asidn)}**

Optional parameter that specifies a list of address space IDs (ASIDs), in hexadecimal, to identify the address spaces for which you want to collect private load module map data. HIS will collect the virtual storage addresses of modules loaded into private virtual storage for the specified ASIDs that are not terminating, swapping, swapped, or inactive. This parameter is only valid when you specify MAP=YES.

Acceptable hexadecimal values are between X'1' and X'7FFF'.

You can specify up to 32 ASIDs on the **MAPASID** parameter.

- If you need to collect data from more than 32 address spaces, you can use the **MAPJOB** parameter in place of or in addition to the **MAPASID** parameter.
- The system supports a total of up to 128 address spaces, including those specified in both the **MAPASID** and **MAPJOB** parameters. If you need load module map data for more than 128 address spaces, specify MAPASID=ALL. When you specify MAPASID=ALL, the load module map data for all active ASIDs can be returned. When collecting data for IBM, specify MAPASID=ALL, unless directed otherwise.
- If you specify **MAPASID** with any value, you will always map the nucleus and LPA areas as well.
- You may want to enable tracking of directed load modules with the **SETPROG TRACKDIRLOAD** command so that your results are complete. See [Tracking directed load modules in z/OS MVS System Commands](#).

**Examples:**

- MAPASID=(7,8,32)
- MAPASID=ALL

**Default:** None.

**{MAPJOB|MJOB}=(job1,job2,...jobn)**

Optional parameter specifying a list of job names for which HIS will collect the virtual storage addresses of modules loaded into private virtual storage. HIS collects the virtual storage addresses for jobs that are not terminating, swapping, swapped, or inactive. The system ignores duplicate, invalid, or inactive job names specified. This parameter is only valid when you specify MAP=YES.

Specify job names with 1-8 characters, following the rules for a valid job name. You can use wildcard characters \* and ? for pattern matching of job names. See [Using wildcards in commands in z/OS MVS System Commands](#) for more information.

You can specify up to 32 job names, including all the pattern matches from wildcard characters. Note that using wildcard characters can result in requesting more than 32 physical jobs.

When you specify job names on **MAPJOB**, the system converts each active job name into one or more ASIDs. If the job is not active when the system does the load module mapping, the system will not produce load module information for that job. The system supports a total of up to 128 address spaces, including those specified on both the **MAPASID** and **MAPJOB** parameters. If you specify more than 128 address spaces, the system produces load module mapping data for the

first 128 address spaces and ignores the rest. For example, **MAPJOB=(\*)** might produce load module mapping information for 128 active address spaces, starting with ASID 1. If you need load module map data for more than 128 address spaces, specify **MAPASID=ALL**. When you specify **MAPASID=ALL**, the load module map data for all active ASIDs can be returned. If you specify **MAPJOB** with any value, you will always map the nucleus and LPA areas as well.

You may want to enable tracking of directed load modules with the **SETPROG TRACKDIRLOAD** command so that your results are complete. See [Tracking directed load modules in z/OS MVS System Commands](#).

**Examples:**

- **MAPJOB=(task1,grs,omvs,db\*,task2)**
- **MAPJOB=(o\*s)**
- **MAPJOB=(JOB1??,JE\*)**

**Default:** None.

**{MAPVERBOSE|MAPV}**

Optional keyword specifying that you want the system to produce additional diagnostic information about any errors encountered during the load module mapping phase of data collection. **MAPVERBOSE** specifies that the system issues system messages to the job log if it encounters multiple errors during load module mapping. This parameter is only valid when you specify **MAP=YES**.

You can only specify **MAPVERBOSE** if you also activate the load module mapping with the **MAPONLY**, **MAPASID**, or **MAPJOB** parameters. If you specify **MAPVERBOSE** without specifying one of these load module mapping parameters, the system ignores the **MAPVERBOSE** parameter.

**Default:** None; no load module mapping diagnostic information is collected.

**{CNTONLY|CTRONLY|MAPONLY}**

These optional keywords allow you to limit collection, as follows:

**CNTONLY|CTRONLY**

Specifies that you want to collect only event counter set data. Data collection for instruction address sampling is not activated. The system generates only a .CNT z/OS UNIX System Services output file. (The system does not generate .SMP and .MAP output files if you specify **CNTONLY**.) **CTRONLY** is an alias of **CNTONLY**.

When this keyword is specified, you cannot specify keywords associated with instruction sampling, such as **SAMPTYPE** and **SAMPFREQ**.

Keywords allowed with **CNTONLY** include **TITLE**, **PATH**, **DURATION**, **CNTSET**.

The **CNTONLY** parameter is equivalent to **CNT=YES**, **SMP=NO**, **MAP=NO**.

**MAPONLY**

Specifies that you want to collect only load module mapping information. Data collection event counter sets and instruction address sampling is not activated. The system generates only a .MAP z/OS UNIX System Services output file. (The system does not generate .SMP and .CNT output files if you specify **MAPONLY**.)

When you specify **MAPONLY**, you must also specify **MAPASID** or **MAPJOB** (or both) to identify the address spaces for which you want private load module map data. You cannot specify keywords associated with sampling and counter sets, such as **SAMPTYPE**, **SAMPFREQ**, or **CNTSET**.

Keywords allowed with **MAPONLY** include **TITLE**, **PATH**, **MAPASID**, **MAPJOB**.

The **MAPONLY** parameter is equivalent to **CNT=NO**, **SMP=NO**, **MAP=YES**.

### **{REFRESH|REFR}**

Optional parameter instructing the system to refresh the HISSERV service. A refresh of the service will allow the service to begin using any counter set or sample type recently authorized through the Hardware Management Console. **REFRESH** is mutually exclusive with all other service parameters.

**Example 1:** Start the HIS Profiler data collection using defaults, including SMF data, and create .CNT and SMP.cpu#. files. By default, the following instrumentation data are captured besides SMF data:

- Data from the basic and problem state counter sets
- Basic sampling data

Because no service parameters were explicitly specified with the PERSIST option, all service parameters will be changed to the defaults.

```
MODIFY hisproc,BEGIN
```

**Example 2:** Start the HIS Profiler for data collection that runs for 12 minutes, collects load module mapping information for jobs PGMA and PGMB. collects SMF data and creates .MAP, .CNT and SMP.cpu#. files. Because all service parameters are explicitly specified with the PERSIST option, they will remain at their prior values.

```
MODIFY hisproc,BEGIN,DURATION=12,MAPJOB=(PGMA,PGMB),BUFCNT=PERSIST,  
SAMPTYPE=PERSIST,SAMPFREQ=PERSIST
```

**Example 3:** End the HIS Profiler for data collection:

```
MODIFY hisproc,END
```

**Example 4:** Refresh HISSERV to begin using the counter sets and sample types that were recently authorized through the Hardware Management Console, and then increase the buffer count to allow for additional data. **SAMPREQ** and **SAMPTYPE** will be implicitly changed to their default values.

```
MODIFY hisproc,SERVICE,REFRESH  
MODIFY hisproc,SERVICE,BUFCNT=100
```

**Example 5:** Direct HISSERV to stop generating sampling data for profilers requesting it. These changes are global and will affect all exploiters. **BUFCNT** and **SAMPFREQ** are specified as PERSIST to maintain their values.

```
MODIFY hisproc,SERVICE,SAMPTYPE=NONE,BUFCNT=PERSIST,SAMPFREQ=PERSIST
```

**Example 6:** Direct HISSERV to generate basic sampling data, the default, at a frequency of 100,000 samples per minute, and use the default buffer count. These changes are global and will affect all exploiters.

```
MODIFY hisproc,SERVICE,SAMPFREQ=100000
```

**Example 7:** Direct the HIS Profiler to begin collecting just counter data for problem diagnosis for IBM. These specific options may be the most commonly requested by IBM. All HIS exploiters are affected. This particular command labels the measurement 'Counters Only', requests data for basic and extended counter sets, sends the output to the .CNT file in the /HIS OMVS directory, (no .MAP or .SMP files are produced). Data collection for instruction address sampling is not activated, and the **SAMPTYPE**, **SAMPFREQ**, and **BUFCNT** values have been preserved with the PERSIST option. SMF record type 113 will be recorded in sync with the SMF global recording interval, and this current run will save all data collected as if the duration of the run has expired. It will then resume collecting hardware event data in the new state until a MODIFY hisproc,END command is issued.

```
MODIFY hisproc,BEGIN,TT='Counters Only',PATH='/HIS',CNTONLY,CNTSET=(B,E),SI=SYNC,  
SC=SAVE,SAMPTYPE=PERSIST,SAMPFREQ=PERSIST,  
BUFCNT=PERSIST
```

**Example 8:** Direct the HIS Profiler to begin collecting extensive data for problem diagnosis for IBM. These specific options may be the most commonly requested by IBM. All HIS exploiters are affected by

the MODIFY *hisproc*,BEGIN command. This command labels the measurement 'FOR IBM', sends the output to the /HIS OMVS directory, requests data for basic, problem, cryptographic, and extended counter sets, requests a map of all swapped in address spaces, and uses a buffer count of 1024 4K buffers. The recommended buffer count value is 1024 for minimal overhead at the expense of more storage (1024 4K blocks per processor.) The sampling data is basic, with the frequency of 800,000 samplings per minute and will run for 10 minutes.

```
MODIFY hisproc,BEGIN,TT='FOR IBM',path=/HIS,CNTSET=(B,P,C,E),MAS=ALL,BUF=1024,ST=B,  
SF=800000,DUR=10
```

**Example 9:** Consider the following example to direct HIS processing to read from SMF in-memory object IFASMF.WIC and write the SMF data to /u/his/wic.

```
SY1 F HIS,WIC,INMEM=IFASMF.WIC,WICPATH='/u/his/wic'  
SY1 HIS038I HIS WIC PARAMETER(S) ACCEPTED  
SY1 HIS015I 14.54.21 DISPLAY HIS 058  
HIS WIC STATUS: ACTIVE  
INMEM= IFASMF.WIC  
WICPATH= /u/his/wic
```

## HIS and IFA system messages

*z/OS MVS System Messages, Vol 6 (GOS-IEA)* and *z/OS MVS System Messages, Vol 8 (IEF-IGD)* contain the following new and updated messages.

<b>HIS009I</b>	<b>MODIFY <i>hisproc</i> COMMAND FAILED. SPECIFIED PATH INACCESSIBLE. BPX1ACC SERVICE FAILED with RC=returncode, RSN=reasoncode.</b>
----------------	--

### Explanation

The MODIFY *hisproc* command fails. Following a MODIFY *hisproc*,BEGIN, the specified (or default) file PATH is not accessible. Following a MODIFY *hisproc*,WIC, the specified WICPATH is not accessible. The path either does not exist, or does not have the required read and write access attributes.

When the system issued the a z/OS UNIX Systems Services BPX1ACC service to verify the PATH or WICPATH value, it fails and returns the listed return code and reason code. See the following list for return and reason code meanings:

#### **RC=9C (EMVSINITIAL) and RSN=xxxx00F9**

The HIS User ID, *hisproc*, has not been defined by the security product. When you add user *hisproc*, you must create an OMVS segment with a default HOME directory.

#### **RC=81 (ENOENT)**

The specified file PATH does not exist.

#### **RC=86 (ENOSYS) or 6F (EACCES)**

HIS cannot access the file path specified (or default) on the MODIFY *hisproc* command. The file path might not have the appropriate read or write access attributes for HIS to access it.

The ENOSYS error might be caused by HIS not having the correct authority to access the HOME directory.

### Other return and reason codes

See *z/OS UNIX System Services Messages and Codes* for information about the return and reason codes.

See *z/OS UNIX System Services Programming: Assembler Callable Services Reference* for information about the BPX1ACC service.

If you did not specify a path on the command, or if a relative path is specified, the system uses the default path that is defined by the HOME directory of the OMVS segment when the *hisproc* user was added to the system (using the ADDUSER TSO command). You must create an OMVS segment with a default HOME directory when you set up the hardware instrumentation services (HIS) user. The default file path, if valid, is the path for all the output files of the MODIFY *hisproc* command. If you specify a file path on the MODIFY *hisproc* command, the command overrides the HOME directory value. For more information about defining OMVS segments, see *z/OS UNIX System Services Planning*.



In the message text:

***hisproc***

The name of the HIS cataloged startup procedure.

***returncode***

The return code from BPX1ACC.

***reasoncode***

The reason code from BPX1ACC.

## System action

The system does not complete the command issued.

## Operator response

Refer to the return code and reason code in the message for the exact cause of this error.

- If you do not specify a path on the MODIFY *hisproc* command, verify that there is a default path defined on the HOME directory keyword in the OMVS segment of the *hisproc* user.
- If you specify a relative path, verify that the full path (when concatenated with the HOME directory in the OMVS segment of the *hisproc* user) exists.
- If the path exists, verify that the path has the proper read and write access attributes.

**Note:** The path name is case-sensitive.

You can display the home directory in the OMVS segment of the *hisproc* user by issuing the following TSO command:

```
ListUser hisproc OMVS noracf
```

You can enter the path name in either absolute form (starting with a "/"), or relative form (without a "/"). If you use the relative form, the system uses the HOME directory of the OMVS segment for the *hisproc* user.

## System programmer response

Ensure your HIS user ID, *hisproc*, is correctly defined:

- If the HIS user ID, *hisproc*, has not been defined by the security product, add the *hisproc* user ID by the following TSO command, where the UID and HOME values are just examples:

```
ADDUSER hisproc OMVS(UID(25) HOME('/user'))
```

- Specify a HOME directory in the OMVS segment for the added *hisproc* user. The HOME directory is the default PATH for the MODIFY *hisproc* command.
- Add an OMVS segment or HOME directory to the *hisproc* user by the following TSO command, where the UID and HOME values are just examples

```
ALTUSER hisproc OMVS(UID(25) HOME('/user'))
```

- Create the HOME directory (or any specific file path) using the mkdir OMVS command. Set up appropriate file access attributes with the chmod or chown OMVS commands.

If the problem was caused by a path or user ID setup error, you must correct the error and then force stop (FORCE *hisproc*,ARM) and restart (START *hisproc*) the HIS address space before beginning data collection again. If you do not stop and restart the HIS address space, the error can recur, even if you corrected it. You must restart the HIS address space for any of the following actions:

- Defining a new path using mkdir
- Modifying the access attributes of an existing path using chmod

- Adding (ADDUSER) or altering (ALTUSER) the *hisproc* user ID. This includes (but is not limited to) altering the OMVS segment or the HOME directory.

## Problem determination

None.

## Source

Hardware instrumentation services (HIS)

## Module

HISICMD

## Routing code

-

## Descriptor code

5

---

**HIS015I** *hh.mm.ss DISPLAY HIS text*

## Explanation

This message is issued in response to the DISPLAY HIS command and displays the status or results of the latest MODIFY *hisproc* command issued to start or stop collection of hardware event data. The contents of the message varies depending on the parameters issued with the MODIFY *hisproc* command.

In response to DISPLAY HIS, the message *text* is:

```
hisproc hisasid status
COMMAND: MODIFY command-data
START TIME: yyyy/mm/dd hh:mm:ss
END TIME:   yyyy/mm/dd hh:mm:ss
COMPLETION STATUS: comp-stat
FILE PREFIX: fileprefix
LOST SAMPLES: samplostcnt
COUNTER VERSION NUMBER 1: ctr-vers1
COUNTER VERSION NUMBER 2: ctr-vers2
COMMAND PARAMETER VALUES USED:
  TITLE= title
  PATH=  pathname
  DDNAME= ddname
  DURATION= duration-value (MINUTES)
  DATALOSS= datalossact
  STATECHANGE= statechange
  SMP=YES|NO
    BUFCNT= bufcnt (PAGES/PROCESSOR) | PERSIST | DEFAULT
    SAMPTYPE= samptype-values | PERSIST | DEFAULT | NONE
    SAMPFREQ= freq (SAMPLES/MINUTE) | PERSIST | DEFAULT
  CNT=YES|NO
    COUNTER SET= ctrset-values
    CNTINTVAL= cntintval
    CNTFILE=YES|NO
  MAP=YES|NO
    MAPASID= xxxx
    MAPJOB=  zzzz
HISSESV STATUS: serv-stat
EVENT
  AUTHORIZED=serv-authservtyp-values
             serv-authservtyp-values
  ENABLED=serv-enabservtyp-values
           serv-enabservtyp-values
SAMPLE
  AUTHORIZED=serv-authsmptyp-values
  ENABLED=serv-enabsmptyp-values
  BUFCNT= serv-bufcnt (PAGES/PROCESSOR)
```

```

SAMPFREQ= serv-freq(SAMPLES/MINUTE)
PROFILER
NAME      START      QUERY      SAMPLE    S F
yyyyyyyy  yyyy/mm/dd hh:mm:ss +hh:mm:ss:uuuuuu +uuuuuu    s f
HIS WIC STATUS: wicstatus
INMEM= inmem
WICPATH= wicpath

```

In response to D HIS,WIC, the message *text* is:

```

HIS WIC STATUS: wicstatus
INMEM= inmem
WICPATH= wicpath

```

In response to D HIS,WIC when the *hisproc* address space is not started or is ended, the message *text* is:

```

hisproc hisasid statusn

```

In response to D HIS,WIC on systems that do not support WIC, the message *text* is:

```

WIC NOT SUPPORTED ON THIS SYSTEM.

```

In the message text:

#### ***hh.mm.ss***

The time in hours (00-23), minutes (00-59), and seconds (00-59) for the DISPLAY HIS command.

#### ***hisproc***

The name of the (HIS) cataloged startup procedure.

#### ***hisasid***

The address space ID of HIS.

#### ***status***

The status of HIS, which is one of the following statuses :

##### **NOT STARTED**

HIS is not started.

##### **IDLE**

HIS is waiting for work.

##### **ACTIVE**

HIS is active and working.

##### **ENDED**

HIS ended.

#### **COMMAND: *command-data***

The text of the MODIFY *hisproc* command.

#### **START TIME: *yyyy/mm/dd hh:mm:ss***

##### ***yyyy/mm/dd***

The date in year/month/day format when the MODIFY *hisproc* command started.

##### ***hh:mm:ss***

The time in hours (00-23): minutes (00-59):seconds (00-59) format when the MODIFY *hisproc* command started.

#### **END TIME: *yyyy/mm/dd hh:mm:ss***

##### ***yyyy/mm/dd***

The date in year/month/day format when the MODIFY *hisproc* command ended. If data collection is still active, the end date is displayed as ----/--/--.

##### ***hh:mm:ss***

The time in hours (00-23): minutes (00-59):seconds (00-59) format when the MODIFY *hisproc* command ended. If data collection is still active, the end time is displayed as --:--:--.

**COMPLETION STATUS: *comp-stat***

The command completion status can be:

**NORMAL**

The Modify *hisproc* command has completed normally. SMF records and OMVS output files are valid.

**ABNORMAL**

The Modify *hisproc* command has abnormally ended. SMF records and OMVS output files might be incomplete and unusable.

-----

Data collection is still active. Output files are incomplete.

**FILE PREFIX: *fileprefix***

The prefix for all the data collection output files. *fileprefix* is in the format SYSHISyyyymmdd.hhmmss.

**LOST SAMPLES: *samplostcnt***

*SampLostCnt* is the cumulative running count of lost samples on all the processors because of buffer overflow errors.

**COUNTER VERSION NUMBER 1: *ctr-vers1***

*ctr-vers1* is the version number for the basic and problem state counter sets that HIS is recording.

Note that the meaning and the number of counters in a counter set might change with a new machine model. This counter set version number defines the contents of the basic and problem-state counter sets that are installed on the machine. For more information about these counters for the supported counter version number for each model, see publications on the [Resource Link home page \(www.ibm.com/servers/resourcelink\)](http://www.ibm.com/servers/resourcelink).

**COUNTER VERSION NUMBER 2: *ctr-vers2***

*ctr-vers2* is the version number for the crypto-activity, extended and MT-diagnostic counter sets that HIS is recording.

Note that the meaning and the number of counters in a counter set might change with a new machine model.

This counter set version number defines the contents of the crypto-activity, extended and MT-diagnostic counter sets that are installed on the machine. For more information about these counters for the supported counter version number for each model, see publications on the [Resource Link home page \(www.ibm.com/servers/resourcelink\)](http://www.ibm.com/servers/resourcelink).

**TITLE= *title***

*title* is the text data from the TITLE= parameter on the MODIFY *hisproc* command.

**PATH= *pathname***

*pathname* is the name of the path specified on the PATH= parameter of the MODIFY *hisproc* command.

**DDNAME= *ddname***

*ddname* is the data definition name specified on the DDNAME= parameter of the MODIFY *hisproc* command.

**DURATION= *duration-value* (MINUTES)**

*duration-value* displays the duration value, in minutes, of this run of HIS data collection.

**DATALOSS= *datalossact***

*datalossact* displays the action requested on the DATALOSS parameter on the MODIFY *hisproc* command for when data loss occurs during the instrumentation run. *datalossact* is one of the following:

**IGNORE**

Ignore the condition and continue with data collection.

**STOP**

End the data collection operation.

**STATECHANGE= *statechange***

Displays the action requested in the MODIFY *hisproc* command if a significant state change is detected. *statechange* is one of the following:

**SAVE**

Save the run data, marking an interval. Start the next interval for the current run.

**IGNORE**

Continue the current run.

**STOP**

Stop the run at the state change.

**SMP=YES|NO**

Displays whether sampling was requested on the MODIFY *hisproc*,BEGIN command.

**YES**

The HIS Profiler is collecting sampling data.

**NO**

The HIS Profiler is not collecting sampling data.

**BUFCNT= *bufcnt* | PERSIST | DEFAULT**

The BUFCNT parameter specified on the MODIFY *hisproc* command.

***bufcnt***

The value of the BUFCNT parameter in number of 4096-byte pages per processor.

**PERSIST**

The command specified that the system not change the current buffer count.

**DEFAULT**

The command did not specify the BUFCNT parameter and the system is using the default BUFCNT.

**SAMPTYPE= *samptype-values* | PERSIST | DEFAULT | NONE**

The status of the sample functions requested in the MODIFY *hisproc* command.

***samptype-values***

The list of the sample functions requested in the MODIFY *hisproc* command.

**PERSIST**

The command specified that the system not change the current sample functions.

**DEFAULT**

The command did not specify the SAMPTYPE and the system is using the default SAMPTYPE.

**NONE**

No sampling functions are performed .

**SAMPFREQ= *freq* | PERSIST | DEFAULT**

Displays the sample frequency requested on the SAMPFREQ parameter on the MODIFY *hisproc* command.

***freq***

The sample frequency that is requested on the SAMPFREQ parameter on the MODIFY *hisproc* command in number of samples per minute.

**PERSIST**

The command specified that the system does not change the current sampling frequency.

**DEFAULT**

The command did not specify the SAMPFREQ parameter and the system is using the default SAMPFREQ.

**CNT=YES|NO**

Displays whether counter data was requested on the MODIFY *hisproc*,BEGIN command.

**YES**

The HIS Profiler is collecting event counter set data.

**NO**

The HIS Profiler is not collecting event counter set data.

**COUNTER SET= *ctrset-values***

*ctrset-values* displays a list of the counter sets requested in the MODIFY *hisproc* command.

**CNTINTVAL= *cntintval***

If SYNC, the recording is synchronized with the SMF global recording interval. Otherwise, the duration of time between SMF record type 113, in minutes. For more information, see the topic on SMF global recording interval in [z/OS MVS System Management Facilities \(SMF\)](#).

**CNTFILE=YES|NO**

Displays whether event counter set data is to be written to the a z/OS UNIX Systems Services CNT output files, as requested on the MODIFY *hisproc*,BEGIN command.

**YES**

The HIS Profiler writes event counter set data to a z/OS UNIX Systems Services CNT output files.

**NO**

The HIS Profiler does not write event counter set data to a z/OS UNIX Systems Services CNT output files.

**MAP=YES|NO**

Displays whether map data was requested on the MODIFY *hisproc*,BEGIN command.

**YES**

The HIS Profiler is producing map data.

**NO**

The HIS Profiler is not producing map data.

**MAPASID= xxxx**

This field is displayed if you requested load module mapping in the MODIFY *hisproc* command. This field displays one or more address space IDs either from those entered on the MAPASID parameter or derived from the MAPJOB parameter. The MAPASID parameter specifies the address space ID for which you needed to collect load module mapping data.

Note that the job names in the MAPJOB parameter are converted to address space IDs at the point when load module mapping data is to be collected, which is at the end of the data collection. No derived address space IDs are displayed before this point.

**MAPJOB= zzzz**

This field is displayed if you requested load module mapping in the MODIFY *hisproc* command. This field displays job name values from the MAPJOB parameter for which you needed to collect load module mapping data.

**HISSERV STATUS: serv-stat**

The value of *serv-stat* might be:

**ACTIVE**

The HISSERV service is currently active.

**INACTIVE**

The HISSERV service is currently inactive.

**EVENT**

The list of active event sets.

**AUTHORIZED=serv-autheventtyp-values**

The event types that are currently authorized by the service.

**ENABLED=serv-enabeventtyp-values**

The event types that are currently enabled by the service.

**SAMPLE**

The sampling types that are currently active in the HISSERV service. The value might be:

**AUTHORIZED=serv-authsmpmptyp-values**

The sampling types that are currently authorized by the service.

**ENABLED=serv-enabsmpmptyp-values**

The sampling types that are currently enabled by the service.

**BUFCNT:bufcnt**

The number of sampling buffers currently allocated for each CPU.

**SAMPREQ:freq**

The current sampling frequency goal.

**PROFILER**

Information describing the profiler.

**NAME *prodname***

The external identifier that this profiler uses for itself.

**START *yyyy/mm/dd hh:mm:ss***

The date and time this profiler began profiling the system.

**QUERY *+hh:mm:ss:uuuuuu***

The average time this profiler waits between querying for counter data, where *hh* is hours (00-23) *mm* is minutes (00-59) *ss* is seconds (00-59) and *uuuuuu* is microseconds. The *+* indicates the exploiter averages over 1 day between counter queries. The greater the number, the less resource intensive the profiler is to the system.

**SAMPLE *+uuuuuu***

The average number of microseconds the profiler takes to handle a sampling data callback. The *+* indicates the exploiter averages over 1 second per callback. The lower the number, the less resource intensive the exploiter is to the system.

**SAMPLING\_PROFILER *s***

The sampling state of the profiler. *s* is one of the following:

***S***

The profiler requested sampling

***blank***

The profiler did not request sampling

***failed f***

The state of the profiler's callback routine. *f* is one of the following:

***F***

The profiler's callback routine has encountered an abend. See preceding CSV430I messages for diagnostic information.

***blank***

The profiler's callback routine has not abended.

**HIS WIC STATUS: *wicstatus***

The status of HIS is displayed about WIC resources.

***wicstatus***

Is one of the following:

***ACTIVE***

HIS is currently reading from the INMEM and writing to the WICPATH.

***INACTIVE***

HIS is not reading from the INMEM or writing to the WICPATH.

**INMEM= *inmem***

Displays the SMF in memory object that HIS is connected to, if any. When '*\*NONE\**' HIS is currently not connected to an INMEM and *wicstatus* is INACTIVE. If *inmem* is not '*\*NONE\**' and *wicstatus* is INACTIVE, HIS reads from this SMF in memory object when a WICPATH is set.

***inmem***

The SMF in memory object that HIS is connected to. When '*\*NONE\**', HIS is not connected to an in memory object.

**WICPATH= *wicpath***

Displays the z/OS UNIX Systems Services path HIS is writing or writes WIC files to. When '*\*NONE\**' HIS is not writing files to the z/OS UNIX Systems Services and *wicstatus* is INACTIVE. If *wicpath* is not '*\*NONE\**', and *wicstatus* is INACTIVE, HIS writes WIC files to this path when an INMEM is set.

***wicpath***

The a z/OS UNIX Systems Services path that HIS writes WIC files to. When '*\*NONE\**' HIS is not currently writing files to a z/OS UNIX Systems Services path.

**WIC NOT SUPPORTED ON THIS SYSTEM**

D HIS,WIC is invoked on a system that does not support WIC processing.

## System action

The system continues processing.

## Operator response

None.

## System programmer response

None.

## Problem determination

None.

## Source

Hardware instrumentation services (HIS)

## Module

HISODSP

## Routing code

-

## Descriptor code

5

---

**HIS037I**

**MODIFY *hisproc* COMMAND FAILED. SPECIFIED INMEM  
INACCESSIBLE. IFAMCON SERVICE FAILED WITH RC=*returncode*  
RSN=*reasoncode*. WIC PARAMETER(S) ACCEPTED.**

## Explanation

The MODIFY *hisproc* command fails. The specified INMEM resource is not accessible.

In the message text:

### ***hisproc***

The name of the HIS cataloged startup procedure.

### ***returncode***

The hexadecimal return code from IFAMCON.

### ***reasoncode***

The hexadecimal reason code from IFAMCON.

## System action

The system does not complete the command issued.

## Operator response

Notify the system programmer.



## System programmer response

Consult the return and reason codes to determine the cause of failure.

See *z/OS MVS Programming: Callable Services for High-Level Languages*, [SMF services](#), sections [SMF real-time interface](#), "IFAMCON", for reason code details.

For reason code x'0803' IFAINMNoConnections, there are no more connections available to INMEM resources. Issue DISPLAY SMF,M to see connected in memory resources. Either remove some connections so HIS can connect, or search problem reporting databases for a fix for the problem. If no fix exists, contact the IBM Support Center.

For reason code x'0807' IFAINMNoSuchResource, the specified SMF INMEM resource does not exist. Specify an existing INMEM resource. Define an INMEM resource for WIC records and specify that resource. It is also possible that the resource exists, but the caller doesn't have access to it.

For reason code x'0C02' IFAINMSMFNotActive, the SMF address space is not active. Start the SMF address space.

For other reason codes, search problem reporting databases for a fix for the problem. If no fix exists, contact the IBM Support Center.

## Problem determination

None.

## Source

Hardware Instrumentation Services (HIS)

## Module

HISICMD

## Routing code

-

## Descriptor code

5

---

<b>HIS038I</b>	<b><i>hisproc</i> WIC PARAMETER(S) ACCEPTED.</b>
----------------	--

## Explanation

WIC parameters specified in a MODIFY *hisproc* command are accepted by the system.

In the message text:

### ***hisproc***

The name of the HIS cataloged startup procedure.

## System action

The system processing continues.

## Operator response

None.

## System programmer response

None.

## Problem determination

None.

## Source

Hardware Instrumentation Services (HIS)

## Module

HISICMD

## Routing code

-

## Descriptor code

5

**HIS039I**

---

***hisproc* WIC TASK FAILED. ERROR DURING INMEM PROCESSING.  
service SERVICE FAILED WITH RC=*returncode*, RSN= *reasoncode*.**

## Explanation

The *hisproc* WIC task failed while using INMEM service *service*.

In the message text:

### ***hisproc***

The name of the HIS cataloged startup procedure.

### ***service***

The name of the failing service, is one of the following:

IFAMDSC

IFAMGET

### ***returncode***

The hexadecimal return code from the *service*.

### ***reasoncode***

The hexadecimal reason code from the *service*.

## System action

The system continues processing.

## Operator response

Notify the system programmer.

## System programmer response

Consult the return and reason codes to determine the cause of failure.

See SMF services, sections of SMF real-time interface, in [z/OS MVS Programming: Callable Services for High-Level Languages](#) for reason code details.

For reason code x'0C02' IFAINMSMFNotActive, the SMF address space is not active. Start the SMF address space.

For other reason codes, search problem reporting databases for a fix for the problem. If no fix exists, contact the IBM Support Center.

**Problem determination**

None.

**Source**

Hardware Instrumentation Services (HIS)

**Module**

HISPWIC

**Routing code**

2

**Descriptor code**

4

---

<b>HIS040I</b>	<b><i>hisproc</i> WIC TASK FAILED BECAUSE SMF INMEM RESOURCE <i>inmem</i> WAS REMOVED FROM THE SMF CONFIGURATION DUE TO: <i>reason</i> ACTION: <i>action</i>.</b>
----------------	---

**Explanation**

The *hisproc* WIC task failed while using INMEM *inmem* because *hisproc* is removed from the SMF configuration. In the message text:

***hisproc***  
The name of the HIS cataloged startup procedure.

***inmem***  
The name of the SMF INMEM resource.

***reason***  
Is one of the following:

- SMF PARAMETER CHANGE - SMF parameters changed to not specify the *inmem*.
- SMF WAS RESTARTED - The SMF AS is force terminated and restarted.
- SMF WAS FORCED - The SMF AS is force terminated.

***action***  
Is one of the following:

- WAIT FOR SMF TO RESTART THEN SPECIFY AN INMEM TO CONTINUE WIC PROCESSING
- SPECIFY AN INMEM TO CONTINUE WIC PROCESSING

**System action**

The system continues processing.

**Operator response**

Notify the system programmer.

## System programmer response

If necessary, wait for the SMF address space to restart. When a new SMF INMEM resource is configured to collect WIC records, use MODIFY *hisproc* to resume WIC collection, pointing to the new INMEM resource.

## Problem determination

None.

## Source

Hardware Instrumentation Services (HIS)

## Module

HISPWIC

## Routing code

2

## Descriptor code

4

---

<b>HIS041I</b>	<b><i>hisproc</i> WIC TASK ENCOUNTERED A LOSS OF DATA IN INMEM RESOURCE <i>inmem</i>.</b>
----------------	---

## Explanation

The *hisproc* WIC task found a loss of data condition while using INMEM *inmem*. There is more data that is written in the last minute than the INMEM object holds.

In the message text:

### ***hisproc***

The name of the HIS cataloged startup procedure.

### ***inmem***

The name of the SMF INMEM resource.

## System action

The system continues processing.

## Operator response

Notify the system programmer.

## System programmer response

Accept the condition or consider increasing the size of the INMEM object, if possible. Consider putting only WIC SMF type 98 records in the INMEM object.

## Problem determination

None.

## Source

Hardware Instrumentation Services (HIS)

## Module

HISPWIC

## Routing code

2

## Descriptor code

4

---

**HIS042I**

*hisproc* WIC SERVICES NOT AVAILABLE.

## Explanation

WIC parameters that are specified in a MODIFY *hisproc* command are not accepted by the system because WIC services are not available on this machine.

In the message text:

### *hisproc*

The name of the HIS cataloged startup procedure.

## System action

The system continues processing.

## Operator response

None.

## System programmer response

None.

## Problem determination

None.

## Source

Hardware Instrumentation Services (HIS)

## Module

HISINIT

## Routing code

-

## Descriptor code

5

---

**HIS043I**

**ERROR PROCESSING DIRECTORY: *directoryname. servicename* FAILED  
WITH RC= *returncode*, RS= *reasoncode*.**

## Explanation

The system detected an error when processing the identified z/OS UNIX System Services directory. Hardware instrumentation services issued a service, but the service fails with a return code and a reason code.

In the message text:

***directoryname***

The name of a z/OS UNIX System Services directory.

***servicename***

The name of the service that failed.

***returncode***

Return code that is issued with the failed service.

***reasoncode***

Reason code that is issued with the failed service.

## System action

The system continues processing.

## Operator response

If the failed service is a BPX4xxx service:

For more information about the failed service, see [\*z/OS UNIX System Services Programming: Assembler Callable Services Reference\*](#).

For the meanings and actions for the return code and reason code that were issued with the failed service, see [\*z/OS UNIX System Services Messages and Codes\*](#).

If the directory no longer exists and RC=81, the directory is deleted by the installation and this reason can be ignored.

Otherwise, search the problem reporting databases for a fix for the problem. If no fix exists, contact the IBM® Support Center.

## System programmer response

See the error or dump messages from the failed service, if any.

## Problem determination

None.

## Source

Hardware Instrumentation Services (HIS)

## Module

HISPWIC

## Routing code

1,2

## Descriptor code

4

---

**IFA714I****hh.mm.ss SMF STATUS**

*text*

## Explanation

In response to the DISPLAY SMF (D SMF) command, or when SMF logstream recording is activated as a result of a SET SMF=xx, or following SMF start up during IPL processing, the system displays the information about the SMF logstreams. In this situation, *text* has the following format:

LOGSTREAM NAME	BUFFERS	STATUS
<i>s-logstream name</i>	<i>nbytes</i>	<i>lstatus</i>

In response to the DISPLAY SMF,M (D SMF,M) command, the system displays information about connections to each in-memory resource. In this situation, *text* has the following format:

```
IN MEMORY CONNECTIONS
Resource: IFASMF.rname
Con#: ccc Connect Time: yyyy.ddd hh:mm:ss
ASID: asid
```

In response to the DISPLAY SMF STATUS (D SMF,S) command, the system displays the information about SMF data sets and log streams status. In this situation, *text* has the following format:

LOGSTREAM NAME	BUFFERS	STATUS
<i>s-logstream name</i>	<i>nbytes</i>	<i>lstatus</i>

NAME	VOLSER	SIZE(BLKS)	%FULL	STATUS
<i>dsname volser</i>	<i>blk</i>	<i>per</i>	<i>status</i>	
<i>dsname volser</i>	<i>blk</i>	<i>per</i>	<i>status</i>	

In response to the DISPLAY SMF WIC (D SMF,WIC) command, the system displays the information about SMF z/OS Workload Interaction Correlator. In this situation, *text* has the following format:

```
SMF WIC STATUS
SPECIFIED SMF PARAMETER: wicParmStatus
WorkloadIntCorr PRODUCT FEATURE: wicFeatureStatus
ST   K PG # AS  ROUTINE  VERSION  R A E LAST ROUTINE CALL TIME
-----
St   k pg NumAs ExitRtn  ExitVer  R A E lastCallTime
...

KEY:
ST      - SUBTYPE NUMBER
K       - BUFFERKEY VALUE
PG      - NUM4KPAGES VALUE
# AS    - NUMBER OF ADDRESS SPACES REGISTERED
ROUTINE - EXIT ROUTINE NAME (CURRENT / PENDING)
VERSION - EXIT ROUTINE VERSION (IN HEXADECIMAL)
R       - SMF PARAMETERS REQUEST SUBTYPE
A       - EXIT ROUTINE TO BE CALLED NEXT INTERVAL
E       - ERROR ADDING EXIT ROUTINE
LAST... - TIME EXIT ROUTINE LAST CALLED
```

On systems that do not support WIC, the following display is seen:

```
SMF WIC STATUS
WIC NOT SUPPORTED ON THIS SYSTEM
```

In the message text:

### **hh.mm.ss**

The time in hours (00-23), minutes (00-59), and seconds (00-59)

### **s**

One of the following indicators:

### **A**

The system is actively using (or attempting to use) this log stream.

## C

The system is no longer using this log stream, but is still cleaning up after removing this log stream from use.

### **logstream name**

The name of the log stream being displayed.

### **nbytes**

The amount of data collected into a log stream buffer in bytes

### **lstatus**

The status for the log stream, which can be one of the following:

#### **Connected**

Indicates that SMF is connected to the log stream

#### **Disconnected**

Indicates that the log stream is disconnected from SMF

### **rname**

The resource name of the in-memory resource.

### **ccc**

The connection number of the connection to this in-memory resource.

### **yyyy.ddd hh:mm:ss**

The Julian date and time of the connection to this in-memory resource.

### **asid**

The address space ID that has a connection to this in-memory resource.

### **dsname**

The name of the SMF data set being displayed.

### **volser**

The volume serial number of the volume containing the SMF data set.

### **blk**

The block size of a data set in blocks.

### **per**

The percent full value for the SMF data set.

### **status**

The status of the SMF data set, which can be one of the following:

#### **ACTIVE**

Indicates that the data set is now being used to record SMF data.

#### **ALTERNATE**

Indicates that the data set is available for use in recording SMF data.

#### **DUMP REQUIRED**

Indicates that the data set must be dumped before being used to record data.

### **wicParmStatus**

WIC | NOWIC - Value specified for SMFPRMxx parameter.

### **wicFeatureStatus**

Is one of the following:

#### **ENABLED**

The product registration policy (IFAPRDxx) for z/OS feature *WorkloadIntCorr* is enabled on the system. When SMF notices the feature is enabled, SMF assumes it remains enabled for the life of the IPL. SMF redrives checking if the feature is enabled at the top of the day and during certain SMF parameter updates. Perform a SETSMF WIC operation to cause SMF to recognize a product registration policy change to enable the *WorkloadIntCorr* feature.

#### **DISABLED**

The product registration policy for the z/OS feature *WorkloadIntCorr* is not recognized as enabled.



**St**

The subtype number for this status line.

**K**

The key of the shared buffer for this subtype. Can be 0, 8 or blank. A blank indicates a system subtype which does not have a shared buffer.

**pg**

The number of BUFFER4KPAGES specified on the IFAWIC request for this subtype. Blanks indicate a system subtype which does not have a shared buffer.

**NumAs**

Number of address spaces registered to collect data for the subtype number. Blanks indicate a system subtype that does not have a shared buffer.

**ExitRtn**

The name of the IFAWIC exit routine that processes shared buffers and generates SMF type 98 records.

When *st* is set to '-', *ExitRtn* and *ExitVer* are not blanks, and the remaining fields are blank, *ExitRtn* represents the new exit routine name that is switched to as a result of an IFAWIC REQUEST=REGISTER. This exit routine is added during the next interval.

**ExitVer**

The version of the IFAWIC exit routine. Blanks indicate a system subtype that does not have a version.

When *st* is set to '-', *ExitRtn* and *ExitVer* are not blanks, and the remaining fields are blank. *ExitVer* represents the new exit routine version of the *ExitRtn* that is switched to as a result of an IFAWIC REQUEST=REGISTER. This exit routine is added during the next interval.

**R**

An indication of the SMF SYS or SUBSYS TYPE parameters specified to collect this subtype. When Y, subtype is requested to be collected. When N, subtype is not requested.

**A**

An indication as to whether the subtype's IFAWIC exit routine is called on the next 5 second interval. Value is set to Y when SMF parameter WIC is specified, SMF parameters indicate to collect the subtype, the *WorkloadIntCorr* product feature is enabled, and the IFAWIC exploiter is registered for this subtype. If any of these are not true, value is set to N.

Set to blank for subtype 1, which belongs to z/OS supervisor and is run every 5 seconds to gather data needed by other IFAWIC exit routines.

**E**

Set to Y if there is an error adding the exit routine, otherwise set to N.

**lastCallTime**

Last time the exit routine is called in *mm/dd/yyyy hh:mm:ss.nnn* format.

**System action**

The system continues processing

**Operator response**

None.

**System programmer response**

None.

**Source**

System management facilities (SMF)

**Module**

IFALSMOD

**Routing code**

\*,2,10

**Descriptor code**

4, 5

---

## Chapter 5. Reference for application programmers

The following topics are of particular interest to application programmers:

- MVS authorized assembler service: [“IFAWIC — IBM z/OS Workload Interaction Correlator”](#) on page 79
- MVS data areas: [“MVS Data Areas changes for IBM z/OS Workload Interaction Correlator”](#) on page 94

---

### IFAWIC — IBM z/OS Workload Interaction Correlator

---

IBM z/OS Workload Interaction Correlator provides infrastructure to enable a z/OS component, middleware, or application product distributed across multiple address spaces to generate high-frequency (every 5 seconds), standardized, synchronized, contextualized activity using a common context. IBM z/OS Workload Interaction Correlator may be referred to herein as WIC.

z/OS Workload Interaction Correlator may be referred to herein as WIC.

#### Description

The IFAWIC service manages a z/OS component, middleware, or application as a WIC exploiter with the caller’s primary address space and the system. The IFAWIC caller identifies itself by providing the IBM-assigned subtype.

The REGISTER request returns a buffer to the caller to use for instrumenting activities for the calling primary address space. The REGISTER request must be invoked in each address space in which a program runs. This supplies each address space with a distinct buffer for instrumenting address space activities.

The IFAWIC caller provides an exit routine that SMF will call during the WIC processing. This exit routine aggregates and summarizes activities from all address spaces that the exploiter registered. The exit routine also prepares and writes an SMF type 98 record for an exploiter's subtype.

The Deregister request indicates that the program is no longer using the instrumentation buffer in the calling primary address space. When a program has been deregistered from all address spaces, the WIC exit routine will no longer be called and the system will clean it up.

Deregistration occurs implicitly at address space or job-step task termination.

#### Environment

The requirements for the caller are:

Environmental factor	Requirement
<b>Minimum authorization:</b>	The caller must be authorized with any of the following attributes: supervisor state, PKM 0-7, PSW key 0-7, or APF-authorized
<b>Dispatchable unit mode:</b>	Task
<b>Cross memory mode:</b>	PASN=HASN, any SASN
<b>AMODE:</b>	31- or 64-bit
<b>ASC mode:</b>	Primary or access register (AR)
<b>Interrupt status:</b>	Enabled for I/O and external interrupts
<b>Locks:</b>	No locks may be held.

Environmental factor	Requirement
<b>Control parameters:</b>	Control parameters must be in the primary address space or, for AR-mode callers, must be in an address space or data space that is addressable through a public entry on the caller's dispatchable unit access list (DU-AL). The control parameters must be accessible using the PSW key of the program making the request.

## Programming requirements

The caller can include the IFAWICCB mapping macro to establish equate symbols for the return codes and reason codes provided by the IFAWIC macro.

The caller can listen for ENF event 85 for notifications about their subtype. See the IFAWICCB macro for details. The WicEnfQual section describes the QUAL input to the ENFREQ request. Use WicEnfQual\_Subtype to get events for the exploiter's subtype only. WicEnf provides a mapping of the parameter list that is passed to the ENF listener exit in the first word of a 6-word parameter list in R1.

The first IFAWIC REQUEST=REGISTER caller on the system for a subtype must specify the ExitVersion and ExitRoutine parameters. This will establish the WIC exit routine to the program.

See IBM z/OS Workload Interaction Correlator in z/OS MVS Programming: Authorized Assembler Services Guide for details about how to code a WIC exit routine and how to use the WIC exit routine services.

## Restrictions

Callers specifying REQUEST=REGISTER must not have functional recovery routines (FRRs).

## Input register information

Before issuing the IFAWIC macro, the caller does not have to place any information into any register unless using it in register notation for a particular parameter or using it as a base register.

## Output register information

When control returns to the caller, the GPRs contain:

### Register

#### Contents

**0**

Reason code, if register 15 contains a non-zero return code

**1**

Used as a work register by the system

**2 - 13**

Unchanged

**14**

Used as a work register by the system

**15**

Return code

When control returns to the caller, the ARs contain:

### Register

#### Contents

**0 - 1**

Used as work registers by the system

**2 - 13**

Unchanged

## 14 - 15

Used as work registers by the system.

Some callers depend on register contents remaining the same before and after issuing a service. If the system changes the contents of registers on which the caller depends, the caller must save them before issuing the service, and restore them after the system returns control.

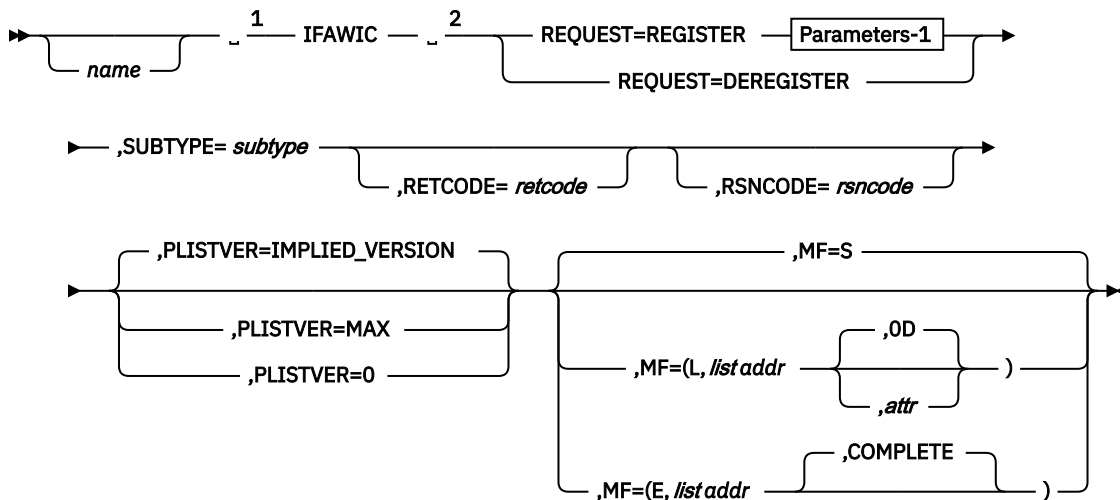
## Performance implications

None.

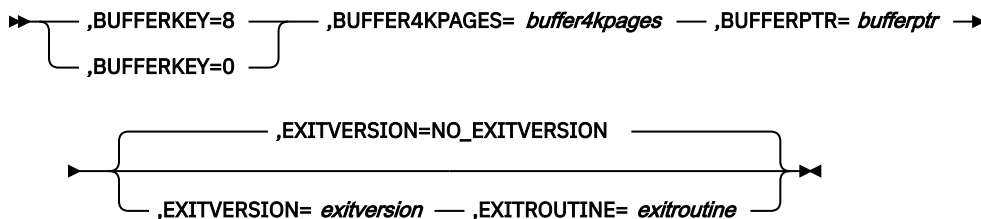
## Syntax

The IFAWIC macro is written as follows:

### Main diagram



### Parameters-1



Notes:

- <sup>1</sup> One or more blanks must precede the IFAWIC keyword.
- <sup>2</sup> One or more blanks must follow the IFAWIC keyword.

## Parameters

The IFAWIC parameters are explained as follows:

### name

An optional symbol, starting in column 1, that is the name on the IFAWIC macro invocation. The name must conform to the rules for an ordinary assembler language symbol.

### REQUEST=REGISTER

### REQUEST=DEREGISTER

A required parameter that specifies the type of IFAWIC request.

**REQUEST=REGISTER**

Indicates a request to register the address space to the WIC instrumentation for the input subtype and to register the input subtype with the system.

When registering the address space for a subtype, the caller provides the desired characteristics of the WIC instrumentation buffer in the BUFFERKEY and BUFFER4KPAGES input fields. The buffer key must be the same across all register requests for the subtype. The caller is provided with a buffer addressable by the BUFFERPTR output field. The provided buffer can be used by the caller to save instrumentation data for the registering program for activity in calling address space.

The first IFAWIC call for a subtype must register the subtype with the system and provide a WIC exit routine by specifying the EXITVERSION and EXITROUTINE fields. Subsequent REGISTER requests may update the exit routine by specifying a higher exit version number.

**REQUEST=Deregister**

Indicates that a program is no longer instrumenting into the buffer for the input subtype in the current primary address space.

The WIC instrumentation buffer for this subtype in this address space will become inaccessible, and the caller must have already stopped referencing this storage. The WIC instrumentation buffer for this address space will still be accessible to the WIC exit in the SMF address space, so ensuring the storage is accessible to the exit is not a concern to the caller.

When all address spaces registered for a program's subtype have deregistered, the system no longer calls the WIC exit routine. The WIC exit routine is cleaned up.

Deregistration occurs implicitly at address space or job-step task termination.

**,BUFFERKEY=8****,BUFFERKEY=0**

When REQUEST=REGISTER is specified, a required input parameter that specifies the storage key of the WIC instrumentation buffer to obtain. The caller must specify the same storage key for all register requests for the same subtype.

**,BUFFERKEY=8**

Indicates to obtain the instrumentation buffer in key 8.

**,BUFFERKEY=0**

Indicates to obtain the instrumentation buffer in key 0.

**,BUFFER4KPAGES=*buffer4kpages***

When REQUEST=REGISTER is specified, a required input parameter that specifies the number of 4K pages in the WIC instrumentation buffer. This number must be 1 - 16 pages.

It is desirable for the BUFFER4KPAGES value to be the same for all register requests for the input subtype; however, the system allows different address spaces to request different buffer sizes for the same input subtype. When specifying different BUFFER4KPAGES values, the WIC exit routine must be able to determine how much data can be processed in each address space's WIC instrumentation buffer. Also, when aggregating data, new data fields may not be available for aggregation from older program versions. Generally, an old exit must be able to process data it is aware of from an old or new buffer, while a new exit must be able to process all data for all buffers.

**To code:** Specify the RS-type address, or address in register (2) - (12), of a fullword field, or specify a literal decimal value.

**,BUFFERPTR=*bufferptr***

When REQUEST=REGISTER is specified, a required output parameter that points to the page-aligned WIC instrumentation buffer for the input subtype for the current primary address space. This buffer must be accessed in AMODE 64.

**To code:** Specify the RS-type address, or address in register (2) - (12), of an eight-byte pointer field.

**,EXITVERSION=exitversion**

**,EXITVERSION=NO\_EXITVERSION**

When REQUEST=REGISTER is specified, an optional input parameter that specifies the exit version number. When specified, the exit version must be an unsigned number greater than zero. The default is NO\_EXITVERSION.

For the first REQUEST=REGISTER request or subsequent REQUEST=REGISTER requests where the exit version number is larger than previously requested for the input subtype, IFAWIC processing saves the input exit routine (*exitroutine*) and calls the exit routine on the next WIC recording interval.

When the EXITVERSION parameter is not specified or when EXITVERSION specifies a version number that is less than previously requested, IFAWIC processing ignores the EXITROUTINE parameter and continues to call the exit routine associated with the largest exit version number.

When the exit version number is equal to a previously specified request, IFAWIC processing validates that the specified exit routine is the same as the previously requested exit routine. If the exit routines are different, the IFAWIC request fails and the system continues to call the last successfully added exit routine.

In order to tell the system to use a new exit routine with the same or different name, specify a higher exit version number.

**To code:** Specify the RS-type address, or address in register (2) - (12), of a fullword field, or specify a literal decimal value.

**,EXITROUTINE=exitroutine**

When REQUEST=REGISTER and EXITVERSION=exitversion are specified, a required input parameter that specifies the name of the WIC exit routine to be given control in the SMF address space to generate SMF type 98 records for the input subtype.

The specified exit routine must be a valid module name residing in the calling program's TASKLIB, STEPLIB, or JOBLIB, the system's LNKLIST concatenation, or LPA. The provided exit routine must be in an APF-authorized library and be re-entrant.

The exit routine must be in AMODE 31 or AMODE 64 and must access the WIC instrumentation buffers in AMODE 64.

**To code:** Specify the RS-type address, or address in register (2) - (12), of an 8-character field.

**,SUBTYPE=subtype**

A required input parameter that specifies the IBM-assigned subtype for writing SMF 98 records from WIC instrumentation data. The value must be a supported subtype and an unsigned number in the range 2 - 32767.

**To code:** Specify the RS-type address, or address in register (2) - (12), of a halfword field, or specify a literal decimal value.

**,RETCODE=retcode**

An optional output parameter into which the return code is to be copied from GPR 15. If you specify 15, GPR15, REG15, or R15 (within or without parentheses), the value will be left in GPR 15.

**To code:** Specify the RS-type address of a fullword field, or register (2) - (12) or (15), (GPR15), (REG15), or (R15).

**,RSNCODE=rsncode**

An optional output parameter into which the reason code is to be copied from GPR 0. If you specify 0, 00, GPR0, GPR00, REG0, REG00, or R0 (within or without parentheses), the value will be left in GPR 0.

**To code:** Specify the RS-type address of a fullword field, or register (0) or (2) - (12), (00), (GPR0), (GPR00), REG0, (REG00), or (R0).

**,PLISTVER=IMPLIED\_VERSION**

**,PLISTVER=MAX**

**,PLISTVER=0**

An optional input parameter that specifies the version of the macro. PLISTVER determines which parameter list the system generates. PLISTVER is an optional input parameter on all forms of the

macro, including the list form. When using PLISTVER, specify it on all macro forms used for a request and with the same value on all of the macro forms. The values are:

- **IMPLIED\_VERSION**, which is the lowest version that allows all parameters specified on the request to be processed. If you omit the PLISTVER parameter, IMPLIED\_VERSION is the default.
- **MAX**, if you want the parameter list to be the largest size currently possible. This size might grow from release to release and affect the amount of storage that your program needs.

If you can tolerate the size change, IBM recommends that you always specify PLISTVER=MAX on the list form of the macro. Specifying MAX ensures that the list-form parameter list is always long enough to hold all the parameters you might specify on the execute form, when both are assembled with the same level of the system. In this way, MAX ensures that the parameter list does not overwrite nearby storage.

- **0**, if you use the currently available parameters.

**To code:** Specify one of the following:

- IMPLIED\_VERSION
- MAX
- A decimal value of 0

**,MF=S**

**,MF=(L,list addr)**

**,MF=(L,list addr,attr)**

**,MF=(L,list addr,0D)**

**,MF=(E,list addr)**

**,MF=(E,list addr,COMPLETE)**

An optional input parameter that specifies the macro form.

Use MF=S to specify the standard form of the macro, which builds an inline parameter list and generates the macro invocation to transfer control to the service. MF=S is the default.

Use MF=L to specify the list form of the macro. Use the list form together with the execute form of the macro for applications that require reentrant code. The list form defines an area of storage that the execute form uses to store the parameters. Only the PLISTVER parameter may be coded with the list form of the macro.

Use MF=E to specify the execute form of the macro. Use the execute form together with the list form of the macro for applications that require reentrant code. The execute form of the macro stores the parameters into the storage area defined by the list form, and generates the macro invocation to transfer control to the service.

**,list addr**

The name of a storage area to contain the parameters. For MF=S and MF=E, this can be an RS-type address or an address in register (1) - (12).

**,attr**

An optional 1- to 60-character input string that you use to force boundary alignment of the parameter list. Use a value of 0F to force the parameter list to a word boundary, or 0D to force the parameter list to a doubleword boundary. If you do not code *attr*, the system provides a value of 0D.

**,COMPLETE**

Specifies that the system is to check for required parameters and supply defaults for omitted optional parameters.

## ABEND codes

None.



## Return and reason codes

When the IFAWIC macro returns control to your program:

- GPR 15 (and *retcode*, when you code RETCODE) contains a return code.
- When the value in GPR 15 is not zero, GPR 0 (and *rsncode*, when you code RSNCODE) contains a reason code.

The IFAWICCB mapping macro provides equate symbols for the return and reason codes. Bits 0 - 15 of the reason code may contain component diagnostic data and must not be assumed to be 0. Logically AND the reason code with the IFAWIC\_Rsncode\_Mask mask in order to isolate the non-component diagnostic portion of the reason code.

The following table identifies the hexadecimal return and reason codes and the equate symbol associated with each reason code. IBM support personnel may request the entire reason code, including the xxxx value.

Table 10. Return and reason codes for the IFAWIC service		
Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
0	–	<p><b>Equate symbol:</b> IFAWIC_Rc_Success</p> <p><b>Meaning:</b> IFAWIC request successful.</p> <p>For REQUEST=REGISTER, a BufferPtr has been provided to the current primary address space.</p> <p>If an ExitVersion has been specified, and this is the first REQUEST=REGISTER on the system for the input subtype, or the input ExitVersion is larger than the current ExitVersion the input subtype, the ExitRoutine has been registered, and the system will switch to calling the new exit routine at the next WIC interval.</p> <p>The system issues ENF 85 signals to inform IFAWIC callers to start or stop writing instrumentation data to their instrumentation buffer. See macro IFAWICCB for additional information.</p> <p>For REQUEST=Deregister, the buffer is made unavailable to the current primary address. By invoking REQUEST=Deregister, the program certifies that the exploiter has stopped instrumenting into its buffer. The program does not have to certify its WIC exit is not accessing the buffer. Once REQUEST=Deregister is invoked, the system is responsible for preventing the WIC exit from accessing the buffer</p> <p>If there are no more address spaces registered for the input subtype, the system will stop calling the exit routine.</p> <p><b>Action:</b> For REQUEST=REGISTER, the program should begin instrumenting.</p> <p>For REQUEST=Deregister, none.</p>

Table 10. Return and reason codes for the IFAWIC service (continued)

Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
4	–	<p><b>Equate symbol:</b> IFAWIC_Rc_Warning</p> <p><b>Meaning:</b> IFAWIC request completed with a warning.</p> <p>The high half word of the reason code may contain indications of other warnings that have occurred besides the reason code portion in the lower half. See Wic_Warning_Rsn in IFAWICCB for a mapping of this area.</p> <p>For REQUEST=REGISTER, a BufferPtr has been provided to the current primary address space.</p> <p>If an ExitVersion has been specified, and this is the first REQUEST=REGISTER on the system for the input subtype, or the input ExitVersion is larger than the current ExitVersion the input subtype, the ExitRoutine has been registered, and the system will switch to calling the new exit routine at the next WIC interval.</p> <p>For REQUEST=Deregister, the buffer is made unavailable to the current primary address.</p> <p>If there are no more address spaces registered for the input subtype, the system will stop calling the exit routine.</p> <p><b>Action:</b> Refer to the action provided with the specific reason code.</p> <p>For REQUEST=REGISTER, the program should check the high half of the reason code for any conditions indicating the WIC services are not fully available by testing for any bits being on in the first byte using the IFAWICCB field Wic_NotFullyAvail_LowHighMask.</p> <p>When no bits in the first byte of the reason code are on, the program should start writing instrumentation data to the instrumentation buffer because the program's WIC exit will be called at the next WIC interval.</p> <p>When any bit in the first byte of the reason code is on, the program should not start writing instrumentation data to the instrumentation buffer because the WIC exit will not be called at the next WIC interval</p> <p>The system issues ENF 85 signals to inform IFAWIC callers to start or stop writing instrumentation data to their instrumentation buffer. See macro IFAWICCB for additional information.</p> <p>For REQUEST=Deregister, none.</p>

Table 10. Return and reason codes for the IFAWIC service (continued)

Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
4	xxxx0401	<p><b>Equate symbol:</b> IFAWIC_Rsn_Buffer4kPagesSmaller</p> <p><b>Meaning:</b> A smaller Buffer4kPages value was requested than the value specified on a previous REQUEST=REGISTER for the same subtype.</p> <p>The system reserves the requested buffer size for this address space.</p> <p><b>Action:</b> It is desirable for the BUFFER4KPAGES value to be the same for all register requests for the input subtype; however, the system allows different address spaces to request different buffer sizes for the same input subtype. When specifying different BUFFER4KPAGES values, the WIC exit routine must be able to determine how much data can be processed in each address space's WIC instrumentation buffer. Also, when aggregating data, new data fields may not be available for aggregation from older program versions. Generally, an old exit must be able to process data it is aware of from an old or new buffer, while a new exit must be able to process all data for all buffers.</p>
4	xxxx0402	<p><b>Equate symbol:</b> IFAWIC_Rsn_Buffer4kPagesLarger</p> <p><b>Meaning:</b> A larger Buffer4kPages value was requested than the value specified on a previous REQUEST=REGISTER for the same subtype.</p> <p>The system reserves the requested buffer size for this address space.</p> <p><b>Action:</b> It is desirable for the BUFFER4KPAGES value to be the same for all register requests for the input subtype; however, the system allows different address spaces to request different buffer sizes for the same input subtype. When specifying different BUFFER4KPAGES values, the WIC exit routine must be able to determine how much data can be processed in each address space's WIC instrumentation buffer. Also, when aggregating data, new data fields may not be available for aggregation from older program versions. Generally, an old exit must be able to process data it is aware of from an old or new buffer, while a new exit must be able to process all data for all buffers.</p>
4	xxxx0403	<p><b>Equate symbol:</b> IFAWIC_Rsn_AlreadyRegistered</p> <p><b>Meaning:</b> A REQUEST=REGISTER was requested from an address space for a subtype that was already registered.</p> <p><b>Action:</b> none needed.</p>
4	xxxx0404	<p><b>Equate symbol:</b> IFAWIC_Rsn_NotRegistered</p> <p><b>Meaning:</b> A REQUEST=DEREGISTER was requested from an address space for a subtype that was not currently registered.</p> <p><b>Action:</b> none needed.</p>

Table 10. Return and reason codes for the IFAWIC service (continued)		
Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
4	xxxx0405	<p><b>Equate symbol:</b> IFAWIC_Rsn_ExitVersionIgnored</p> <p><b>Meaning:</b> A REQUEST=REGISTER specified an ExitVersion that was less than the current ExitVersion. The ExitVersion and ExitRoutine parameters are ignored and the system continues using the current highest versioned exit routine.</p> <p><b>Action:</b> none needed.</p>
4	xxxx0481	<p><b>Equate symbol:</b> IFAWIC_Rsn_WicFeatureNotEnabled</p> <p><b>Meaning:</b> The WorkloadIntCorr feature was not enabled to product registry.</p> <p><b>Action:</b> Wait for ENF 85 signal and verify the WicEnf bit WicEnf_InstrumentationRequested is on before writing to the instrumentation buffer.</p>
4	xxxx0482	<p><b>Equate symbol:</b> IFAWIC_Rsn_SubtypeNotCollected</p> <p><b>Meaning:</b> SMF has not been configured to collect the input SMF 98 subtype record.</p> <p><b>Action:</b> Wait for ENF85 signal and verify the WicEnf bit WicEnf_InstrumentationRequested is on before writing to the instrumentation buffer.</p>
8	–	<p><b>Equate symbol:</b> IFAWIC_Rc_UserError</p> <p><b>Meaning:</b> The IFAWIC request specified parameters that are not valid or the request is issued in an user-controllable environment that is not valid.</p> <p>The IFAWIC service did not complete successfully. For REQUEST=REGISTER, no WIC instrumentation buffer was provided to the caller.</p> <p><b>Action:</b> Refer to the action provided with the specific reason code.</p>
8	xxxx0801	<p><b>Equate symbol:</b> IFAWIC_Rsn_ParmListAlet</p> <p><b>Meaning:</b> Unable to use ALET of the IFAWIC parameter list.</p> <p><b>Action:</b> Provide a valid ALET for the parameter list. The access register might not have been set up correctly.</p>
8	xxxx0802	<p><b>Equate symbol:</b> IFAWIC_Rsn_BadVersion</p> <p><b>Meaning:</b> The provided PLISTVER is not valid. This suggests the parameter list used to call IFAWIC was overlaid, or was not generated by the IFAWIC macro.</p> <p><b>Action:</b> Check for possible storage overlay of the parameter list. Use the IFAWIC service to produce the parameter list.</p>

Table 10. Return and reason codes for the IFAWIC service (continued)		
Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
8	xxxx0803	<b>Equate symbol:</b> IFAWIC_Rsn_ReservedNot0 <b>Meaning:</b> The parameter list contains non-0 input in reserved fields. This suggests the parameter list used to call IFAWIC was overlaid, or was not generated by the IFAWIC macro. <b>Action:</b> Check for possible storage overlay of the parameter list. Use the IFAWIC service to produce the parameter list.
8	xxxx0804	<b>Equate symbol:</b> IFAWIC_Rsn_BadRequestType <b>Meaning:</b> The request type is not valid. This suggests the parameter list used to call IFAWIC was overlaid, or was not generated by the IFAWIC macro. <b>Action:</b> Check for possible storage overlay of the parameter list. Use the IFAWIC service to produce the parameter list.
8	xxxx0805	<b>Equate symbol:</b> IFAWIC_Rsn_BadBufferKey <b>Meaning:</b> The buffer key type is not valid. This suggests the parameter list used to call IFAWIC was overlaid, or was not generated by the IFAWIC macro. <b>Action:</b> Check for possible storage overlay of the parameter list. Use the IFAWIC service to produce the parameter list.
8	xxxx0807	<b>Equate symbol:</b> IFAWIC_Rsn_ParmListFetch <b>Meaning:</b> An error was encountered when fetching the supplied parameter list. <b>Action:</b> Call IFAWIC with a parameter list properly addressable.
8	xxxx0808	<b>Equate symbol:</b> IFAWIC_Rsn_ParmListWrite <b>Meaning:</b> An error was encountered when writing to the supplied parameter list. <b>Action:</b> Call IFAWIC with a parameter list properly addressable and able to be written to.
8	xxxx0810	<b>Equate symbol:</b> IFAWIC_Rsn_NotEnabled <b>Meaning:</b> The IFAWIC caller was not enabled. <b>Action:</b> Call IFAWIC only when enabled.
8	xxxx0811	<b>Equate symbol:</b> IFAWIC_Rsn_Locked <b>Meaning:</b> The IFAWIC caller was locked. <b>Action:</b> Call IFAWIC without holding locks.
8	xxxx0812	<b>Equate symbol:</b> IFAWIC_Rsn_HomeNotPrimary <b>Meaning:</b> The IFAWIC caller's primary home space was not equal to the caller's primary address space. <b>Action:</b> Call IFAWIC when home equals primary address space.

Table 10. Return and reason codes for the IFAWIC service (continued)		
Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
8	xxxx0813	<b>Equate symbol:</b> IFAWIC_Rsn_CallerFRR <b>Meaning:</b> The IFAWIC REQUEST=REGISTER caller had FRR recovery established. <b>Action:</b> Call IFAWIC without FRRs established.
8	xxxx0814	<b>Equate symbol:</b> IFAWIC_Rsn_NotAuthorized <b>Meaning:</b> The IFAWIC caller is not authorized. <b>Action:</b> Call IFAWIC only when authorized.
8	xxxx0820	<b>Equate symbol:</b> IFAWIC_Rsn_SubtypeInput <b>Meaning:</b> The provided SUBTYPE is not supported. <b>Action:</b> Supply the IBM provided program subtype.
8	xxxx0821	<b>Equate symbol:</b> IFAWIC_Rsn_ExitVersionZero <b>Meaning:</b> The ExitVersion input must not be 0. <b>Action:</b> Specify an ExitVersion that is not 0.
8	xxxx0822	<b>Equate symbol:</b> IFAWIC_Rsn_Buffer4kPagesZero <b>Meaning:</b> The Buffer4kPages must not be 0. <b>Action:</b> Specify Buffer4kPages that is not 0.
8	xxxx0823	<b>Equate symbol:</b> IFAWIC_Rsn_Buffer4kPagesTooLarge <b>Meaning:</b> The Buffer4kPages input must be less than or equal to 16. <b>Action:</b> Specify Buffer4kPages as less than or equal to 16.
8	xxxx0824	<b>Equate symbol:</b> IFAWIC_Rsn_BufferKeyMismatch <b>Meaning:</b> A previous REQUEST=REGISTER requested and received a buffer for a BUFFERKEY value that was different than the BUFFERKEY specified on this request. <b>Action:</b> Update the program to request the same BUFFERKEY for each call to IFAWIC, even across different program versions.
8	xxxx0825	<b>Equate symbol:</b> IFAWIC_Rsn_SRBMode <b>Meaning:</b> IFAWIC was issued in SRB mode <b>Action:</b> Do not issue IFAWIC in SRB mode
8	xxxx0830	<b>Equate symbol:</b> IFAWIC_Rsn_SameVerExitRoutineMismatch <b>Meaning:</b> A ExitRoutine and ExitVersion was specified such that the ExitVersion is the same as a previous specification, however, the ExitRoutine is different. <b>Action:</b> Specify the same ExitRoutine to correspond with the same ExitVersion. If the program introduces a new exit routine, the ExitVersion must be made larger.

Table 10. Return and reason codes for the IFAWIC service (continued)		
Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
8	xxxx0831	<p><b>Equate symbol:</b> IFAWIC_Rsn_ExitRtnNotFound</p> <p><b>Meaning:</b> The exit routine could not be found.</p> <p><b>Action:</b> Ensure the exit routine is located in either the invoking program's JOBLIB, STEPLIB, TASKLIB, in the system's LNKST concatenation, or in LPA.</p>
8	xxxx0832	<p><b>Equate symbol:</b> IFAWIC_Rsn_ExitRtnNotInAPFLib</p> <p><b>Meaning:</b> IFAWIC was unable to load the provided exit routine because did not reside in an APF authorized library.</p> <p>The xxxx portion of the reason code contains an informational code as documented with ABEND 306 reasons. No abend was generated.</p> <p><b>Action:</b> See <i>z/OS MVS System Codes</i>, sections "System completion codes," "306," with the xxxx portion of the reason code to see what action is needed to resolve the condition.</p> <p>Ensure the exit routine resides in an APF-authorized library and is re-entrant.</p>
8	xxxx0833	<p><b>Equate symbol:</b> IFAWIC_Rsn_ExitRtnAmode24</p> <p><b>Meaning:</b> The ExitRoutine must not be in AMODE 24.</p> <p><b>Action:</b> Provide an exit routine with an AMODE of 31 or 64.</p>
8	xxxx0834	<p><b>Equate symbol:</b> IFAWIC_Rsn_ExitRtnNotReEntrant</p> <p><b>Meaning:</b> The ExitRoutine must be re-entrant.</p> <p><b>Action:</b> Provide an exit routine that is linked as re-entrant.</p>
8	xxxx0835	<p><b>Equate symbol:</b> IFAWIC_Rsn_ExitRtnNameNotUnique</p> <p><b>Meaning:</b> Another subtype is using the same exit routine name. The exit routine name must be unique per subtype.</p> <p><b>Action:</b> Provide a unique exit routine name.</p>
C	–	<p><b>Equate symbol:</b> IFAWIC_Rc_EnvError</p> <p><b>Meaning:</b> Environmental error</p> <p>The IFAWIC service did not complete successfully. For REQUEST=REGISTER, no WIC instrumentation buffer was provided to the caller.</p> <p><b>Action:</b> Refer to the action provided with the specific reason code.</p>
C	xxxx0C01	<p><b>Equate symbol:</b> IFAWIC_Rsn_SystemNotReady</p> <p><b>Meaning:</b> IFAWIC was issued before the system is ready to start processing requests.</p> <p><b>Action:</b> Wait for the system to be ready to accept IFAWIC requests and re-issue the IFAWIC request.</p>

Table 10. Return and reason codes for the IFAWIC service (continued)

Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
C	xxxx0C02	<p><b>Equate symbol:</b> IFAWIC_Rsn_UnsupportedMachine</p> <p><b>Meaning:</b> IFAWIC was issued from an unsupported machine. WIC requires IBM z14 or later hardware.</p> <p><b>Action:</b> Issue IFAWIC on a supported machine.</p>
C	xxxx0C03	<p><b>Equate symbol:</b> IFAWIC_Rsn_EnvNoExit</p> <p><b>Meaning:</b> IFAWIC REQUEST=REGISTER requests omitting ExitVersion and ExitRoutine, require a WIC exit routine to have been already established by a previous IFAWIC REQUEST=REGISTER call specifying the ExitVersion and ExitRoutine parameters.</p> <p><b>Action:</b> Specify ExitVersion and ExitRoutine or ensure an IFAWIC REQUEST=REGISTER call has already completed specifying ExitVersion and ExitRoutine before issuing IFAWIC REQUEST=REGISTER.</p>
C	xxxx0C04	<p><b>Equate symbol:</b> IFAWIC_Rsn_SMFNOWICSpecified</p> <p><b>Meaning:</b> SMF parameters specified NOWIC which prevents programs from issuing IFAWIC REQUEST=REGISTER.</p> <p><b>Action:</b> Accept that WIC services will be unavailable for the program, or choose to handle WIC registration dynamically. If the program can support dynamic WIC registration, wait for ENF 85 signal and check the WicEnf bit WicEnf_RegisterIsAvailable. When WicEnf_RegisterIsAvailable is on, The ENF listener exit can cause a task to wake up to re-issue IFAWIC REQUEST=REGISTER for the program.</p> <p>The system issues ENF 85 signals to inform IFAWIC callers that IFAWIC Register is available when SMF parameter WIC is specified. See macro IFAWICCB for additional information.</p>
C	xxxx0C10	<p><b>Equate symbol:</b> IFAWIC_Rsn_ExitRtnNoStorage</p> <p><b>Meaning:</b> There was not sufficient storage for IFAWIC to process the exit routine.</p> <p><b>Action:</b> Contact your system programmer. there is a shortage of common storage.</p>
C	xxxx0C11	<p><b>Equate symbol:</b> IFAWIC_Rsn_UnexpectedLoadError</p> <p><b>Meaning:</b> IFAWIC encountered errors trying to load the provided exit routine.</p> <p>The "xxxx" portion of the reason code contains a code from the load service of what would have been the ABEND code. No abend was generated.</p> <p><b>Action:</b> Alert the system programmer. See <i>z/OS MVS System Codes</i>, sections "System completion codes," with the "xxxx" portion as the abend code to see what action is needed to resolve the condition. This may be accompanied by system log messages with the CSV prefix.</p>



Table 10. Return and reason codes for the IFAWIC service (continued)		
Return code (hex)	Reason code (hex)	Equate symbol, meaning, and action
10	–	<b>Equate symbol:</b> IFAWIC_Rc_CompError <b>Meaning:</b> Unexpected failure. <b>Action:</b> Refer to the action provided with the specific reason code.
10	xxxx1001	<b>Equate symbol:</b> IFAWIC_Rsn_CompError <b>Meaning:</b> Unexpected failure. The state of the request is unpredictable. <b>Action:</b> Contact your system programmer to report the problem to IBM service

## Example

This example performs the following operations:

1. Issue the IFAWIC service to start WIC instrumentation for a test subtype.
2. Issue the IFAWIC service to end WIC instrumentation for a test subtype.

The code is as follows:

```

... other code here ...

    LARL 13,Wic_Fields
    USING Wic_Fields,13
    ENFREQ ACTION=LISTEN, CODE=ENFPC085, EOM=YES,          *
        SRBEXIT=MyWicEnfExit,                             *
        QUAL=MySubtype, QMASK=(BYTE3,BYTE4),              *
        DTOKEN=MyWicEnfToken Listen for ENF 85
    IFAWIC REQUEST=REGISTER,                                *
        SUBTYPE=MySubtype,                                 *
        ExitRoutine=MyExitRtn,                             *
        ExitVersion=MyExitVer,                             *
        Buffer4kPages=MyBuf4kPg,                           *
        BUFFERKEY=8,                                       *
        BUFFERPTR=MyBuffer@ Register WIC subtype
    CFI 15,IFAWIC_Rc_Warning Check for a warning RC
    JH WicRegFailed RC>4 is an error
    JL WicRegSuccess RC=0 is a success
    TMLH 0,Wic_NotFullyAvail_LowHighMask WIC services *
        fully available?
    JZ WicRegSuccess Yes, instrument
    J WicRegDone No, do not instrument
WicRegFailed DS 0H Register RC > 4
    NI MyWicFlags,MyWicRegisterFailed remember failed
    ENFREQ ACTION=DELETE, CODE=ENFPC085,                  *
        DTOKEN=MyWicEnfToken Stop listening to the ENF
    J WicRegDone
WicRegSuccess DS 0H Register succeeded
    NI MyWicEnfFlags,WicEnf_InstrumentationRequested *
        Indicate to write
        instrumentation data
WicRegDone DS 0H
*
* Place additional return/ reason code checking here
*

... other code here ...

    TM MyWicEnfFlags,WicEnf_InstrumentationRequested *
        Instrumentation ok now?
    JZ InstrumentDone No, do not instrument

... code using the provided buffer here ...

InstrumentDone DS 0H After WIC instrumentation

... other code here ...

```

```

        TM    MyWicFlags,MyWicRegisterFailed WIC Reg fail?
        JO    WicDeregDone          Skip dereg if reg failed
        IFAWIC REQUEST=DEREGISTER,          *
              SUBTYPE=MySubtype    Deregister WIC subtype
        ENFREQ ACTION=DELETE, CODE=ENFPC085, *
              DTOKEN=MyWicEnfToken Stop listening to the ENF
*
* Place code to check return/reason codes here
*
WicDeregDone DS 0H
... other code here ...
WICENFXT DS 0H
        LARL 15,Wic_Fields          Entry to SRB
        USING Wic_Fields,15         Load the working storage ptr
        L    1,0(1)                Load WicEnf Parmlist
        USING WICENF,1
        IC    0,WicEnf_Flags        Copy the WicEnf_Flags
        STC    0,MyWicEnfFlags      Save in MyWicEnfFlags
        BR    14                   Return from SRB

... other code here ...
Wic_Fields DS 0D
MyBuffer0 DS D
MyExitRtn DC CL8'WICEXIT '
MySubtype DC F'24576'
MyExitVer DC F'1'
MyBuf4kPg DC F'4'
MyWicEnfToken DS F
MyWicEnfExit DC A(WICENFXT)
MyWicFlags DC X'00'
MyWicRegisterFailed EQU X'80'
MyWicEnfFlags DC X'00'

... other code here ...

```

## MVS Data Areas changes for IBM z/OS Workload Interaction Correlator

IBM z/OS Workload Interaction Correlator updates or adds the following MVS data areas:

- **CVT**

New bits are defined in the CVTFLAG9 field, as follows. CVTFLAG9 is a 1-byte field at CVT + X'1AD' (429).

- Bit 6 (.....1.) – CVTIFAWICAvailable – IFAWIC service is available for use.
- Bit 7 (.....1.) – CVTIFAWICInstalled – IFAWIC service is installed.

- **IFAWICCB**

IFAWICCB is a new data area.

## CVT information

### CVT programming interface information

**ONLY** the following fields are part of the programming interface information:

- CVTABEND
- CVTAMFF
- CVTASMVT
- CVTASVT
- CVTBRET
- CVTBSMOF
- CVTCSD

- CVTCTLFG
- CVTDCB
- CVTDCPA
- CVTDFA
- CVTECVT
- CVTEDAT2
- CVTEPLPS
- CVTEXIT
- CVTEXP1
- CVTFLAG2
- CVTFLAG3
- CVTFLAG4
- CVTFLAG5
- CVTFLAG6
- CVTFLAG7
- CVTFLAG9
- CVTFLGBT
- CVTGDA
- CVTGRSST
- CVTGVT
- CVTHID
- CVTIXAVL
- CVTJESCT
- CVTLCCAT
- CVTLDTO
- CVTLINK
- CVTLSO
- CVTMAXMP
- CVTMDL
- CVTMSER
- CVTOPCTP
- CVTOSLVL
- CVTOVER
- CVTPCCAT
- CVTPCNET
- CVTPRLTV
- CVTPROD
- CVTPSXM
- CVTPVTP
- CVTQTD00
- CVTQTE00
- CVTRAC
- CVTRCEP

- CVTRCZRT
- CVTRELNO
- CVTRI
- CVTRTMCT
- CVTSAF
- CVTSCPIN
- CVTSDBF
- CVTSDUMP
- CVTSMCA
- CVTSNAME
- CVTSUBSP
- CVTSVT
- CVTSYSAD
- CVTTPC
- CVTTVT
- CVTTX
- CVTTXC
- CVTTXTE
- CVTTZ
- CVTUCBSC
- CVTUNDAItVM
- CVTUNDVM
- CVTUNDzVM
- CVTUSER
- CVTVERID
- CVTVFGET
- CVTVFIND
- CVTVMEnv
- CVTVPSIB
- CVTVWAIT
- CVTZPDT
- CVT0EF00
- CVT0PT0E
- CVT0PT02
- CVT0PT03
- CVT0SCR1

## CVT heading information

<b>Common name:</b>	Communications Vector Table
<b>Macro ID:</b>	CVT
<b>DSECT name:</b>	CVT(when DSECT=YES is coded and PREFIX=YES is not coded) CVTFIX(when DSECT=YES and PREFIX=YES is coded) CVTMAP(or name user coded in label field of CVT invocation) CVTVSTGX(DSECT name of virtual storage extension) CVTXTNT1(DSECT name of OS-OS/VS common extension) CVTXTNT2(DSECT name of OS/VS1-OS/VS2 common extension)

**Owning component:** Common Macros (SC101)

**Eye-catcher ID:** CVT  
Offset: 96  
Length: 4

**Storage attributes:** Subpool: Nucleus  
Key: 0  
Residency: Below 16M line

**Size:** Prefix: 256 bytes  
CVT: 1280 bytes  
Virtual storage address extension: 80 bytes  
OS - OS/VS common extension: 12 bytes  
OS/VS1 - OS/VS2 common extension: 132 bytes

**Created by:** IEAVCVT

**Pointed to by:** FLCCVT field of the PSA data area (location X'10')  
FLCCVT2 field of the PSA data area  
CVTSMEXT points to the Virtual address storage extension  
OS/VS - OS/VS extension is pointed to by CVTEXT1  
OS/VS1 - OS/VS2 extension is pointed to by CVTEXT2

**Serialization:** Based on the individual fields being referenced.

**Function:** The CVT provides the means by which non-nucleus-resident routines may refer to information in the nucleus of the control program. It contains addresses of other control blocks and tables used by the control program routines.

## CVT mapping

Table 11. Structure CVTFIX

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
-256	(-100)	STRUCTURE	0	CVTFIX	- CVTMAP-256 - PREFIX
-256	(-100)	CHARACTER	216		- RESERVED
-40	(-28)	CHARACTER	16	CVTPROD(0)	- SYSTEM CONTROL PROGRAM PRODUCT LEVEL.
-40	(-28)	CHARACTER	8	CVTPRODN	PRODUCT NAME OF THE CONTROL PROGRAM, EX.(SP3.2). This is maintained for compatibility reasons only. The true product name version, release, and modification level information is in the ECVT (ECVTPNAM, PVER, PREL, PMOD). This can be considered a shorthand for the official name

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
For z/OS V3R2 (HBB77F0), the value is SP7.3.2 For z/OS V3R1 (HBB77E0), the value is SP7.3.1 For z/OS V2R5 (HBB77D0), the value is SP7.2.5 For z/OS V2R4 (HBB77C0), the value is SP7.2.4 For z/OS V2R3 (HBB77B0), the value is SP7.2.3 For z/OS V2R2 (HBB77A0), the value is SP7.2.2 For z/OS V2R1 (HBB7790), the value is SP7.2.1 For z/OS R13 (HBB7780), the value is SP7.1.3 For z/OS R12 (HBB7770), the value is SP7.1.2 For z/OS R11 (HBB7760), the value is SP7.1.1 For z/OS R10 (HBB7750), the value is SP7.1.0 For z/OS R9 (HBB7740), the value is SP7.0.9 For z/OS R8 (HBB7730), the value is SP7.0.8 For z/OS R7.1 (JBB772S), the value is SP7.S.7 For z/OS R7 (HBB7720), the value is SP7.0.7 For z/OS R6.1 (JBB77S9), the value is SP7.S.6 For z/OS R6 (HBB7709), the value is SP7.0.6 For z/OS R5 (HBB7708), the value is SP7.0.5 For z/OS R4 (HBB7707), the value is SP7.0.4 For z/OS R3 (HBB7706), the value is SP7.0.3 For z/OS R2 (HBB7705), the value is SP7.0.2 For z/OS R1 (JBB7713), the value is SP7.0.1 For OS/390 R10 (HBB7703), the value is SP6.1.0 For OS/390 R9 (JBB6609), the value is SP6.0.9 For OS/390 R8 (HBB6608), the value is SP6.0.8 For OS/390 R7 (JBB6607), the value is SP6.0.7 For OS/390 R6 (HBB6606), the value is SP6.0.6 For OS/390 R5 (HBB6605), the value is SP6.0.5 For OS/390 R4 (JBB6604), the value is SP6.0.4 For OS/390 R3 (HBB6603), the value is SP6.0.3 For OS/390 R2 (JBB6602), the value is SP6.0.2 For OS/390 R1 (HBB6601), the value is SP6.0.1 For MVS/ESA SP5.2.2 (JBB5522) the value is SP5.2.2 For MVS/ESA SPa.b.c the value is SPa.b.c					
-32	(-20)	CHARACTER	8	CVTPRODI	PRODUCT FMID IDENTIFIER FOR THE CONTROL PROGRAM, EX.(JBB1328).
-24	(-18)	CHARACTER	16	CVTVERID	OPTIONAL USER PERSONALIZATION OF SOFTWARE SYSTEM VERSION. (MDC415)
-8	(-8)	SIGNED	2		- RESERVED
-6	(-6)	CHARACTER	2	CVTMDL	- CPU NUMBER IN SIGNLESS PACKED DECIMAL, I.E., A 3090 PROCESSOR WOULD APPEAR AS X'3090'
-4	(-4)	CHARACTER	4	CVTRELNO(0)	- RELEASE NUMBER (EBCDIC)
-4	(-4)	CHARACTER	2	CVTNUMB	- RELEASE NUMBER YM2188
-2	(-2)	CHARACTER	2	CVTLEVL	- LEVEL NUMBER OF THIS RELEASE YM2188
END OF CVT PREFIX THE FOLLOWING LINE ESTABLISHES THE HIGHEST DEGREE OF ALIGNMENT REQUIRED IN CVT PROPER, SO THAT THE BOUNDARY ALIGNMENT IS NOT DETERMINED BY THE ENDING OFFSET OF THE PRECEDING SECTION					
0	(0)	DBL WORD	8	(0)	- ESTABLISHES ALIGNMENT
0	(0)	X'100'	0	CVTMAP	"*" - CVTPTR CONTENT POINTS HERE
0	(0)	ADDRESS	4	CVTTCBP	"V(IEATCBP)" - Address of PSATNEW.
4	(4)	ADDRESS	4	CVT0EF00	"V(IEA0EF00)" - ADDRESS OF ROUTINE TO SCHEDULE ASYNCHRONOUS EXITS
8	(8)	ADDRESS	4	CVTLINK	"V(IEFLINK)" - ADDRESS OF DCB FOR SYS1.LINKLIB DATA SET. UPDATED BY CONTENTS SUPERVISION RIM. OWNERSHIP: CONTENTS SUPERVISION.
12	(C)	ADDRESS	4	CVTAUSCB	"V(IEFAUSDM)" - ADDRESS OF ASSIGN/ UNASSIGN SERVICE DATA MODULE.
16	(10)	ADDRESS	4	CVTBUF	- ADDRESS OF THE BUFFER OF THE RESIDENT CONSOLE INTERRUPT ROUTINE
20	(14)	ADDRESS	4	CVTXAPG	"V(IECXAPG)" - ADDRESS OF I/O SUPERVISOR APPENDAGE VECTOR TABLE

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
24	(18)	ADDRESS	4	CVT0VL00	"V(IEA0VL00)" - ADDRESS OF ENTRY POINT OF THE TASK SUPERVISOR'S ADDRESS VALIDITY CHECKING ROUTINE
28	(1C)	ADDRESS	4	CVTPCNVT	"V(IEPCNVT)" - ADDRESS OF ENTRY POINT OF THE ROUTINE WHICH CONVERTS A RELATIVE TRACK ADDRESS (TTR) TO AN ABSOLUTE TRACK ADDRESS (MBBCCHHR)
32	(20)	ADDRESS	4	CVTPRLTV	"V(IECPRLTV)" - ADDRESS OF ENTRY POINT OF THE ROUTINE WHICH CONVERTS AN ABSOLUTE TRACK ADDRESS (MBBCCHHR) TO A RELATIVE TRACK ADDRESS (TTR)
36	(24)	ADDRESS	4	CVTL LCB	"V(CSVLLCB)" - ADDRESS OF THE LLCB.
40	(28)	ADDRESS	4	CVTL LTRM	"V(CSVLLTRM)" - ADDRESS OF LLA'S MEMORY TERMINATION RESOURCE MANAGER.
44	(2C)	ADDRESS	4	CVTXTLER	"V(IECXTLER)" - ADDRESS OF ERROR RECOVERY PROCEDURE (ERP) LOADER (IECVERPL) ENTRY POINT IECXTLER (MDC349)
48	(30)	ADDRESS	4	CVTSYSAD	- UCB ADDRESS FOR THE SYSTEM RESIDENCE VOLUME (MDCXXX)
52	(34)	ADDRESS	4	CVTBTERM	"V(IEAVTRG1)" - ADDRESS OF ENTRY POINT OF THE ABTERM ROUTINE @(DCR854)
56	(38)	SIGNED	4	CVTDATE	- CURRENT DATE IN PACKED DECIMAL
60	(3C)	ADDRESS	4	CVTMSLT	"V(IEEMSER)" - ADDRESS OF THE MASTER COMMON AREA IN MASTER SCHEDULER RESIDENT DATA AREA. NOTE - USE CVTMSER INSTEAD TO ADDRESS MASTER SCHEDULER RESIDENT DATA AREA
64	(40)	ADDRESS	4	CVTZDTAB	- ADDRESS OF I/O DEVICE CHARACTERISTIC TABLE
68	(44)	ADDRESS	4	CVTXITP	"V(IECXITP)" - ADDRESS OF ERROR INTERPRETER ROUTINE
72	(48)	ADDRESS	4	CVT0EF01	"V(IEA0EF01)" ENTRY POINT IN STAGE II EXIT EFFECTOR USED BY SCHEDXIT MACRO
76	(4C)	SIGNED	4	CVTVPRM(0)	VECTOR PARAMETERS
76	(4C)	SIGNED	2	CVTVSS	VECTOR SECTION SIZE
78	(4E)	SIGNED	2	CVTVPSM	VECTOR PARTIAL SUM NUMBER
80	(50)	SIGNED	2	CVTEXTIT	- An SVC 3 instruction. Exit to dispatcher. This is a programming interface for IRBs only. An IRB may return to the system by branching to this location
82	(52)	SIGNED	2	CVTBRET	- A BR 14 INSTRUCTION. RETURN TO CALLER (USED BY DATA MANAGEMENT ROUTINES)
84	(54)	ADDRESS	4	CVTSVDCB	"V(IEASVDCB)" - ADDRESS OF THE DCB FOR THE SYS1.SVCLIB DATA SET
88	(58)	ADDRESS	4	CVTTPC	"V(IEATPC)" - ADDRESS OF THE TIMER SUPERVISOR WORK AREA
92	(5C)	SIGNED	4	CVTFLGCS(0)	- Flags set by CS
92	(5C)	BITSTRING	1	CVTFLGC0	- Flags
	1... ..			CVTMULNF	"X'80'" - For users of IFAUSAGE, REQUEST=FUNCTIONxxx calls need not be done. This correlates to the NOMULCFUNC parameter in SMFPRMxx
93	(5D)	BITSTRING	1	CVTFLGC1	- Flags
94	(5E)	SIGNED	2	CVTICPID	- IPL'ED CPU PHYSICAL ID
96	(60)	CHARACTER	4	CVTCVT	- CVT ACRONYM IN EBCDIC (EYE-CATCHER)

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
100	(64)	ADDRESS	4	CVTCUCB	"V(IEECUCB)" - ADDRESS OF THE UNIT CONTROL MODULE (UCM)
104	(68)	ADDRESS	4	CVTQTE00	"V(IEAQTE00)" - ADDRESS OF THE TIMER ENQUEUE ROUTINE FOR INTERVAL TIMER
108	(6C)	ADDRESS	4	CVTQTD00	"V(IEAQTD00)" - ADDRESS OF THE TIMER DEQUEUE ROUTINE FOR INTERVAL TIMER
112	(70)	ADDRESS	4	CVTSTB	- ADDRESS OF THE I/O DEVICE STATISTICS TABLE
116	(74)	BITSTRING	1	CVTDCB	- OPERATING SYSTEM FOR S/370-XA MODE EXECUTION, CVTMVSE, CVT4MS1, CVT0SEXT, CVT6DAT, AND CVTMVS2 ARE SET ON AT CVT CREATION
		1... ....		CVTMVSE	"X'80'" - S/370-XA MODE EXECUTION
		.1.. ....		CVT1SSS	"X'40'" - OPTION 1 (PCP) SSS. ALSO, LANGUAGE COMPILERS MAY USE THIS BIT TO DETERMINE IF THEY ARE RUNNING UNDER OS OR VM (WILL BE 0 FOR OS).
		..1. ....		CVT2SPS	"X'20'" - OPTION 2 (MFT) SPS, OS/VS1, VSE
		...1 ....		CVT4MS1	"X'10'" - OPTION 4 (MVT) MS1, OS/VS2
		.... 1...		CVT0SEXT	"X'08'" - INDICATOR THAT THE CVTOSLVL AREA IS PRESENT AND MAY BE REFERENCED. This bit is on for all releases starting with MVS/SP3.1.0.
		.... .1..		CVT4MPS	"X'04'" - MODEL 65 MULTIPROCESSING
		.... ..1.		CVT6DAT	"X'02'" - DYNAMIC ADDRESS TRANSLATION BY CPU (OS/VS1, OS/VS2)
		.... ...1		CVTMVS2	"X'01'" - MULTIPLE MEMORY OPTION OF OS/VS2 IS PRESENT MDC131
116	(74)	X'12'	0	CVT8A0S2	"CVT4MS1+CVT6DAT" - OS/VS2 SYSTEM
117	(75)	ADDRESS	3	CVTDCBA	"VL3(IFBDCB)" - ADDRESS OF THE DCB FOR THE SYS1.LOGREC (OUTBOARD RECORDER) DATA SET FOR SYSTEM ENVIRONMENT RECORDING (SER)
120	(78)	SIGNED	4	CVTSV76M	- SVC 76 MESSAGE COUNT FIELD (OS/VS2) (MDC372)
124	(7C)	ADDRESS	4	CVTIXAVL	"V(IECIXAVL)" - ADDRESS OF THE I/O SUPERVISOR'S FREELIST POINTER WHICH CONTAINS THE ADDRESS OF THE NEXT REQUEST ELEMENT (OS/VS1) ADDRESS OF THE I/O SUPERVISOR'S COMMUNICATION AREA (IOCOM) (OS/VS2)
128	(80)	ADDRESS	4	CVTNUCB	- RESERVED (MDCXXX)
132	(84)	ADDRESS	4	CVTFBOSV	"V(IEWFBOSV)" - ADDRESS OF PROGRAM FETCH ROUTINE
136	(88)	ADDRESS	4	CVT0DS	"V(IEA0DS)" - ADDRESS OF ENTRY POINT OF THE DISPATCHER
140	(8C)	ADDRESS	4	CVTECVT	"V(IEAECVT)" - POINTER TO THE EXTENDED CVT
144	(90)	ADDRESS	4	CVTDAIRX	- ADDRESS OF THE 31- BIT ENTRY POINT OF IKJDAIR, TSO DYNAMIC ALLOCATION INTERFACE ROUTINE.
148	(94)	ADDRESS	4	CVTMSER	"V(IEEMSER)" - ADDRESS OF DATA AREA OF MASTER SCHEDULER RESIDENT DATA AREA
152	(98)	ADDRESS	4	CVT0PT01	"V(IEA0PT01)" - ADDRESS OF BRANCH ENTRY POINT OF POST ROUTINE
156	(9C)	ADDRESS	4	CVTTVT	- ADDRESS OF TSO VECTOR TABLE



Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
160	(A0)	SIGNED	4	CVT040ID	- IFB040I WTO MESSAGE ID. OWNERSHIP: OUTBOARD RECORDING (OBR). SERIALIZATION: COMPARE AND SWAP.
164	(A4)	BITSTRING	4	CVTMZ00	- HIGHEST ADDRESS IN VIRTUAL STORAGE FOR THIS MACHINE @(PCC0178)
168	(A8)	ADDRESS	4	CVT1EF00	- ADDRESS OF ROUTINE WHICH CREATES IRB'S FOR EXITS
172	(AC)	ADDRESS	4	CVTQOCR	- GRAPHICS INTERFACE TASK (GFX) FIELD. ADDRESS OF SEVENTH WORD OF GFX PARAMETER LIST, IF GFX IS ACTIVE. ZERO IF GFX IS NOT ACTIVE
176	(B0)	ADDRESS	4	CVTQMWR	"V(IEFQMWR)" - ADDRESS OF THE ALLOCATION COMMUNICATION AREA (MAPPED BY IEFZB432) - CONTAINS THE ADDRESSES OF SERVICE ROUTINES AND THE CHAIN OF MOUNT AND VERIFY COMMUNICATION AREAS.
180	(B4)	SIGNED	2	CVTSNCTR	- SERIAL NUMBER COUNTER FOR ASSIGNING SERIAL NUMBERS TO NON-SPECIFIC, UNLABELED MAGNETIC TAPE VOLUMES
182	(B6)	BITSTRING	1	CVTOPTA	- OPTION INDICATORS
		1... ..		CVTCCH	"X'80'" - CHANNEL CHECK HANDLER (CCH) OPTION PRESENT - RECOVERY MANAGEMENT SUPPORT (RMS) XM4686
		.1.. ..		CVTAPR	"X'40'" - ALTERNATE PATH RETRY (APR) OPTION PRESENT - RECOVERY MANAGEMENT SUPPORT (RMS)
		..1. ....		CVTDDR	"X'20'" - DYNAMIC DEVICE RECONFIGURATION (DDR) OPTION PRESENT - RECOVERY MANAGEMENT SUPPORT (RMS) (OS/VS1) DDR SYSTEM-INITIATED SWAP ACTIVE (OS/VS2) MDC126
		...1 ....		CVTNIP	"X'10'" - NIP IS EXECUTING
		.... 1...		CVTWARNUND	"X'08'" WARNUND processing is in effect
		.... .1..		CVT121TR	"X'04',,C'X'" - DO NOT TRANSLATE EXCP V=R.
		.... ..1.		CVTASCII	"X'02'" - ASCII TAPE PROCESSING IS GENERATED IN THIS SYSTEM
		.... ...1		CVTXPPF	"X'01'" - CPU HAS EXTENDED PRECISION FLOATING POINT FEATURE
183	(B7)	BITSTRING	1	CVTOPTB	- MISCELLANEOUS FLAGS
		1... ..		CVTPROT	"X'80'" - CPU HAS STORE PROTECTION FEATURE (OS/VS1)
		.1.. ....		CVTCTIMS	"X'40'" - IF ON, HARDWARE HAS THE CLOCK COMPARATOR AND CPU TIMER FEATURE INSTALLED, AND OS/VS1 SYSGEN HAS SPECIFIED THIS FEATURE (OS/VS1) MDC127
		..1. ....		CVTTOD	"X'20'" - CPU HAS TIME-OF-DAY CLOCK FEATURE
		...1 ....		CVTNLOG	"X'10'" - SYS1.LOGREC IS UNAVAILABLE FOR ERROR RECORDING. ALWAYS SET TO ZERO FOR OS/VS1. MDC127
		.... 1...		CVTAPTHR	"X'08'" - NIP SETS THIS BIT TO 1 WHEN DEVICE TESTING IS COMPLETE. IF 1, I/O SUPERVISOR USES AN ALTERNATE PATH TO A DEVICE WHEN A CONDITION CODE OF 3 EXISTS. THIS BIT IS RESET TO 0 BY NIP AFTER THE LINK PACK AREA IS INITIALIZED. YM2670
		.... .1..		CVTFP	"X'04'" - CPU HAS FETCH PROTECTION FEATURE (OS/VS1) ICB427

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.... ..1.		CVTVS1A	"X'02'" - VS1 ASSIST IS AVAILABLE FOR USE (OS/VS1) (MDC353)
		.... ...1		CVTVS1B	"X'01'" - VS1 ASSIST SUBSET IS AVAILABLE FOR USE (OS/VS1) (MDC365)
184	(B8)	ADDRESS	4	CVTQCDSR	"V(IEAQCDJR)" - CDE SEARCH ROUTINE ADDRESS (OS/VS2)
188	(BC)	ADDRESS	4	CVTQLPAQ	"V(IEAQLPAQ)" - ADDRESS OF POINTER TO MOST RECENT ENTRY ON LINK PACK AREA CDE QUEUE (OS/VS2)
192	(C0)	ADDRESS	4	CVTENFCT	"V(IEFENFDM)" EVENT NOTIFICATION CONTROL TABLE (MDC409)
196	(C4)	ADDRESS	4	CVTSMCA	- ADDRESS OF THE SYSTEM MANAGEMENT CONTROL AREA (SMCA) IF THE SYSTEM MANAGEMENT FACILITIES (SMF) OPTION IS PRESENT IN THE SYSTEM. OTHERWISE, ZERO.
200	(C8)	ADDRESS	4	CVTABEND	"V(IEABEND)" - ADDRESS OF SECONDARY CVT FOR ABEND IN EOT (OS/VS2)
204	(CC)	ADDRESS	4	CVTUSER	- A WORD AVAILABLE TO THE USER
208	(D0)	ADDRESS	4	CVTMDLDS	- RESERVED FOR MODEL-DEPENDENT SUPPORT
212	(D4)	SIGNED	2	CVTQABST	- AN SVC 13 (ABEND) INVOCATION (OS/VS2)
214	(D6)	SIGNED	2	CVTLNKSC	- AN SVC 6 (LINK) INVOCATION
216	(D8)	ADDRESS	4	CVTTSCE	- ADDRESS OF THE FIRST TIME SLICE CONTROL ELEMENT (TSCE)
220	(DC)	ADDRESS	4	CVTPATCH	"V(IEAPATCH)" - ADDRESS OF A 200-BYTE FE PATCH AREA
224	(E0)	ADDRESS	4	CVTRMS	"V(IGFRVT)" - RECOVERY MANAGEMENT SUPPORT (RMS) COMMUNICATIONS VECTOR. ADDRESS OF A MACHINE STATUS BLOCK.
228	(E4)	ADDRESS	4	CVTSPDME	- SERVICE PROCESSOR DAMAGE MONITOR ECB.
232	(E8)	ADDRESS	4	CVT0SCR1	"V(IEC0SCR1)" - ADDRESS OF THE SECTOR CALCULATION ROUTINE FOR ROTATIONAL POSITION SENSING (RPS)
236	(EC)	ADDRESS	4	CVTGTF(0)	- GENERALIZED TRACE FACILITY (GTF) CONTROL WORD ICB312
236	(EC)	ADDRESS	1	CVTGTFST	- GTF FLAG BYTES ICB312
		1... ....		CVTGTFAV	"X'80'" - IF ZERO, GTF NOT ACTIVE. IF ONE, GTF ACTIVE. (OS/VS2) MDC098
		.1... ....		CVTSPD	"X'40'" - SERVICE PROCESSOR DAMAGE. (1) INDICATES DAMAGE. (0) INDICATES NO DAMAGE.
		..1. ....		CVTWSPR	"X'20'" - WAITING FOR SERVICE PROCESSOR RESPONSE. (1) INDICATES OUTSTANDING REQUEST. (0) INDICATES NO OUTSTANDING REQUEST.
		.... .1..		CVTUSR	"X'04'" - TRACE=USR SPECIFIED. USER-REQUESTED TRACE DATA IS TO BE INCLUDED IN THE TRACE DATA SET. (MDC317)
		.... ..1.		CVTRNIO	"X'02'" - GTF IS ACTIVE AND TRACING RNIO EVENTS MDC187
237	(ED)	ADDRESS	3	CVTGTF A	"VL3(AHLHEAD)" -ADDRESS OF MAIN MONITOR CALL ROUTING TABLE, MCHEAD (OS/VS2) MDC156

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
240	(F0)	ADDRESS	4	CVTAQAVT(0)	- ADDRESS OF THE FIRST WORD OF THE TCAM DISPATCHER WHICH CONTAINS THE ADDRESS OF THE ADDRESS VECTOR TABLE (AVT). IF ZERO, TCAM IS NOT STARTED.
240	(F0)	BITSTRING	1	CVTTCMFG	- TCAM FLAGS
		1... ..		CVTTCDY	"X'80'" - TCAM IS READY TO ACCEPT USERS
		.1.. ..		CVTLDEV	"X'40'" - LOCAL DEVICE ATTACHED TO TCAM (MDC357)
		..1. ....		CVTNWTCM	"X'20'" - MULTIPLE TCAM FEATURE ACTIVE.
241	(F1)	ADDRESS	3	CVTAQAVB	- SAME AS CVTAQAVT ABOVE
244	(F4)	BITSTRING	1	CVTFLAG5	Flags, refreshed upon error, set during NIP and never changed
		1... ..		CVTVEF	"X'80'" Vector Extension Facility
		1... ..		CVTZ1	"X'80'" Z1
		.1.. ..		CVTEEC	"X'40'" EEC
		..1. ....		CVTNPAF	"X'20'" NNPA facility
245	(F5)	BITSTRING	1	CVTFLAG6	More flags
		1... ..		CVTSOLED	"X'80'" Solution Edition
246	(F6)	BITSTRING	1	CVTFLAG7	More flags
		1... ..		CVTSYSPLEXZAAPONLINE	"X'80'" The sysplex has one or more zAAP processors currently online
		.1.. ..		CVTSYSPLEXZCBPONLINE	"X'40'" The sysplex has one or more zCBP processors currently online
247	(F7)	BITSTRING	1	CVTFLAG8	More flags. IBM use only
		1... ..		CVTPQAP	"X'80'"
248	(F8)	ADDRESS	4	CVTSAF	- ADDRESS OF ROUTER VECTOR TABLE. @(PCC0549)
252	(FC)	ADDRESS	4	CVTEXT1	- ADDRESS OF OS - OS/VS COMMON EXTENSION ICB421
256	(100)	ADDRESS	4	CVTCBSP	- ADDRESS OF ACCESS METHOD CONTROL BLOCK STRUCTURE MDC195
260	(104)	ADDRESS	4	CVTPURG(0)	- ADDRESS OF SUBSYSTEM PURGE ROUTINE ICB330
260	(104)	BITSTRING	1		- RESERVED - FIRST BYTE OF CVTPURG
261	(105)	ADDRESS	3	CVTPURGA	- ADDRESS OF SUBSYSTEM PURGE ROUTINE ICB330
264	(108)	BITSTRING	4	CVTAMFF	- RESERVED FOR ACCESS METHOD FLAGS ICB436
268	(10C)	ADDRESS	4	CVTQMSG	"V(IEAVMSG)" - ADDRESS OF INFORMATION TO BE PRINTED BY ABEND. @(PCC0521)
272	(110)	ADDRESS	4	CVTDMSR(0)	- SAME AS CVTDMRA BELOW ICB346
272	(110)	BITSTRING	1	CVTDMSRF	- OPEN/CLOSE/EOV FLAG BYTE. SETTING BOTH BIT 0 AND BIT 1 ON WILL CAUSE BOTH KINDS OF DUMPS TO BE TAKEN. THESE BITS ARE USED DURING TESTING AND DEBUGGING WHEN OTHER DEBUG METHODS ARE INEFFECTIVE. (OS/VS2) (MDC350)
		1... ..		CVTSDUMP	"X'80'" - SET BY COREZAP. WILL CAUSE AN SDUMP TO BE TAKEN AND IEC999I MESSAGE ISSUED FOR EVERY ABEND ISSUED WITHIN AN OPEN/CLOSE/EOV OR DADSM FUNCTION. (OS/VS2) (MDC351)

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.1.. ....		CVTUDUMP	"X'40'" - SET BY COREZAP. WILL CAUSE AN ABEND DUMP TO BE TAKEN FOR EVERY ABEND ISSUED WITHIN AN OPEN/CLOSE/EOV OR DADSM FUNCTION. (OS/VS2) (MDC352)
273	(111)	ADDRESS	3	CVTDMSRA	- ADDRESS OF THE OPEN/CLOSE/EOV SUPERVISORY ROUTINE IN LPA. THIS ROUTINE HANDLES THE ROUTING OF CONTROL AMONG THE I/O SUPPORT ROUTINES. ICB346
276	(114)	ADDRESS	4	CVTSFR	"V(IEAVTSFR)" - ADDRESS OF SETFRR ROUTINE (IEAVTSFR) (MDC414)
280	(118)	ADDRESS	4	CVTGXL	"V(CSVMEM)" - ADDRESS OF CONTENTS SUPERVISION MEMORY TERMINATION ROUTINE OWNERSHIP - CONTENTS SUPERVISION.
284	(11C)	ADDRESS	4	CVTREAL	- ADDRESS OF THE VIRTUAL STORAGE BYTE FOLLOWING THE HIGHEST V=R STORAGE ADDRESS. @(PCC1294)
288	(120)	ADDRESS	4	CVTPTRV	"V(IEAPTRV)" - ADDRESS OF PAGING SUPERVISOR GENERAL ROUTINE TO TRANSLATE 24 BIT REAL ADDRESSES TO VIRTUAL ADDRESSES.
292	(124)	ADDRESS	4	CVTIHVP	- POINTER TO IHV\$COMM. INITIALIZED TO ZERO. OWNER: IHV/DATA HANDLER. SET BY: IHVSTRM. SERIALIZATION: NONE.
296	(128)	ADDRESS	4	CVTJESCT	"V(IEFJESCT)" - ADDRESS OF JOB ENTRY SUBSYSTEM (JES) CONTROL TABLE ICB342
300	(12C)	BITSTRING	4	CVTRS12C	- RESERVED
304	(130)	SIGNED	4	CVTTZ	- Difference between local time and UTC (Coordinated Universal Time) in binary units of 1.048576 seconds. Contains the same value as CVTLDTOL. CVTLDT0 (which contains CVTLDTOL) has this difference to a finer degree of accuracy.
308	(134)	ADDRESS	4	CVTMCHPR	- ADDRESS OF MACHINE CHECK PARAMETER LIST
312	(138)	ADDRESS	4	CVTEORM	- POTENTIAL REAL HIGH STORAGE ADDRESS. ONLY VALID PRE-z/Architecture. (SEE ECVTORM IN IHAECVT).
316	(13C)	ADDRESS	4	CVTPTRV3	"V(IEAVTRV3)" - ADDRESS OF PAGING SUPERVISOR ROUTINE TO TRANSLATE REAL ADDRESSES WHICH MAY EXCEED 24 BITS TO VIRTUAL ADDRESSES.
320	(140)	ADDRESS	4	CVTLKRM	"V(IEAVLKRM)" ADDRESS OF CML LOCK RESOURCE MANAGER
324	(144)	ADDRESS	4	CVTAPF(0)	- SAME AS CVTAPFA BELOW ICB360
324	(144)	BITSTRING	1		- RESERVED - FIRST BYTE OF CVTAPF
325	(145)	ADDRESS	3	CVTAPFA	"VL3(IEAVTEST)" ADDRESS OF BRANCH ENTRY POINT IN AUTHORIZED PROGRAM FACILITY (APF) ROUTINE
328	(148)	ADDRESS	4	CVTEXT2(0)	- ADDRESS OF OS/VS1 - OS/VS2 COMMON EXTENSION ICB330
328	(148)	BITSTRING	1		- RESERVED - FIRST BYTE OF CVTEXT2
329	(149)	ADDRESS	3	CVTEXT2A	SAME AS CVTEXT2 ABOVE ICB330
332	(14C)	ADDRESS	4	CVTHJES(0)	- SAME AS CVTHJESA BELOW ICB454
332	(14C)	BITSTRING	1		- RESERVED - FIRST BYTE OF CVTHJES
333	(14D)	ADDRESS	3	CVTHJESA	- ADDRESS OF OPTIONAL JOB ENTRY SUBSYSTEM (JES) COMMUNICATION VECTOR TABLE ICB454

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
336	(150)	BITSTRING	4	CVTRSTW2(0)	- STATUS DATA FOR RESTART FLIH OWNERSHIP: RESTART FLIH SERIALIZATION: RESTART RESOURCE
336	(150)	BITSTRING	1	CVTRS150	- Reserved. Was CVTRSTCP: LOGICAL CPU ADDRESS OF TARGET OF RESTART.
337	(151)	BITSTRING	1	CVTRSTRS	- RESTART REASON.
338	(152)	SIGNED	2	CVTRCP2B	- Logical CPU address of target of the restart.
340	(154)	CHARACTER	8	CVTSNAME	SYSTEM NAME FOR CURRENT SYSTEM. OWNERSHIP: IPL/NIP. SERIALIZATION: NONE. @(PCC0452)
348	(15C)	ADDRESS	4	CVTGETL	- ADDRESS OF IKJGETL, TSO GET LINE ROUTINE (MDC206) YM2225
352	(160)	ADDRESS	4	CVTLPSDR	"V(IEAVMSR)" - ADDRESS OF LINK PACK AREA (LPA) DIRECTORY SEARCH ROUTINE
356	(164)	ADDRESS	4	CVTPVTP	"V(IARMPVT)" - ADDRESS OF PAGE VECTOR TABLE
360	(168)	ADDRESS	4	CVTLPDIA(0)	- ADDRESS OF LINK PACK AREA (LPA) DIRECTORY (ON PAGE BOUNDARY)
360	(168)	BITSTRING	1	CVTDIRST	- FLAG BYTE
		1... ....		CVTDICOM	"X'80'" - LPA DIRECTORY HAS BEEN INITIALIZED BY NIP
361	(169)	ADDRESS	3	CVTLPDIR	- ADDRESS OF LINK PACK AREA (LPA) DIRECTORY (ON PAGE BOUNDARY)
364	(16C)	ADDRESS	4	CVTRBCB	- ADDRESS OF THE RECORD BUFFER'S CONTROL BLOCK
368	(170)	BITSTRING	4	CVTRS170	- RESERVED
372	(174)	CHARACTER	4	CVTSLIDA(0)	- IDENTITY OF TCB CAUSING SUPERVISOR LOCK BYTE (CVTSYLK) TO BE SET OR IDENTITY OF TCB THAT SECOND EXIT PROCESSING IS FOR WHEN CVTSEIC=1
372	(174)	BITSTRING	1	CVTSYLK	- SUPERVISOR LOCK. ONLY ENABLED TASKS MAY BE DISPATCHED
		1111 1111		CVTSYLKS	"X'FF'" - SET LOCK BYTE
		.... ....		CVTSYLKR	"X'00'" - RESET LOCK BYTE
373	(175)	ADDRESS	3	CVTSLID	- SAME AS CVTSLIDA ABOVE
376	(178)	SIGNED	4	CVTFLAGS(0)	- SYSTEM GLOBAL FLAGS
376	(178)	BITSTRING	1	CVTFLAG1	- FLAG BYTE
		1... ....		CVTRSMWD	"X'80'" - IF ONE REAL STORAGE MANAGER WINDOW WAS BUILT, INITIALIZED BY NIP.
		.1.. ....		CVTSVPRC	"X'40'" - SERVICE PROCESSOR ARCHITECTURE SUPPORTED.
		..1. ....		CVTCUSE	"X'20'" - CUSE. SET BY NIP
		...1 ....		CVTMVPG	"X'10'" - IF ONE, MOVEPAGE CAPABILITY IS PRESENT ON THIS SYSTEM. INITIALIZED BY NIP
		.... 1...		CVTOVER	"X'08'" - SUBPOOL OVERRIDE IS SUPPORTED. INITIALIZED BY NIP.
		.... .1..		CVTCSTR	"X'04'" - IF ONE, CSTRING FACILITY IS PRESENT ON THIS SYSTEM. INITIALIZED BY NIP.
		.... ..1.		CVTSUBSP	"X'02'" - IF ONE, SUBSPACE FACILITY IS PRESENT ON THIS SYSTEM. INITIALIZED BY NIP.
		.... ...1		CVTKPAR	"X'01'" - RESERVED FOR USE BY RTM ONLY. OWNERSHIP: RTM SERIALIZATION: NONE.

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
377	(179)	BITSTRING	1	CVTFLAG2	- FLAG BYTE
		1... ....		CVTCMPSC	"X'80'" - IF ONE, INDICATES PRESENCE OF MVS COMPRESSION/EXPANSION SERVICE. INITIALIZED ON. OWNERSHIP: CALLABLE SERVICES. SERIALIZATION: NONE.
		.1.. ....		CVTCMPSH	"X'40'" - IF ONE, INDICATES PRESENCE OF CMPSC COMPRESSION/EXPANSION HARDWARE INSTRUCTION SET BY NIP. OWNERSHIP: CALLABLE SERVICES. SERIALIZATION: NONE (UNCHANGED AFTER NIP).
		..1. ....		CVTSOPF	"X'20'" - IF ONE, INDICATES PRESENCE OF THE SUPPRESSION-ON-PROTECTION HARDWARE FACILITY. SET BY NIP. OWNERSHIP: SUPERVISOR CONTROL. SERIALIZATION: NONE (UNCHANGED AFTER NIP).
		...1 ....		CVTBFPH	"X'10'" - If one, indicates presence of BFP hardware instruction set. Set by NIP. Ownership: Supervisor. Serialization: None (unchanged after NIP).
		.... 1...		CVTPER2	"X'08'" - If one, indicates presence of PER2 hardware on all CPUs. Set by NIP. Ownership: Supervisor. Serialization: None (unchanged after NIP).
		.... .1..		CVTIQD	"X'04'" - If one, indicates that Internal Queued Direct Communications is supported. Set by IOS during NIP. Ownership: IOS. Serialization: None (unchanged after NIP).
		.... ..1.		CVTALR	"X'02'" - If one, indicates ASN and LX Reuse Architecture is enabled. Set by NIP. Ownership: Supervisor. Serialization: None (unchanged after NIP).
		.... ...1		CVTEDAT	"X'01'" - If one, indicates that the Enhanced DAT Architecture is available. Set by NIP. Ownership: Supervisor. Serialization: None (unchanged after NIP).
378	(17A)	BITSTRING	1	CVTFLAG3	- FLAG BYTE refreshed upon error, set during NIP and never changed
		1... ....		CVTESAME	"X'80'" -
		1... ....		CVTZARCH	"X'80'" - If one, indicates presence of z/Architecture hardware. Note that it is often simpler to check if PSA field FLCARCH is non-zero to determine this. Set by NIP. Ownership: Supervisor. Serialization: None (unchanged after NIP).
		.1.. ....		CVTPROCASCORE	"X'40'" A processor resource is viewed as a CPU Core
		..1. ....		CVTMULTICPUSPERCORE	"X'20'" When CvtProcAsCore is on, this indicates there are multiple CPUs defined within a CPU Core (On MT hardware). When CvtProcAsCore is off, this is always off
		...1 ....		CVTCPUASALIASTOCORE	"X'10'" When CvtProcAsCore is on, the term "CPU" is treated as an alias to "CORE" for D M and CF system commands. When CvtProcAsCore is off, this is always off
		.... 1...		CVTFLAG3DIAG	"X'08'" Diagnostic data for IBM use only

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.... .1..		CVTFLAG3RSVD	"X'04'" Reserved for IBM use
		.... ..1.		CVTZCBP	"X'02'" When bit is on, system fields with zCBP names and aliases with corresponding zAAP names contain data about zCBP processors.
		.... ...1		CVTGSF	"X'01'" GSF is available
379	(17B)	BITSTRING	1	CVTFLAG4	- Flag byte This is an interface for CA CVTZNALC, CVTDCPA, CVTTX, CVTTXC, CVTEDAT2 only
		1... ....		CVTP001I	"X'80'" If one, indicates that P001 support is installed
		.1.. ....		CVTP001A	"X'40'" If one, indicates that the system is in P001_Active mode Ownership: IPL/NIP Serialization: SALLOC
		..1. ....		CVTZNALC	"X'20'" zNALC
		...1 ....		CVTDCPA	"X'10'" Dynamic CPU Addition is enabled
		.... 1...		CVTTX	"X'08'" TX support is enabled
		.... 1...		CVTP002	"X'08'" P002 support is enabled
		.... .1..		CVTTXC	"X'04'" TXC support is enabled
		.... .1..		CVTP002C	"X'04'" P002C support is enabled
		.... ..1.		CVTRI	"X'02'" RI support is enabled
		.... ...1		CVTEDAT2	"X'01'" EDAT2 is enabled
380	(17C)	ADDRESS	4	CVTRT03	"V(IEAVRT03)" - ADDRESS OF SRB TIMING INITIALIZATION MODULE. (MDC406)
384	(180)	BITSTRING	8	CVTRS180	- RESERVED
392	(188)	ADDRESS	4	CVTEXSNR	"V(IEEVEXSN)" - ADDRESS OF EXCESSIVE SPIN NOTIFICATION ROUTINE (MDC386)
396	(18C)	BITSTRING	1	CVTEXSNL	- SERIALIZATION BYTE FOR EXCESSIVE SPIN NOTIFICATION ROUTINE (MDC387)
397	(18D)	ADDRESS	1	CVTSPVLK	- NUMBER OF TASKS WHICH HAVE TERMINATED WHILE OWNING SUPERVISOR LOCK WITHOUT OPERATOR HAVING YET BEEN NOTIFIED
398	(18E)	BITSTRING	1	CVTCTFLG	- SYSTEM CONTROL FLAGS
		1... ....		CVTTXTE	"X'80'" A Transactional Execution test environment is available. When only such a test environment exists, you should not use Transactional Execution in product code. In this test environment, the limited diagnostic data available upon such failures as program interrupts may well be inadequate to debug programs
		1... ....		CVTTXJ	"X'80'" Not a programming interface
		.11. ....		CVTCTLFGDIAG	"X'60'" Diagnostic data for IBM use only
		...1 ....		CVTDSTAT	"X'10'" - DEVSTAT OPTION IN EFFECT. DEVICE ADDRESS FOR 2319, 3330, 2314, 3330-1, 3340 CAN VARY ACROSS SYSTEMS. Not a programming interface. MDC189
		.... 1...		CVTDRMOD	"X'08'" - Set on when DRMODE=YES was specified.
		.... .1..		CVTNOMP	"X'04'" - MULTIPROCESSING CODE IS NOT IN THE SYSTEM. Not a programming interface. MDC176

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
		.... ..1.		CVTGTRCE	"X'02'" - GENERALIZED TRACE FACILITY (GTF) HAS SUPPRESSED SUPERVISOR TRACE. Not a programming interface. ICB446
		.... ...1		CVTSDTRC	"X'01'" - SVC DUMP HAS SUPPRESSED SUPERVISOR TRACE. Not a programming interface. ICB446
399	(18F)	SIGNED	1	CVTAPG	- DISPATCHING PRIORITY OF AUTOMATIC PRIORITY GROUP (APG) MDC152
404	(194)	ADDRESS	4	CVTRSCN	"V(IEATRSCN)" - ADDRESS OF ROUTINE TO SCAN TCB TREE
408	(198)	ADDRESS	4	CVTTAS	- ADDRESS OF ROUTINE TO TRANSFER ADDRESS SPACE YM2706
412	(19C)	ADDRESS	4	CVTTRCRM	- ADDRESS POINTER OF THE SYSTEM TRACE RESOURCE MANAGER. @(DCR816)
416	(1A0)	ADDRESS	4	CVTSHRVM	- LOWEST ADDRESS OF SHARED VIRTUAL STORAGE AREA. THIS ADDRESS WILL BE THE BEGINNING OF THE COMMON SERVICE AREA (CSA) (MDC324)
420	(1A4)	ADDRESS	4	CVT0VL01	"V(IEA0VL01)" - ENTRY POINT ADDRESS OF VALIDITY CHECK ROUTINE (IEA0VL01) USED TO COMPARE PROTECT KEY OF AN ADDRESS WITH TCB PROTECT KEY
424	(1A8)	ADDRESS	4	CVTPPGMX	"V(CSVEXIT)" - ADDRESS POINTER FOR MVS/370-XA.
428	(1AC)	BITSTRING	1	CVTGRSST	- GRS status. SERIALIZATION: None.
		1... ....		CVTE51GN	"X'80'" - When on, global resource contention data normally reported via ENF event code 51 to listeners on this system is unavailable or incomplete.
		.1.. ....		CVTE51LN	"X'40'" - When on, local resource contention data normally reported via ENF event code 51 to listeners on this system is unavailable or incomplete.
429	(1AD)	BITSTRING	1	CVTFLAG9	- Function-available flags
		.1.. ....		CVT_LLACOPY_DEBLOCKEXCLOK	"X'40'" LLACOPY supports DEBLOCKEXCLOK=YES
		.... ..1.		CVTIFAWICAVAILABLE	"X'02'" IFAWIC service is available for use
		.... ...1		CVTIFAWICINSTALLED	"X'01'" IFAWIC service is installed
430	(1AE)	BITSTRING	2	CVTBSM0F	- Return via reg 15, BSM 0,15
432	(1B0)	ADDRESS	4	CVTGVT	"V(GVT)" - ADDRESS OF THE GRS VECTOR TABLE (MDC414)
436	(1B4)	ADDRESS	4	CVTASCRF	- CREATED ASCB QUEUE HEADER (MDC337)
440	(1B8)	ADDRESS	4	CVTASCRL	- CREATED ASCB QUEUE TRAILER (MDC338)
444	(1BC)	ADDRESS	4	CVTPUTL	- ADDRESS OF IKJPUTL, TSO PUT LINE ROUTINE (MDC207) YM2225
448	(1C0)	ADDRESS	4	CVTSRBRT	"V(IEAPDSRT)" - DISPATCHER RETURN ADDRESS FOR SRB ROUTINES MDC130
452	(1C4)	ADDRESS	4	CVTOLT0A	"V(IFDOLT0A)" - BRANCH ENTRY TO OLTEP MEMORY TERMINATION RESOURCE MANAGER MDC129
456	(1C8)	ADDRESS	4	CVTSMFEX	"V(IEASMFEG)" - BRANCH ENTRY TO SYSTEM MANAGEMENT FACILITIES (SMF) EXCP COUNTING ROUTINE FOR VAM WINDOW INTERCEPT



Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
460	(1CC)	ADDRESS	4	CVTCSPIE	- ENTRY POINT ADDRESS OF THE SUPERVISOR CHECKPOINT/RESTART ROUTER (IEAVCKRS). RESOLVED BY IEAVNP05 AFTER THE LPA HAS BEEN BUILT. PREVIOUSLY CONTAINED THE ENTRY POINT ADDRESS OF THE RTM CHECKPOINT/RESTART EXIT ROUTINE (IEAVSPI).
464	(1D0)	ADDRESS	4	CVTPTGT	- ADDRESS OF IKJPTGT, TSO PUTGET ROUTINE (MDC208) YM2225
468	(1D4)	BITSTRING	1	CVTSIGPT	- SIGP TIMEOUT VALUE. OWNERSHIP: MACHINE CHECK HANDLER (MCH). SERIALIZATION: NONE (SET DURING IPL).
469	(1D5)	BITSTRING	1	CVTSPDMC	- SERVICE PROCESSOR DAMAGE MACHINE CHECK LOCK BYTE.
470	(1D6)	BITSTRING	1	CVTDSSAC	- DYNAMIC SUPPORT SYSTEM (DSS) ACTIVATED FLAG - USED BY RESTART FLIH. IF X'00', DSS NOT INITIALIZED. IF X'FF', DSS HAS BEEN INITIALIZED. MDC163
471	(1D7)	BITSTRING	1	CVTRS1D7	- RESERVED
472	(1D8)	ADDRESS	4	CVTSTCK	- ADDRESS OF IKJSTCK, TSO STACK ROUTINE (MDC209) YM2225
476	(1DC)	SIGNED	2	CVTMAXMP	- Maximum CPU address available for this IPL
478	(1DE)	BITSTRING	2	CVTBSM2	- RETURN VIA REG 2, BSM 0,2.
480	(1E0)	ADDRESS	4	CVTSCAN	- ADDRESS OF IKJSCAN, TSO SCAN ROUTINE (MDC210) YM2225
484	(1E4)	ADDRESS	4	CVTAUTHL	- POINTER TO AUTHORIZED LIBRARY TABLE. X'7FFFF001' IF DYNAMIC FORMAT APF TABLE. OWNED AND SET BY CONTENTS SUPERVISOR.
488	(1E8)	ADDRESS	4	CVTBLDCP	"V(IEAVBLDP)" - BRANCH ENTRY TO BUILD POOL MDC003
492	(1EC)	ADDRESS	4	CVTGETCL	"V(IEAVGTCL)" - BRANCH ENTRY TO GET CELL MDC004
496	(1F0)	ADDRESS	4	CVTFRECL	"V(IEAVFRCL)" - BRANCH ENTRY TO FREE CELL MDC005
500	(1F4)	ADDRESS	4	CVTDELC	"V(IEAVDELP)" - BRANCH ENTRY TO DELETE POOL MDC006
504	(1F8)	ADDRESS	4	CVTCRMN	"V(CRBRANCH)" - BRANCH ENTRY TO SVC 120 (GETMAIN/FREEMAIN CRBRANCH) MDC007
508	(1FC)	ADDRESS	4	CVTCRAS	"V(IGVGCAS)" - POINTER DEFINED ADDRESS OF BRANCH ENTRY TO 'CREATE ADDRESS SPACE'
512	(200)	ADDRESS	4	CVTQSAS	"V(IGVQSPET)" - POINTER DEFINED ADDRESS OF BRANCH ENTRY TO TASK TERMINATION
516	(204)	ADDRESS	4	CVTFRAS	"V(IGVGFAS)" - POINTER DEFINED ENTRY TO TASK TERMINATION
520	(208)	ADDRESS	4	CVTS1EE	"V(IGC043BR)" - BRANCH ENTRY TO STAGE 1 EXIT EFFECTOR MDC011
524	(20C)	ADDRESS	4	CVTPARS	- ADDRESS OF IKJPARS, TSO PARSE ROUTINE (MDC211) YM2225
528	(210)	ADDRESS	4	CVTQUIS	"V(IEAVAR02)" - BRANCH ENTRY TO QUIESCE MDC013
532	(214)	ADDRESS	4	CVTSTXU	- BRANCH ENTRY TO ATTENTION EXIT EPILOGUE (MDC321)
536	(218)	ADDRESS	4	CVTOPTTE	"V(IRARMIO0)" - BRANCH ENTRY ADDRESS TO SYSEVENT MDC015

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
540	(21C)	ADDRESS	4	CVTSDRM	- BRANCH ENTRY ADDRESS OF THE RESOURCE MANAGER ROUTINE FOR SVC DUMP. THIS ROUTINE CAN BE INVOKED BY MEMORY TERMINATION
544	(220)	ADDRESS	4	CVTCSRT	"V(CSRTABLE)" - POINTER TO CALLABLE SERVICE REQUEST TABLE OWNERSHIP: CSR SERIALIZATION: NONE
548	(224)	ADDRESS	4	CVTAQTOP	"V(IEFAQTOP)" - POINTER TO THE ALLOCATION QUEUE LOCK AREA.
552	(228)	ADDRESS	4	CVTVVMDI	- CONSTANT USED BY PAGED LINK PACK AREA (LPA) DIRECTORY SEARCH ALGORITHM
556	(22C)	ADDRESS	4	CVTASVT	- POINTER TO ADDRESS SPACE VECTOR TABLE (ASVT)
560	(230)	ADDRESS	4	CVTGDA	- POINTER TO GLOBAL DATA AREA (GDA) IN SQA
564	(234)	ADDRESS	4	CVTASCBH	"V(IEAMASCB)" - POINTER TO HIGHEST PRIORITY ADDRESS SPACE CONTROL BLOCK (ASCB) ON THE ASCB DISPATCHING QUEUE (HEAD OF ASCB QUEUE) MDC027
568	(238)	ADDRESS	4	CVTASCBL	"V(IEAMASCB)" - POINTER TO LOWEST PRIORITY ASCB ON THE ASCB DISPATCHING QUEUE (MDC339)
572	(23C)	ADDRESS	4	CVTRTMCT	- POINTER TO RECOVERY/TERMINATION CONTROL TABLE
576	(240)	ADDRESS	4	CVTSV60	"V(IEAVSTAG)" - BRANCH ENTRY ADDRESS FOR 24 OR 31 BIT ADDRESSING MODE USERS OF SVC 60. @(DCR854) ENTRY TO A GLUE ROUTINE.
580	(244)	ADDRESS	4	CVTSDMP	"V(IEAVTSGL)" - ADDRESS OF SVC DUMP BRANCH ENTRY POINT @(DCR664)
584	(248)	ADDRESS	4	CVTSCBP	"V(IEAVTSBP)" - ADDRESS OF SCB PURGE RESOURCE MANAGER.
588	(24C)	BITSTRING	4	CVTSDBF	- Address of 4K SQA buffer used by SVC Dump. High-order bit of this CVT word is used as lock to indicate buffer is in use. See related bit ASCBSDBF in macro IHAASCB.
592	(250)	ADDRESS	4	CVTRTMS	- ADDRESS OF SERVICABILITY LEVEL INDICATOR PROCESSING (SLIP) HEADER (MDC358)
596	(254)	ADDRESS	4	CVTTPIOS	- ADDRESS OF THE TELEPROCESSING I/O SUPERVISOR ROUTINE (TPIOS)
600	(258)	ADDRESS	4	CVTSIC	- BRANCH ADDRESS OF THE ROUTINE TO SCHEDULE SYSTEM INITIALIZED CANCEL
604	(25C)	ADDRESS	4	CVTOPCTP	"V(IRARMCNS)" - ADDRESS OF SYSTEM RESOURCES MANAGER (SRM) CONTROL TABLE MDC043
608	(260)	ADDRESS	4	CVTEXPRO	"V(IEAVEXPR)" - ADDRESS OF EXIT PROLOGUE/TYPE 1 EXIT MDC044
612	(264)	ADDRESS	4	CVTGSMQ	"V(IEAGSMQ)" - ADDRESS OF GLOBAL SERVICE MANAGER QUEUE MDC045
616	(268)	ADDRESS	4	CVTLSMQ	"V(IEALSMQ)" - ADDRESS OF LOCAL SERVICE MANAGER QUEUE MDC046
620	(26C)	BITSTRING	4	CVTRS26C	- RESERVED.
624	(270)	ADDRESS	4	CVTVWAIT	"V(IEAVWAIT)" - ADDRESS OF WAIT ROUTINE MDC048
628	(274)	ADDRESS	4	CVTPARRL	"V(CSVARMGR)" - ADDRESS OF PARTIALLY LOADED DELETE QUEUE.
632	(278)	ADDRESS	4	CVTAPFT	- ADDRESS OF AUTHORIZED PROGRAM FACILITY (APF) TABLE. INITIALIZED BY NIP.

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
636	(27C)	ADDRESS	4	CVTQCS01	"V(IEAQCS01)" - BRANCH ENTRY ADDRESS TO PROGRAM MANAGER USED BY ATTACH MDC051
640	(280)	SIGNED	4	CVTFQCB	- FORMERLY USED BY ENQ/DEQ. SHOULD ALWAYS BE ZERO. (MDC414)
644	(284)	SIGNED	4	CVTLQCB	- FORMERLY USED BY ENQ/DEQ. SHOULD ALWAYS BE ZERO. (MDC414)
648	(288)	ADDRESS	4	CVTRENQ	"V(IEAVENQ2)" - RESOURCE MANAGER ADDRESS FOR ENQ
652	(28C)	ADDRESS	4	CVTRSPIE	- RESOURCE MANAGER FOR SPIE. @ (PCC1076)
656	(290)	ADDRESS	4	CVTLKRM	"V(IEAVELRM)" - RESOURCE MANAGER ADDRESS FOR LOCK MANAGER.
660	(294)	ADDRESS	4	CVTCSD	- VIRTUAL ADDRESS OF COMMON SYSTEM DATA AREA (CSD). INITIALIZED BY NIP.
664	(298)	ADDRESS	4	CVTDQIQE	"V(IEADQIQE)" - RESOURCE MANAGER FOR EXIT EFFECTORS.
668	(29C)	ADDRESS	4	CVTRPOST	"V(IEARPOST)" - RESOURCE MANAGER FOR POST.
672	(2A0)	ADDRESS	4	CVT062R1	"V(IGC062R1)" - BRANCH ENTRY TO DETACH MDC060
676	(2A4)	ADDRESS	4	CVTVEAC0	"V(IEAVEAC0)" - ASCBCHAP BRANCH ENTRY MDC061
680	(2A8)	ADDRESS	4	CVTGLMN	"V(GLBRANCH)" - GLOBAL BRANCH ENTRY ADDRESS FOR GETMAIN/FREEMAIN MDC062
684	(2AC)	ADDRESS	4	CVTSPSA	"V(IEAVGWSA)" - POINTER TO GLOBAL WORK/SAVE AREA VECTOR TABLE (WSAG) MDC071
688	(2B0)	ADDRESS	4	CVTWSAL	"V(IEAVWSAL)" - ADDRESS OF TABLE OF LENGTHS OF LOCAL WORK/SAVE AREAS MDC072
692	(2B4)	ADDRESS	4	CVTWSAG	"V(IEAVWSAG)" - ADDRESS OF TABLE OF LENGTHS OF GLOBAL WORK/SAVE AREAS (MDC391)
696	(2B8)	ADDRESS	4	CVTWSAC	"V(IEAVWSAC)" - ADDRESS OF TABLE OF LENGTHS OF CPU WORK/SAVE AREAS MDC074
700	(2BC)	ADDRESS	4	CVTRECRQ	"V(IEAVTRGR)" - ADDRESS OF THE RECORDING REQUEST FACILITY (PART OF RTM1 - CALLED BY RTM2 AND RMS). @ (DCR854)
704	(2C0)	ADDRESS	4	CVTASMT	"V(ASMT)" - POINTER TO AUXILIARY STORAGE MANAGEMENT VECTOR TABLE (AMVT) (MDC340)
708	(2C4)	ADDRESS	4	CVTIOBP	"V(IDA121CV)" - ADDRESS OF THE BLOCK PROCESSOR CVT (MDC079) YM0029
712	(2C8)	ADDRESS	4	CVTSPOST	"V(IEASPOST)" - POST RESOURCE MANAGER TERMINATION ROUTINE (RMTR) ENTRY POINT MDC085
716	(2CC)	SIGNED	4	CVTRSTWD(0)	- RESTART RESOURCE MANAGEMENT WORD. CONTAINS IDENTIFIER OF USER IF RESTART IS IN USE. OTHERWISE, ZERO.
716	(2CC)	SIGNED	2	CVTRSTCI	- CPU ID OF THE CPU HOLDING THE RESTART RESOURCE.
718	(2CE)	BITSTRING	2	CVTRSTRI	- IDENTIFIER OF OWNING ROUTINE
720	(2D0)	ADDRESS	4	CVTFETCH	"V(IEWMSEPT)" - ADDRESS OF ENTRY POINT FOR BASIC FETCH.
724	(2D4)	ADDRESS	4	CVT044R2	"V(IGC044R2)" - ADDRESS OF IGC044R2 IN CHAP SERVICE ROUTINE MDC197

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
728	(2D8)	ADDRESS	4	CVTPERFM	- ADDRESS OF THE PERFORMANCE WORK AREA. SET BY IGX00018. MDC205
732	(2DC)	ADDRESS	4	CVTDAIR	- ADDRESS OF IKJDAIR, TSO DYNAMIC ALLOCATION INTERFACE ROUTINE (MDC212) YM2225
736	(2E0)	ADDRESS	4	CVTEHDEF	- ADDRESS OF IKJEHDEF, TSO DEFAULT SERVICE ROUTINE. @(PCC0919)
740	(2E4)	ADDRESS	4	CVTEHCIR	- ADDRESS OF IKJEHCIR, TSO CATALOG INFORMATION ROUTINE. @(PCC0919)
744	(2E8)	ADDRESS	4	CVTSSAP	- ADDRESS OF SYSTEM SAVE AREA
748	(2EC)	ADDRESS	4	CVTAIDVT	- POINTER TO APPENDAGE ID VECTOR TABLE
752	(2F0)	ADDRESS	4	CVTIPCDS	"V(IEAVEDR1)" - BRANCH ENTRY FOR DIRECT SIGNAL SERVICE ROUTINE
756	(2F4)	ADDRESS	4	CVTIPCRI	"V(IEAVERI1)" - BRANCH ENTRY FOR REMOTE IMMEDIATE SIGNAL SERVICE ROUTINE
760	(2F8)	ADDRESS	4	CVTIPCRP	"V(IEAVERP1)" - BRANCH ENTRY FOR REMOTE PENDABLE SIGNAL SERVICE ROUTINE
764	(2FC)	ADDRESS	4	CVTPCCAT	- POINTER TO PHYSICAL CCA VECTOR TABLE
768	(300)	ADDRESS	4	CVTLCCAT	- POINTER TO LOGICAL CCA VECTOR TABLE
772	(304)	ADDRESS	4	CVTXSFT	"V(IEAVXSFT)" - ADDRESS OF SYSTEM FUNCTION TABLE CONTAINING LINKAGE INDEX (LX) AND ENTRY INDEX (EX) NUMBERS FOR SYSTEM ROUTINES. (MDC414)
776	(308)	ADDRESS	4	CVTXSTKS	"V(IEAVXSTS)" - ADDRESS OF PCLINK STACK (SAVE=YES) ROUTINE. (MDC395)
780	(30C)	ADDRESS	4	CVTXSTKN	"V(IEAVXSTN)" - ADDRESS OF PCLINK STACK (SAVE=NO) ROUTINE. (MDC395)
784	(310)	ADDRESS	4	CVTXUNSS	"V(IEAVXUNS)" - ADDRESS OF PCLINK UNSTACK (SAVE=YES) ROUTINE. (MDC395)
788	(314)	ADDRESS	4	CVTPWI	- ADDRESS OF THE WINDOW INTERCEPT ROUTINE (MDC104) YM4043
792	(318)	ADDRESS	4	CVTPVBP	- ADDRESS OF THE VIRTUAL BLOCK PROCESSOR (MDC105) YM4043
796	(31C)	ADDRESS	4	CVTMFCTL	- POINTER TO MEASUREMENT FACILITY CONTROL BLOCK MDC100
800	(320)	ADDRESS	4	CVTMFRTR	- IF MEASUREMENT FACILITY IS ACTIVE, CONTAINS ADDRESS OF MEASUREMENT FACILITY ROUTINE. OTHERWISE, ADDRESS OF CVTBRET. MDC101
		1... ....		CVTMFACT	"X'80'" - IF ONE, I/O SUPERVISOR AND TIMER SECOND LEVEL INTERRUPT HANDLER HOOKS BRANCH TO MEASUREMENT FACILITY ROUTER. USED TO SET HIGH-ORDER BIT OF CVTMFRTR. MDC102
804	(324)	ADDRESS	4	CVTVPSIB	"V(IARPSIV)" - BRANCH ENTRY TO PAGE SERVICES
808	(328)	ADDRESS	4	CVTVSI	"V(IARXVIO)" - POINTER DEFINED, BRANCH ENTRY TO VAM SERVICES. @(DCR938)
812	(32C)	ADDRESS	4	CVTEXCL	"V(IECVEXCL)" - ADDRESS POINTER TO THE EXCP TERMINATION ROUTINE.
816	(330)	ADDRESS	4	CVTXUNSN	"V(IEAVXUNN)" - ADDRESS OF PCLINK UNSTACK (SAVE=NO) ROUTINE. (MDC395)
820	(334)	ADDRESS	4	CVTISNBR	"V(ISNBRNCH)" - ENTRY POINT ADDRESS OF DISABLED SERVICE PROCESSOR INTERFACE MODULE

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
824	(338)	ADDRESS	4	CVTXEXTR	"V(IEAVXEXT)" - ADDRESS OF PCLINK EXTRACT ROUTINE (MDC395)
828	(33C)	ADDRESS	4	CVTMSFRM	"V(IEAVMFRM)" - ADDRESS OF THE PROCESSOR CONTROLLER.
832	(340)	ADDRESS	4	CVTSCPIN	- ADDRESS OF IPL-TIME SCPINFO DATA BLOCK (ECVTSCPIN has address of "current"). Mapped by IHASCCB
836	(344)	ADDRESS	4	CVTWSMA	ADDRESS OF WAIT STATE MESSAGE AREA MUST BE DISPLAYABLE BY OPERATOR
840	(348)	ADDRESS	4	CVTRMBR	"V(RMBRANCH)" - ADDRESS OF REGMAIN BRANCH ENTRY. MDC123
844	(34C)	ADDRESS	4	CVTLFRM	"V(FMBRANCH)" - LIST FORMAT FREEMAIN BRANCH ENTRY MDC124 POINT. MDC124
848	(350)	ADDRESS	4	CVTGMBR	"V(GMBRANCH)" - LIST FORMAT GETMAIN BRANCH ENTRY MDC125 POINT. MDC125
852	(354)	ADDRESS	4	CVT0TC0A	- ADDRESS OF TASK CLOSE MODULE MDC128 IFG0TC0A. MDC128
856	(358)	SIGNED	4	CVTRLSTG	- SIZE OF ACTUAL REAL STORAGE ONLINE AT IPL TIME IN 'K'. ACCURATE ONLY BELOW 4TB, FOR LARGER SYSTEMS THIS CONTAINS 'FFFFFFF'X. SEE RCE_ONLINEFRAMECOUNTIPL.
860	(35C)	ADDRESS	4	CVTSPFRR	"V(IEAVESPR)" - 'SUPER FRR' ADDRESS (ADDRESS OF FUNCTIONAL RECOVERY ROUTINE ESTABLISHED AT NIP0 TIME TO PROTECT SUPERVISOR CONTROL PROGRAM).
864	(360)	BITSTRING	4	CVTRS360	- RESERVED.
868	(364)	ADDRESS	4	CVTSVT	"V(IEAVESVT)" - ADDRESS POINTER FOR FETCH PROTECTED PSASVT.
872	(368)	ADDRESS	4	CVTIRECM	- ADDRESS OF INITIATOR RESOURCE MDC158 MANAGER. MDC158
876	(36C)	ADDRESS	4	CVTDARCM	- ADDRESS OF DEVICE ALLOCATION MDC159 RESOURCE MANAGER. MDC159
880	(370)	ADDRESS	4	CVT0PT02	"V(IEA0PT02)" - ADDRESS OF POST ENTRY POINT MDC160 IEA0PT02. MDC160
884	(374)	BITSTRING	4	CVTRS374	RESERVED
888	(378)	ADDRESS	4	CVTWTCB	"V(IEAWTCB)" - ADDRESS OF WAIT STATE TCB. MDC164
892	(37C)	ADDRESS	4	CVTVACR	- ACR/VARY CPU CHANNEL RECOVERY MDC178 ROUTINE ADDRESS. ADDRESS FILLED IN MDC178 BY VARY CPU PROCESSOR. MDC178
896	(380)	ADDRESS	4	CVTRECON	- VARY CPU SHUTDOWN ROUTINE ADDRESS. ADDRESS FILLED IN BY VARY CPU PROCESSOR.
900	(384)	ADDRESS	4	CVTGTFR8	"V(AHLVCCR8)" - GENERALIZED TRACE FACILITY (GTF) MDC180 CONTROL REGISTER 8 INITIALIZATION MDC180 ROUTINE ADDRESS. MDC180
904	(388)	ADDRESS	4	CVTVSTOP	"V(IEEVSTOP)" - ADDRESS OF VARY CPU STOP CPU MDC169 ROUTINE. MDC169
908	(38C)	ADDRESS	4	CVTVPSA	- ADDRESS OF COPY OF SYSGEN'ED PSA - MDC170 PLACED HERE BY NIP. MDC170
912	(390)	ADDRESS	4	CVTRMPTT	- ADDRESS OF ISTRAMA1, THE VTAM RESOURCE MANAGER FOR NORMAL AND ABNORMAL TASK TERMINATION. (MDC322)
916	(394)	ADDRESS	4	CVTRMPMT	- ADDRESS OF ISTRAMA2, THE VTAM RESOURCE MANAGER FOR NORMAL AND ABNORMAL MEMORY TERMINATION. (MDC323)

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
920	(398)	ADDRESS	4	CVTEXP1	"V(IEAVEXP1)" - ADDRESS OF EXIT PROLOGUE WHICH MDC173 RETURNS TO THE DISPATCHER. MDC173
924	(39C)	ADDRESS	4	CVTCSDDL	- REAL ADDRESS OF COMMON SYSTEM DATA MDC174 AREA (CSD). INITIALIZED BY NIP. MDC174
928	(3A0)	ADDRESS	4	CVTSSRB	"V(IGC07903)" - STATUS STOP SRB ENTRY. MDC175
932	(3A4)	BITSTRING	4	CVTRS3A4	- RESERVED
936	(3A8)	ADDRESS	4	CVTQV1	"V(IEAVGQV1)" - ADDRESS OF QUEUE VERIFICATION FOR SINGLE THREADED QUEUES WITH MDC181 HEADERS ONLY. MDC181
940	(3AC)	ADDRESS	4	CVTQV2	"V(IEAVEQV2)" - ADDRESS OF QUEUE VERIFICATION FOR MDC182 SINGLE THREADED QUEUES WITH MDC182 HEADER AND TRAILER. MDC182
944	(3B0)	ADDRESS	4	CVTQV3	"V(IEAVEQV3)" - ADDRESS OF QUEUE VERIFICATION FOR MDC183 DOUBLE THREADED QUEUES. MDC183
948	(3B4)	ADDRESS	4	CVTGSDA	"V(IEAVGSDA)" - ADDRESS OF GLOBAL SYSTEM DUPLEX AREA.
		1... ..		CVTGSDAB	"X'80'" - IF HIGH-ORDER BIT IS ONE, THERE IS MDC185 A VALID VALUE IN FOLLOWING 31 BITS. MDC185
952	(3B8)	ADDRESS	4	CVTADV	"V(IEAVEADV)" - ADDRESS OF ADDRESS VERIFICATION MDC186 ROUTINE. MDC186
956	(3BC)	ADDRESS	4	CVTTPIO	"V(IGC124)" - ADDRESS OF VTAM TPIO (SVC 124) MDC193 ROUTINE. MDC193
960	(3C0)	BITSTRING	4	CVTRS3C0	- RESERVED
964	(3C4)	ADDRESS	4	CVTEVENT	"V(IEAVEVT2)" - BRANCH ENTRY ADDRESS TO EVENTS (FAST MULTIPLE WAIT ROUTINE). @(DCR738)
968	(3C8)	ADDRESS	4	CVTSSCR	- ADDRESS OF STORAGE SYSTEM CONTROLLER RECOVERY MANAGER CLEANUP ROUTINE (SSC RMCR). (MDC319)
972	(3CC)	ADDRESS	4	CVTCBBR	"V(CBBRANCH)" - BRANCH ENTRY ADDRESS TO GETMAIN/FREEMAIN. (MDC325)
976	(3D0)	ADDRESS	4	CVTEFF02	- ADDRESS OF IKJEFF02, TSO MESSAGE ISSUER SERVICE ROUTINE. (MDC326)
980	(3D4)	ADDRESS	4	CVTLSCH	"V(IEAVESC1)" - ADDRESS OF LOCAL SCHEDULE. (MDC364)
984	(3D8)	ADDRESS	4	CVTCDEQ	- ADDRESS OF PROGRAM MANAGER AVAILBLE CDE QUEUE CONTROL AREA. (MDC369)
988	(3DC)	ADDRESS	4	CVTHSM	- POINTER TO HIERARCHICAL STORAGE MANAGER (HSM) QUEUE CONTROL TABLE. (MDC375)
992	(3E0)	ADDRESS	4	CVTRAC	- ADDRESS OF ACCESS CONTROL CVT. (MDC320)
996	(3E4)	ADDRESS	4	CVTCGK	"V(IARXKEY)" - ADDRESS OF ROUTINE USED TO CHANGE THE KEY OF VIRTUAL PAGES. @(PCC0529)
1000	(3E8)	ADDRESS	4	CVTSRM	"V(IRARMEPS)" - ADDRESS OF ENTRY TABLE FOR SRM, ENTRY TABLE IS INITIALIZED BY NIP10. (MDC367)
1004	(3EC)	ADDRESS	4	CVT0PT0E	"V(IEA0PT0E)" - ENTRY POINT TO IDENTIFY POST EXIT ROUTINES. (MDC334)
1008	(3F0)	ADDRESS	4	CVT0PT03	"V(IEA0PT03)" - POST REINVOICATION ENTRY POINT FROM POST EXIT ROUTINES. (MDC335)

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
1012	(3F4)	ADDRESS	4	CVTTCASP	- POINTER TO THE TSO/VTAM TERMINAL CONTROL ADDRESS SPACE (TCAS) TABLE. (MDC336)
1016	(3F8)	ADDRESS	4	CVTCTTVT	- CTT VT
1020	(3FC)	ADDRESS	4	CVTJTERM	"V(ILRJTERM)" - POINTER DEFINED ADDRESS OF AUXILIARY STORAGE MANAGEMENT JOB TERMINATION RESOURCE MANAGER.
1024	(400)	ADDRESS	4	CVTRSUME	"V(IEAVRSUH)" - ADDRESS OF RESUME FUNCTION. (MDC414)
1028	(404)	ADDRESS	4	CVTTCTL	"V(IEAVTCTL)" - ADDRESS OF TRANSFER CONTROL (TCTL) FUNCTION. (MDC345)
1032	(408)	ADDRESS	4	CVTRMT	- ADDRESS OF RESOURCE MANAGER CONTROL STRUCTURE (RMT) OWNERSHIP: RTM. SERIALIZATION: NONE.
1036	(40C)	ADDRESS	4	CVTT6SVC	"V(IEAVET6E)" - ENTRY POINT ADDRESS FOR TYPE 6 SVC EXIT FUNCTION. (MDC347)
1040	(410)	ADDRESS	4	CVTSUSP	"V(IEAVSPND)" - ADDRESS OF SUSPEND ROUTINE. (MDC348)
1044	(414)	ADDRESS	4	CVTIHASU	"V(IEAIHASU)" - ADDRESS OF BIT STRING. (MDC355)
1048	(418)	ADDRESS	4	CVTSFV	"V(IEAVTSFV)" - ADDRESS OF SETFRR ROUTINE ABOVE 16M
1052	(41C)	ADDRESS	4	CVTIDEVN	"V(IOSVDEVN)" - ADDRESS OF DEVICE NUMBER CONVERSION ROUTINE OWNERSHIP: IOS. SERIALIZATION: NONE.
1056	(420)	ADDRESS	4	CVTSMF83	- ADDRESS OF BRANCH ENTRY TO SMF SVC 83. (MDC378)
1060	(424)	ADDRESS	4	CVTSMFSP	"V(IEASMFSP)" - ADDRESS OF SMF SUSPEND HANDLER.
1064	(428)	ADDRESS	4	CVTMSFCB	- ADDRESS OF MAINTENANCE AND SERVICE FACILITY CONTROL BLOCK (MSFCB). (MDC396)
1068	(42C)	ADDRESS	4	CVTHID	"V(IOSVHID)" - ADDRESS OF SCP HOST ID. @(DCR819)
1072	(430)	ADDRESS	4	CVTPSXM	"V(IARPSXM)" - ADDRESS OF CROSS MEMORY PAGE FIX AND PAGE FREE. (MDC414)
1076	(434)	ADDRESS	4	CVTUCBSC	"V(IOSVUCB)" - ADDRESS OF UCB SCAN SERVICE. @(DCR377)
1080	(438)	ADDRESS	4	CVTTPUR	- DDR QUEUE OF TAPE UNIT-RECORD SWAP REQUESTS.
1084	(43C)	ADDRESS	4	CVTDPUR	- DDR QUEUE OF DASD SWAP REQUESTS.
1088	(440)	ADDRESS	4	CVTTRPOS	- DDR QUEUE OF TAPES TO BE REPRESENTED.
1092	(444)	ADDRESS	4	CVTRS444	- Reserved, must always be 0. Was CVTRESTX, VIRTUAL ADDRESS OF TEXT TO BE PLACED ON CONSOLE FRAME.
1096	(448)	SIGNED	2	CVTXCPCT	- MAXIMUM EXCP COUNT PER ADDRESS SPACE.
1098	(44A)	SIGNED	2	CVTCALL	- A BASSM 14,15 INSTRUCTION. POINTER USED VIA AN EXECUTE INSTRUCTION TO BRANCH TO USERS EXITS
1100	(44C)	ADDRESS	4	CVTVFIND	"V(CSVVFIND)" - THE POINTER TO VIRTUAL FETCH BUILD AND FIND ROUTINE.
1104	(450)	ADDRESS	4	CVTVFGET	"V(CSVVFGET)" - THE POINTER TO VIRTUAL FETCH GET ROUTINE.

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
1108	(454)	ADDRESS	4	CVTVFMEM	RESERVED. THIS FIELD IS NO LONGER USED. ANYONE USING IT AS A POINTER WILL PROGRAM CHECK
1112	(458)	ADDRESS	4	CVTVFCB	- THE POINTER TO VIRTUAL FETCH INTERNAL CONTROL BLOCK IN CSA, INITIALIZED TO ZERO AND SET TO NON-ZERO VALUE BY VIRTUAL FETCH INITIALIZATION ROUTINE.
1116	(45C)	ADDRESS	4	CVTPGSR	"V(IARPIBR)" - POINTER DEFINED ADDRESS OF ENTRY TO PAGE SERVICES (FIX,FREE,LOAD, OUT,RLSE,ANYWHERE).
1120	(460)	ADDRESS	4	CVTTSKI	"V(IGVSTSKI)" - POINTER DEFINED ADDRESS OF TASK MANAGEMENT/STORAGE MANAGEMENT INTERFACE ROUTINE.
1124	(464)	ADDRESS	4	CVTCPGUB	"V(IGVCPGUB)" - POINTER DEFINED ADDRESS OF CPOOL GET UNCONDITIONAL BRANCH ENTRY ROUTINE.
1128	(468)	ADDRESS	4	CVTCPGUP	"V(IGVCPGUP)" - POINTER DEFINED ADDRESS OF CPOOL GET UNCONDITIONAL PC-ENTRY ROUTINE.
1132	(46C)	ADDRESS	4	CVTCPGTC	"V(IGVCPGTC)" - POINTER DEFINED ADDRESS OF GET UNCONDITIONAL ROUTINE.
1136	(470)	ADDRESS	4	CVTCPFRE	"V(IGVCPFRE)" - POINTER DEFINED ADDRESS OF CPOOL FREE ROUTINE.
1140	(474)	ADDRESS	4	CVTSLIST	"V(IGVSLIST)" - POINTER DEFINED ADDRESS OF VSM LIST SERVICE.
1144	(478)	ADDRESS	4	CVTSREGN	"V(IGVSREGN)" - POINTER DEFINED ADDRESS OF VSM REGION SIZE.
1148	(47C)	ADDRESS	4	CVTSLOC	"V(IGVSLOC)" - POINTER DEFINED ADDRESS OF VSM LOCATOR SERVICE.
1152	(480)	ADDRESS	4	CVTCPBDB	"V(IGVCPBDB)" - POINTER DEFINED ADDRESS OF CPOOL BUILD ENTRY ROUTINE.
1156	(484)	ADDRESS	4	CVTCPDLB	"V(IGVCPDLB)" - POINTER DEFINED ADDRESS OF CPOOL DELETE BRANCH ENTRY ROUTINE.
1160	(488)	ADDRESS	4	CVTDOFFS	- STARTING REAL ADDRESS OF DAT-OFF NUCLEUS.
1164	(48C)	ADDRESS	4	CVTDOFFE	- ENDING REAL ADDRESS OF DAT-OFF NUCLEUS.
1168	(490)	ADDRESS	4	CVTRCEP	"V(IARMRRCE)" - ADDRESS OF THE RSM CONTROL AND ENUMERATION AREA.
1172	(494)	ADDRESS	4	CVTCPGUS	"V(IGVCPGUS)" - ADDRESS OF CPOOL GET UNCONDITIONAL PC-ENTRY ROUTINE WHICH SAVES SECONDARY ASID STATUS. @(DCR722)
1176	(498)	ADDRESS	4	CVTGRRGN	"V(IGVGRRGN)" - POINTER DEFINED ADDRESS OF GET REAL REGION ROUTINE.
1180	(49C)	ADDRESS	4	CVTGVRGN	"V(IGGVVRGN)" - POINTER DEFINED ADDRESS OF GET VIRTUAL REGION ROUTINE.
1184	(4A0)	BITSTRING	1	CVTIONLV	- DEFAULT VALUE OF IOS LEVEL. @(PCC0461)
1185	(4A1)	BITSTRING	3	CVTRS4A1	- RESERVED
EXIT CODE FOR NORMAL AND/OR ABNORMAL END APPENDAGES FOR I/O DRIVERS.					
1190	(4A6)	BITSTRING	2		- RETURN VIA A BSM.
1192	(4A8)	BITSTRING	4	CVTFUNC	- Reserved for solution/offering use. Must be zero for full function MVS.



Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		1... ..		CVTSOLN	"X'80'" - If high order bit is on, this is not a full function MVS system, but rather, a solution/offering.
1196	(4AC)	ADDRESS	4	CVTSMEXT	- ADDRESS OF STORAGE MAP EXTENSION.
1200	(4B0)	ADDRESS	4	CVTNUCMP	- ADDRESS OF NUCLEUS MAP.
1204	(4B4)	BITSTRING	1	CVTXAFL	- FLAG BYTE FOR MVS/XA PROCESSING. @(PCC3762)
		1... ..		CVTCSRIM	"X'80'" - EXPLICIT LOAD PROCESSING REQUIRED FOR CONTENTS SUPERVISOR RIM.
1205	(4B5)	BITSTRING	3	CVTRS4B5	- RESERVED
1208	(4B8)	ADDRESS	4	CVTVTAM	- ADDRESS OF VTAM COMMAND PROCESSOR (ISTCFF3D). @(DCR642)
1212	(4BC)	ADDRESS	4	CVTSPIIP	- ADDRESS OF RTM INTERFACE TO RETURN PROGRAM MASK TO CONTENTS SUPERVISOR, (ON SPIE/ESPIE).
1216	(4C0)	ADDRESS	4	CVTCKRAS(0)	- OLD NAME FOR CVTDFA FIELD.
1216	(4C0)	ADDRESS	4	CVTDFA	"V(DFAIDTAB)" - ADDRESS OF DFP ID TABLE, MAPPED BY THE DFA. OWNERSHIP: DFP.
1220	(4C4)	ADDRESS	4	CVTNVT0	"V(IEAVNVT0)" - ADDRESS OF DATA IN DAT-ON NUCLEUS
1224	(4C8)	ADDRESS	4	CVTCSOMF	- OWNER OF CHANNEL MEASUREMENT FACILITY. @(DCR1020)
1228	(4CC)	ADDRESS	4	CVTCSOAL	- OWNER OF ADDRESS LIMIT FACILITY.
1232	(4D0)	ADDRESS	4	CVTICHPT	- ADDRESS OF THE INSTALLED CHANNEL PATH TABLE. @(DCR719)
1236	(4D4)	ADDRESS	4	CVTCSOCR	- CHANNEL SUBSYSTEM OWNER - CHANNEL PATH RESET FACILITY. @(DCR719)
1240	(4D8)	ADDRESS	4	CVTCSOCS	- CHANNEL SUBSYSTEM OWNER - CHANNEL PATH STATUS FACILITY. @(DCR719)
1244	(4DC)	ADDRESS	4	CVTLTATA	- LINK LIST TABLE ADDRESS. @(DCR719)
1248	(4E0)	ADDRESS	4	CVTDCQA	- ADDRESS OF DEVICE CLASS QUEUE
1252	(4E4)	ADDRESS	4	CVTUCBA	- ADDRESS OF THE FIRST UCB IN THE CHAIN OF UCB'S. @(DCR719)
1256	(4E8)	ADDRESS	4	CVTVESTU	"V(IEAVESTU)" - ADDRESS OF THE ENTRY POINT OF THE SVC UPDATE ROUTINE. @(DCR825)
1260	(4EC)	ADDRESS	4	CVTNUCLU	"V(IEAVENLU)" - ADDRESS TO SUPPORT THE NUCLEUS MAP LOOKUP ROUTINE. @(DCR892)
1264	(4F0)	BITSTRING	16	CVTOSLVL(0)	SYSTEM LEVEL INDICATORS The presence of certain hardware functions is indicated within the SCCB (mapped by macro IHASCCB, pointed to by CVTSCPIN and/or ECVTSCPIN). The presence of other hardware functions can be found within CVT field CVTFLAGS2.
1264	(4F0)	BITSTRING	1	CVTOSLV0	BYTE 0 OF CVTOSLVL
		1... ..		CVTH3310	"X'80'" HBB3310 FUNCTIONS ARE PRESENT
		1... ..		CVTESA	"X'80'" ESA/370 IS SUPPORTED
		1... ..		CVTXAX	"X'80'" ESA/370 IS SUPPORTED (XAX - OLD NAME)
		.1.. ....		CVTH4420	"X'40'" HBB4420 FUNCTIONS ARE PRESENT.
		..1. ....		CVTJ3313	"X'20'" JBB3313 FUNCTIONS ARE PRESENT
		...1 ....		CVTJ3311	"X'10'" JBB3311 FUNCTIONS ARE PRESENT

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
1265	(4F1)	BITSTRING	1	CVTHIPER	"X'10'" HIPERSPACES ARE SUPPORTED
				CVTH4410	"X'08'" HBB4410 FUNCTIONS ARE PRESENT.
				CVTLKR	"X'08'" SPIN LOCK RESTRUCTURE INDICATOR.
				CVTUCBSV	"X'08'" UCB SERVICES INSTALLED.
				CVTCADS	"X'04'" SCOPE=COMMON DATA SPACES SUPPORTED
				CVTCRPTL	"X'02'" ENCRYPTION ASYMMETRIC FEATURE IS SUPPORTED
				CVTJ4422	"X'01'" JBB4422 FUNCTIONS ARE PRESENT
				CVTOSLV1	BYTE 1 OF CVTOSLVL
				CVTH4430	"X'80'" HBB4430 FUNCTIONS ARE PRESENT
				CVTDYAPF	"X'80'" DYNAMIC APF, THROUGH CSVAPF, PRESENT
				CVTWLM	"X'40'" WORKLOAD MANAGER IS INSTALLED
				CVTH5510	"X'20'" HBB5510 FUNCTIONS ARE PRESENT
				CVTDYNEX	"X'20'" CSVDYNEX FOR DYNAMIC EXITS IS PRESENT
				CVTH5520	"X'10'" HBB5520 FUNCTIONS ARE PRESENT
				CVTENCLV	"X'10'" ENCLAVES FUNCTION IS PRESENT
				CVTJ5522	"X'08'" JBB5522 FUNCTIONS ARE PRESENT
				CVTH5530	"X'04'" HBB6603 FUNCTIONS ARE PRESENT
				CVTH6603	"X'04'" HBB6603 FUNCTIONS ARE PRESENT
				CVTOS390_010300	"X'04'" OS/390 R3
				CVTOS390_R3	"X'04'" OS/390 R3
				CVTDYNL	"X'04'" Dynamic LNKST, via CSVDYNL, is present
				CVTH6601	"X'02'" OS/390 release 1
				CVTOS390	"X'02'" OS/390 release 1 This bit is on for all releases starting with OS/390 release 1
				CVTOS390_010100	"X'02'" OS/390 R1
				CVTOS390_R1	"X'02'" OS/390 R1
				CVTPRDED	"X'02'" Product enable/disable (IFAEDxxx) is present
				CVTJ6602	"X'01'" OS/390 release 2
				CVTOS390_010200	"X'01'" OS/390 R2
				CVTOS390_R2	"X'01'" OS/390 R2
				CVTPARMC	"X'01'" Logical Parmlib Service is available via IEFPRMLB.
1266	(4F2)	BITSTRING	1	CVTOSLV2	BYTE 2 OF CVTOSLVL
				CVTOS390_010400	"X'80'" OS/390 R4
				CVTOS390_020400	"X'80'" OS/390 R4
				CVTOS390_R4	"X'80'" OS/390 R4
				CVTJ6604	"X'80'" OS/390 R4
				CVTDYLPA	"X'80'" Dynamic LPA (CSVDYLPA) available
				CVTRTLS	"X'80'" Runtime Library Services (CSVRTLs)
				CVTOS390_020500	"X'40'" OS/390 R5

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.1.. ....		CVTOS390_R5	"X'40'" OS/390 R5
		.1.. ....		CVTH6605	"X'40'" OS/390 R5
		..1. ....		CVTOS390_020600	"X'20'" OS/390 R6
		..1. ....		CVTOS390_R6	"X'20'" OS/390 R6
		..1. ....		CVTH6606	"X'20'" OS/390 R6
		...1 ....		CVTBFP	"X'10'" Binary Floating Point support (simulated unless CVTBFPH is on)
		.... 1...		CVTOS390_020700	"X'08'" OS/390 R7
		.... 1...		CVTOS390_R7	"X'08'" OS/390 R7
		.... 1...		CVTJ6607	"X'08'" OS/390 R7
		.... .1..		CVTOS390_020800	"X'04'" OS/390 R8
		.... .1..		CVTOS390_R8	"X'04'" OS/390 R8
		.... .1..		CVTH6608	"X'04'" OS/390 R8
		.... ..1.		CVTOS390_020900	"X'02'" OS/390 R9
		.... ..1.		CVTOS390_R9	"X'02'" OS/390 R9
		.... ..1.		CVTJ6609	"X'02'" OS/390 R9
		.... ..1.		CVTH6609	"X'02'" OS/390 R9
		.... ...1		CVTOS390_021000	"X'01'" OS/390 R10 functions are present
		.... ...1		CVTOS390_R10	"X'01'" OS/390 R10 functions are present
		.... ...1		CVTH7703	"X'01'" OS/390 R10 functions are present
1267	(4F3)	BITSTRING	1	CVTOSLV3	BYTE 3 OF CVTOSLVL
		1... ....		CVTPAUSE	"X'80'" Pause/Release services are present
		.1.. ....		CVTPAUS2	"X'40'" IEAVAPE2 and related services, and Ownership options.
		..1. ....		CVTZOS	"X'20'" z/OS V1R1 functions are present This bit is on for all releases starting with z/OS release 1.
		..1. ....		CVTZOS_010100	"X'20'" z/OS V1R1 functions are present
		..1. ....		CVTZOS_V1R1	"X'20'" z/OS V1R1 functions are present
		..1. ....		CVTJ7713	"X'20'" JBB7713
		..1. ....		CVTLPARC	"X'20'" LPAR Clustering is present.
		...1 ....		CVTZOS_010200	"X'10'" z/OS V1R2 functions are present
		...1 ....		CVTZOS_V1R2	"X'10'" z/OS V1R2 functions are present
		...1 ....		CVTH7705	"X'10'" HBB7705 functions are present
		...1 ....		CVTV64	"X'10'" 64-bit virtual services are present. You should ensure FLCARCH (in IHAPSA) is non-zero before using
		.... 1...		CVTZOS_010300	"X'08'" z/OS V1R3 functions are present
		.... 1...		CVTZOS_V1R3	"X'08'" z/OS V1R3 functions are present
		.... 1...		CVTH7706	"X'08'" HBB7706 functions are present
		.... .1..		CVTZOS_010400	"X'04'" z/OS V1R4 functions are present

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.... .1..		CVTZOS_V1R4	"X'04'" z/OS V1R4 functions are present
		.... .1..		CVTH7707	"X'04'" HBB7707 functions are present
		.... ..1.		CVTZOS_010500	"X'02'" z/OS V1R5 functions are present
		.... ..1.		CVTZOS_V1R5	"X'02'" z/OS V1R5 functions are present
		.... ..1.		CVTH7708	"X'02'" HBB7708 functions are present
		.... ...1		CVTZOS_010600	"X'01'" z/OS V1R6 functions are present
		.... ...1		CVTZOS_V1R6	"X'01'" z/OS V1R6 functions are present
		.... ...1		CVTH7709	"X'01'" HBB7709 functions are present
1268	(4F4)	BITSTRING	1	CVTOSLV4	BYTE 4 OF CVTOSLVL
		1... ....		CVTCSRSI	"X'80'" CSRSI service is available
		.1.. ....		CVTUNICS	"X'40'" Unicode callable services available
		..1. ....		CVTCSRUN	"X'20'" CSRUNIC callable service available
		...1 ....		CVTILM	"X'10'" IBM License Manager functions are present
		.... 1...		CVTALRS	"X'08'" ASN-and-LX-Reuse architecture is supported. It might not be enabled. See CVTALR.
		.... .1..		CVTT0CP	"X'04'" TIMEUSED TIME_ON_CP
		.... ..1.		CVTZIIP	"X'02'" zIIP support is present
		.... ..1.		CVTSUP	"X'02'" zIIP support is present
		.... ...1		CVTIFAR	"X'01'" IFA routine is present
1269	(4F5)	BITSTRING	1	CVTOSLV5	BYTE 5 OF CVTOSLVL
		1... ....		CVTZOSE	"X'80'" z/OS.e
		1... ....		CVTZOSAS	"X'80'" z/OS.e
		1... ....		CVTPUMA	"X'80'" z/OS.e
		.1.. ....		CVTZOS_010700	"X'40'" z/OS V1R7 functions are present
		.1.. ....		CVTZOS_V1R7	"X'40'" z/OS V1R7 functions are present
		.1.. ....		CVTH7720	"X'40'" HBB7720 functions are present
		..1. ....		CVTZOS_010800	"X'20'" z/OS V1R8 functions are present
		..1. ....		CVTZOS_V1R8	"X'20'" z/OS V1R8 functions are present
		..1. ....		CVTH7730	"X'20'" HBB7730 functions are present
		...1 ....		CVTZOS_010900	"X'10'" z/OS V1R9 functions are present
		...1 ....		CVTZOS_V1R9	"X'10'" z/OS V1R9 functions are present
		...1 ....		CVTH7740	"X'10'" HBB7740 functions are present
		.... 1...		CVTZOS_011000	"X'08'" z/OS V1R10 functions are present
		.... 1...		CVTZOS_V1R10	"X'08'" z/OS V1R10 functions are present
		.... 1...		CVTH7750	"X'08'" HBB7750 functions are present

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.... .1..		CVTZOS_011100	"X'04'" z/OS V1R11 functions are present
		.... .1..		CVTZOS_V1R11	"X'04'" z/OS V1R11 functions are present
		.... .1..		CVT_G64CPU_INFRASTRUCTURE	"X'04'" G64CPU Infrastructure present
		.... .1..		CVTH7760	"X'04'" HBB7760 functions are present
		.... ..1.		CVTZOS_011200	"X'02'" z/OS V1R12 functions are present
		.... ..1.		CVTZOS_V1R12	"X'02'" z/OS V1R12 functions are present
		.... ..1.		CVTH7770	"X'02'" HBB7770 functions are present
		.... ...1		CVTZOS_011300	"X'01'" z/OS V1R13 functions are present
		.... ...1		CVTZOS_V1R13	"X'01'" z/OS V1R13 functions are present
		.... ...1		CVTH7780	"X'01'" HBB7780 functions are present
1270	(4F6)	BITSTRING	1	CVTOSLV6	BYTE 6 OF CVTOSLVL
		1... ....		CVTZOS_020100	"X'80'" z/OS V2R1 functions are present
		1... ....		CVTZOS_V2R1	"X'80'" z/OS V2R1 functions are present
		1... ....		CVTH7790	"X'80'" HBB7790 functions are present
		.1.. ....		CVTZOS_020200	"X'40'" z/OS V2R2 functions are present
		.1.. ....		CVTZOS_V2R2	"X'40'" z/OS V2R2 functions are present
		.1.. ....		CVTPAUSEMULTIPLE	"X'40'" Pause Multiple
		.1.. ....		CVTH77A0	"X'40'" HBB77A0 functions are present
		..1. ....		CVTJ778H	"X'20'" JBB778H functions are present
		..1. ....		CVTZOS_V1R13_JBB778H	"X'20'" JBB778H functions are present
		..1. ....		CVTZOS_011300_JBB778H	"X'20'" JBB778H
		...1 ....		CVTZOS_020300	"X'10'" z/OS V2R3 functions are present
		...1 ....		CVTZOS_V2R3	"X'10'" z/OS V2R3 functions are present
		...1 ....		CVTH77B0	"X'10'" HBB77B0 functions are present
		.... 1...		CVTZOS_020400	"X'08'" z/OS V2R4 functions are present
		.... 1...		CVTZOS_V2R4	"X'08'" z/OS V2R4 functions are present
		.... 1...		CVTH77C0	"X'08'" HBB77C0 functions are present
		.... .1..		CVTZOS_020500	"X'04'" z/OS V2R5 functions are present
		.... .1..		CVTZOS_V2R5	"X'04'" z/OS V2R5 functions are present
		.... .1..		CVTH77D0	"X'04'" HBB77D0 functions are present
		.... ..1.		CVTZOS_030100	"X'02'" z/OS V3R1 functions are present
		.... ..1.		CVTZOS_V3R1	"X'02'" z/OS V3R1 functions are present
		.... ..1.		CVTH77E0	"X'02'" HBB77E0 functions are present
		.... ...1		CVTZOS_030200	"X'01'" z/OS V3R2 functions are present

Table 11. Structure CVTFIX (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
		.... ...1		CVTZOS_V3R2	"X'01'" z/OS V3R2 functions are present
		.... ...1		CVTH77F0	"X'01'" HBB77F0 functions are present
1271	(4F7)	BITSTRING	1	CVTOSLV7	BYTE 7 OF CVTOSLVL
1272	(4F8)	BITSTRING	1	CVTOSLV8	BYTE 8 OF CVTOSLVL
		1... ....		CVTPAUS3	"X'80'" IEA4xxxx
		.1.. ....		CVTPAUS4	"X'40'" Pause with checkpoint-OK
		..1. ....		CVTECT1	"X'20'" TIMEUSED ECT=YES with TIME_ON_CP, OFFLOAD_TIME, OFFLOAD_ON_CP
		...1 ....		CVT00CP	"X'10'" TIMEUSED with TIME_ON_CP and OFFLOAD_ON_CP
		.... 1...		CVTIEFOPZ	"X'08'" IEFOPZ
		.... .1..		CVTB00ST	"X'04'" Support for B00ST system parameter is available
		.... ..1.		CVT_CSVVOLDNS	"X'02'" Support for OutVolDNS in CSVQUERY is available
		.... ...1		CVT_RPILOCALLOCK	"X'01'" IEAVRPI/4RPI/VRPI2/4RPI2 support being called while holding LOCAL or CML lock
1273	(4F9)	BITSTRING	1	CVTOSLV9	BYTE 9 OF CVTOSLVL
1274	(4FA)	BITSTRING	1	CVTOSLVA	BYTE 10 OF CVTOSLVL
1275	(4FB)	BITSTRING	1	CVTOSLVB	BYTE 11 OF CVTOSLVL
1276	(4FC)	BITSTRING	1	CVTOSLVC	BYTE 12 OF CVTOSLVL
1277	(4FD)	BITSTRING	1	CVTOSLVD	BYTE 13 OF CVTOSLVL
1278	(4FE)	BITSTRING	1	CVTOSLVE	BYTE 14 OF CVTOSLVL
1279	(4FF)	BITSTRING	1	CVTOSLVF	BYTE 15 OF CVTOSLVL

Table 12. Structure CVTVSTGX

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	CVTVSTGX	
0	(0)	SIGNED	4	CVTBLDLS	- RESERVED - WAS STARTING ADDRESS OF BLDL LIST. MUST BE ZERO NOW.
4	(4)	SIGNED	4	CVTBLDLE	- RESERVED - WAS ENDING ADDRESS OF BLDL LIST. MUST BE ZERO NOW.
8	(8)	ADDRESS	4	CVTMLPAS	- STARTING VIRTUAL ADDRESS OF MLPA.
12	(C)	ADDRESS	4	CVTMLPAE	- ENDING VIRTUAL ADDRESS OF MLPA.
16	(10)	ADDRESS	4	CVTFLPAS	- STARTING VIRTUAL ADDRESS OF FLPA.
20	(14)	ADDRESS	4	CVTFLPAE	- ENDING VIRTUAL ADDRESS OF FLPA.
24	(18)	ADDRESS	4	CVTPLPAS	- STARTING VIRTUAL ADDRESS OF PLPA.
28	(1C)	ADDRESS	4	CVTPLPAE	- ENDING VIRTUAL ADDRESS OF PLPA.
32	(20)	ADDRESS	4	CVTRWNS	- STARTING VIRTUAL ADDRESS OF READ-WRITE NUCLEUS. (MDCXXX)
36	(24)	ADDRESS	4	CVTRWNE	- ENDING VIRTUAL ADDRESS OF READ-WRITE NUCLEUS. (MDCXXX)
40	(28)	ADDRESS	4	CVTRONS	- STARTING VIRTUAL ADDRESS OF READ-ONLY NUCLEUS. (MDCXXX)
44	(2C)	ADDRESS	4	CVTRONE	- ENDING VIRTUAL ADDRESS OF READ-ONLY NUCLEUS. (MDCXXX)
48	(30)	ADDRESS	4	CVTERWNS	- STARTING EXTENDED ADDRESS READ/WRITE NUCLEUS. @(DCR658)

Table 12. Structure CVTVSTGX (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
52	(34)	ADDRESS	4	CVTERWNE	- ENDING EXTENDED ADDRESS READ/WRITE NUCLEUS. @(DCR658)
56	(38)	ADDRESS	4	CVTEPLPS	- Starting virtual address of extended PLPA. This is an interface for accessing the LPAT only.
60	(3C)	ADDRESS	4	CVTEPLPE	- ENDING VIRTUAL ADDRESS OF EXTENDED PLPA.
64	(40)	ADDRESS	4	CVTEFLPS	- STARTING VIRTUAL ADDRESS OF EXTENDED FLPA.
68	(44)	ADDRESS	4	CVTEFLPE	- ENDING VIRTUAL ADDRESS OF EXTENDED FLPA.
72	(48)	ADDRESS	4	CVTEMLPS	- STARTING VIRTUAL ADDRESS OF EXTENDED MLPA.
76	(4C)	ADDRESS	4	CVTEMLPE	- ENDING VIRTUAL ADDRESS OF EXTENDED MLPA.

Table 13. Structure CVTXTNT1

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
0	(0)	STRUCTURE	0	CVTXTNT1	
0	(0)	ADDRESS	4	CVTFACHN	- ADDRESS OF CHAIN OF DCB FIELD AREAS (ISAM). ICB421
4	(4)	BITSTRING	8	CVT1R004	RESERVED

Table 14. Structure CVTXTNT2

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
0	(0)	STRUCTURE	0	CVTXTNT2	
0	(0)	BITSTRING	4	CVT2R000	RESERVED
4	(4)	CHARACTER	1	CVTNUCLS	- IDENTIFICATION OF THE NUCLEUS MEMBER NAME
5	(5)	BITSTRING	1	CVTFLGBT	- Flag byte. This byte is an interface only for bits CVTUNDzVM (CVTUNDVM), CVTzPDT CVTUNDAltVM, CVTzPDT & CVTMENV
		1... ....		CVTNPE	"X'80'" - INDICATES NON-PAGING ENVIRONMENT (VM HANDSHAKING) (OS/VS1) (MDC302)
		.1.. ....		CVTVME	"X'40'" - INDICATES MACHINE IS OPERATING IN VM ENVIRONMENT (OS/VS1) (MDC303)
		..1. ....		CVTBAH	"X'20'" - INDICATES THAT THE VM/370 - OS/VS1 BTAM AUTOPOLL HANDSHAKE IS OPERATIONAL (OS/VS1) (MDC356)
		...1 ....		CVTUNDZVM	"X'10'" - Running under z/VM (this is not the same as running under VICOM) Bit is set to 0 when CVTUNDAltVM is 1.
		...1 ....		CVTUNDVM	"X'10'" - Same as CVTUNDzVM
		.... 1...		CVTVICOM	"X'08'" - Running under VICOM
		.... .1..		CVTZPDT	"X'04'" - Running on zPDT (includes running on zD&T)

Table 14. Structure CVTXTNT2 (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.... ..1.		CVTUNDALTM	"X'02'" - Running under an alternate virtual machine environment. To determine which alternate virtual machine, the STSI output will list the control program identifier in field SI22V3DBCPIIdentifier mapped by CSRSIIDF and can be obtained by the CSRSI service. Bit is set to 0 when CVTUNDzVM is 1
		.... ...1		CVTMENV	"X'01'" - Running as a virtual machine environment. Bit is set to 1 when CVTUNDzVM or CVTUNDAlTM is set to 1
6	(6)	BITSTRING	2	CVTIOCID	- EBCDIC IDENTIFIER OF THE ACTIVE I/O CONFIGURATION SPECIFIED BY THE OPERATOR
8	(8)	ADDRESS	4	CVTDEBVR	"V(IFGDEBVR)" - ADDRESS OF BRANCH ENTRY POINT OF DEB VALIDITY CHECK ROUTINE (ICB380) XM9028
12	(C)	ADDRESS	4	CVTCVAF	- POINTER TO THE CVAF TABLE, WHICH CONTAINS THE CVAF BRANCH ENTRY ADDRESS AND NEXT VIB ADDRESS. (MDC410)
16	(10)	ADDRESS	4	CVTMMVT	"V(ICYMMVTC)" ADDRESS OF THE MEDIA MANAGER VECTOR TABLE (MDC410)
20	(14)	ADDRESS	4	CVTNCVP	ADDRESS OF CSA BUFFER POOL - USED BY NETWORK MANAGEMENT FACILITY (NMF)
24	(18)	ADDRESS	4	CVTQID(0)	- SAME AS CVTQIDA BELOW ICB381
24	(18)	BITSTRING	1		- RESERVED - FIRST BYTE OF CVTQID
25	(19)	ADDRESS	3	CVTQIDA	- ADDRESS OF QUEUE IDENTIFICATION (QID) TABLE PREFIX ICB381
28	(1C)	ADDRESS	4	CVTOLTEP	- POINTER TO CONTROL BLOCK CREATED BY SVC 59 TO POINT TO PSEUDO-DEB'S
32	(20)	BITSTRING	4	CVT2R020	- RESERVED
36	(24)	ADDRESS	4	CVTAVVT(0)	ADDRESS OF AVM CONTROL BLOCK OWNERSHIP: AVM SERIALIZATION: CS
		1... ....		CVTAVIN	"X'80'" INDICATES AVM INSTALLED
36	(24)	BITSTRING	4		
40	(28)	ADDRESS	4	CVTCCVT	- ADDRESS OF CRYPTOGRAPHIC FACILITY CVT (MDC370)
44	(2C)	ADDRESS	4	CVTSKTA	- ADDRESS OF STORAGE KEY TABLE (VM HANDSHAKING) (OS/VS1) (MDC304)
48	(30)	ADDRESS	4	CVTICB	- ADDRESS OF MASS STORAGE SYSTEM (MSS) CONTROL BLOCK (MDC305)
52	(34)	BITSTRING	1	CVTFBYT1	- FLAG BYTE
		1... ....		CVTRDE	"X'80'" - RELIABILITY DATA EXTRACTOR INDICATOR OWNERSHIP: DFP. SERIALIZATION: NONE.
53	(35)	BITSTRING	3	CVT2R035	- RESERVED
CVTLDT0 contains the offset value needed to adjust the TOD value to the local date and time of day. It is similar to CVTTZ except that CVTLDT0 is a doubleword value. CVTLDTOL and CVTTZ contain the same value.					
56	(38)	DBL WORD	8	CVTLDT0(0)	LOCAL TIME/DATE OFFSET
56	(38)	BITSTRING	4	CVTLDTOL	HIGH WORD
60	(3C)	BITSTRING	4	CVTLDTOR	LOW WORD
64	(40)	ADDRESS	4	CVTATCVT	- POINTER TO VTAM'S CVT



Table 14. Structure CVTXTNT2 (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
		1... ..		CVTATACT	"X'80'" IF ON, VTAM IS ACTIVE MDC081
68	(44)	BITSTRING	4	CVT2R044	- RESERVED
72	(48)	SIGNED	4	CVTBCLMT	- NUMBER OF 130-BYTE RECORDS SET ASIDE FOR BROADCAST MESSAGES
76	(4C)	SIGNED	4	CVT2R04C	RESERVED
80	(50)	DBL WORD	8	CVTLS0(0)	LEAP SECOND OFFSET IN TOD FORMAT
80	(50)	BITSTRING	4	CVTLS0H	HIGH WORD
84	(54)	BITSTRING	4	CVTLS0L	LOW WORD
88	(58)	BITSTRING	44	CVT2R058	RESERVED
END OF CVT					

Table 15. Cross Reference for CVT

Name	Offset	Hex	Tag
CVT_CSVVOLDN	4F8		2
CVT_G64CPU_INFRASTRUCTURE	4F5		4
CVT_LLACOPY_DEBLOCKEXCLOK	1AD		40
CVT_RPILOCALLOCK	4F8		1
CVTABEND	C8		
CVTADV	3B8		
CVTAIDVT	2EC		
CVTALR	179		2
CVTALRS	4F4		8
CVTAMFF	108		0
CVTAPF	144		
CVTAPFA	145		
CVTAPFT	278		
CVTAPG	18F		7
CVTAPR	B6		40
CVTAPTHR	B7		8
CVTAQAVB	F1		
CVTAQAVT	F0		
CVTAQTOP	224		
CVTASCBH	234		
CVTASCBL	238		
CVTASCTII	B6		2
CVTASCRF	1B4		
CVTASCRL	1B8		
CVTASMT	2C0		
CVTASVT	22C		
CVTATACT	40		80

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTATCVT	40	
CVTAUSCB	C	
CVTAUTHL	1E4	
CVTAVIN	24	80
CVTAVVT	24	
CVTBAH	5	20
CVTBCLMT	48	0
CVTBFP	4F2	10
CVTBFPH	179	10
CVTBLDCP	1E8	
CVTBLDLE	4	0
CVTBLDLS	0	0
CVTB00ST	4F8	4
CVTBRET	52	
CVTBSM0F	1AE	B0F
CVTBSM2	1DE	B02
CVTBTERM	34	
CVTBUF	10	
CVTCADS	4F0	4
CVTCALL	44A	
CVTCBBR	3CC	
CVTCBSP	100	
CVTCCH	B6	80
CVTCCVT	28	
CVTCDEQ	3D8	
CVTCGK	3E4	
CVTCKRAS	4C0	
CVTCMPSC	179	80
CVTCMPSH	179	40
CVTCPBDB	480	
CVTCPDLB	484	
CVTCPFRE	470	
CVTCPGTC	46C	
CVTCPGUB	464	
CVTCPGUP	468	
CVTCPGUS	494	
CVTCPUASALIASTOCORE	17A	10
CVTCRAS	1FC	
CVTCRMN	1F8	

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTCRPTL	4F0	2
CVTCSD	294	
CVTCSOAL	39C	
CVTCSOAL	4CC	
CVTCSOCR	4D4	
CVTCSOCS	4D8	
CVTCSOMF	4C8	
CVTCSPIE	1CC	
CVTCSRIM	4B4	80
CVTCSRSI	4F4	80
CVTCSRT	220	
CVTCSRUN	4F4	20
CVTCSTR	178	4
CVTCTIMS	B7	40
CVTCTLFG	18E	10
CVTCTLFGDIAG	18E	60
CVTCTTVT	3F8	
CVTCUCB	64	
CVTCUSE	178	20
CVTCVAF	C	
CVTCVT	60	40C3E5E3
CVTDAIR	2DC	
CVTDAIRX	90	
CVTDARCM	36C	
CVTDATE	38	0
CVTDCB	74	9B
CVTDCBA	75	
CVTDCPA	17B	10
CVTDCQA	4E0	
CVTDDR	B6	20
CVTDEBVR	8	
CVTDELCF	1F4	
CVTDFA	4C0	
CVTDICOM	168	80
CVTDIRST	168	0
CVTDMSR	110	
CVTDMSRA	111	
CVTDMSRF	110	0
CVTDOFFE	48C	

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTDOFFS	488	
CVTDPUR	43C	
CVTDQIQE	298	
CVTDRMOD	18E	8
CVTDSSAC	1D6	0
CVTDSTAT	18E	10
CVTDYAPF	4F1	80
CVTDYLPA	4F2	80
CVTDYNEX	4F1	20
CVTDYNL	4F1	4
CVTECT1	4F8	20
CVTECVT	8C	
CVTEDAT	179	1
CVTEDAT2	17B	1
CVTEEC	F4	40
CVTEFF02	3D0	
CVTEFLPE	44	
CVTEFLPS	40	
CVTEHCIR	2E4	
CVTEHDEF	2E0	
CVTEMLPE	4C	
CVTEMLPS	48	
CVTENCLV	4F1	10
CVTENFCT	C0	
CVTEORM	138	
CVTEPLPE	3C	
CVTEPLPS	38	
CVTERWNE	34	
CVTERWNS	30	
CVTESA	4F0	80
CVTESAME	17A	80
CVTEVENT	3C4	
CVTEXCL	32C	
CVTEXTIT	50	
CVTEXPRO	260	
CVTEXP1	398	
CVTEXSNL	18C	0
CVTEXSNR	188	
CVTEXT1	FC	

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTEXT2	148	
CVTEXT2A	149	
CVTE51GN	1AC	80
CVTE51LN	1AC	40
CVTFACHN	0	
CVTFBOSV	84	
CVTFBYT1	34	0
CVTFETCH	2D0	
CVTFIX	-100	
CVTFLAGS	178	
CVTFLAG1	178	0
CVTFLAG2	179	80
CVTFLAG3	17A	0
CVTFLAG3DIAG	17A	8
CVTFLAG3RSVD	17A	4
CVTFLAG4	17B	80
CVTFLAG5	F4	40
CVTFLAG6	F5	0
CVTFLAG7	F6	0
CVTFLAG8	F7	0
CVTFLAG9	1AD	41
CVTFLGBT	5	0
CVTFLGCS	5C	
CVTFLGC0	5C	0
CVTFLGC1	5D	0
CVTFLPAE	14	
CVTFLPAS	10	
CVTFP	B7	4
CVTFQCB	280	0
CVTFRAS	204	
CVTFRECL	1F0	
CVTFUNC	4A8	0
CVTGDA	230	
CVTGETCL	1EC	
CVTGETL	15C	
CVTGLMN	2A8	
CVTGMBR	350	
CVTGRRGN	498	
CVTGRSST	1AC	0

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTGSDA	3B4	
CVTGSDAB	3B4	80
CVTGSF	17A	1
CVTGSMQ	264	
CVTGTF	EC	
CVTGTFA	ED	
CVTGTFAV	EC	80
CVTGTFR8	384	
CVTGTFST	EC	
CVTGTRCE	18E	2
CVTGVRGN	49C	
CVTGV	1B0	
CVTGXL	118	
CVTHID	42C	
CVTHIPER	4F0	10
CVTHJES	14C	
CVTHJESA	14D	
CVTHSM	3DC	
CVTH3310	4F0	80
CVTH4410	4F0	8
CVTH4420	4F0	40
CVTH4430	4F1	80
CVTH5510	4F1	20
CVTH5520	4F1	10
CVTH5530	4F1	4
CVTH6601	4F1	2
CVTH6603	4F1	4
CVTH6605	4F2	40
CVTH6606	4F2	20
CVTH6608	4F2	4
CVTH6609	4F2	2
CVTH77A0	4F6	40
CVTH77B0	4F6	10
CVTH77C0	4F6	8
CVTH77D0	4F6	4
CVTH77E0	4F6	2
CVTH77F0	4F6	1
CVTH7703	4F2	1
CVTH7705	4F3	10

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTH7706	4F3	8
CVTH7707	4F3	4
CVTH7708	4F3	2
CVTH7709	4F3	1
CVTH7720	4F5	40
CVTH7730	4F5	20
CVTH7740	4F5	10
CVTH7750	4F5	8
CVTH7760	4F5	4
CVTH7770	4F5	2
CVTH7780	4F5	1
CVTH7790	4F6	80
CVTICB	30	
CVTICHPT	4D0	
CVTICPID	5E	0
CVTIDEVN	41C	
CVTIEFOPZ	4F8	8
CVTIFAR	4F4	1
CVTIFAWICAVAILABLE	1AD	2
CVTIFAWICINSTALLED	1AD	1
CVTIHASU	414	
CVTIHVP	124	
CVTILM	4F4	10
CVTIOBP	2C4	
CVTIOCID	6	0
CVTIONLV	4A0	0
CVTIPCDS	2F0	
CVTIPCRI	2F4	
CVTIPCRP	2F8	
CVTIQD	179	4
CVTIRECM	368	
CVTISNBR	334	
CVTIXAVL	7C	
CVTJESCT	128	
CVTJTERM	3FC	
CVTJ3311	4F0	10
CVTJ3313	4F0	20
CVTJ4422	4F0	1
CVTJ5522	4F1	8

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTJ6602	4F1	1
CVTJ6604	4F2	80
CVTJ6607	4F2	8
CVTJ6609	4F2	2
CVTJ7713	4F3	20
CVTJ778H	4F6	20
CVTKPAR	178	1
CVTLCCAT	300	
CVTLDEV	F0	40
CVTLDT0	38	
CVTLDTOL	38	0
CVTLDTOR	3C	0
CVTLEVL	-2	
CVTLFRM	34C	
CVTLINK	8	
CVTLKR	4F0	8
CVTLKRM	140	
CVTLKRMA	290	
CVTLLCB	24	
CVTLLTA	4DC	
CVTLLTRM	28	
CVTLNKSC	D6	
CVTLPARC	4F3	20
CVTLPDIA	168	
CVTLPDIR	169	
CVTLPDSR	160	
CVTLQCB	284	0
CVTLSCH	3D4	
CVTLSEQ	268	
CVTLS0	50	
CVTLS0H	50	0
CVTLS0L	54	0
CVTMAP	0	100
CVTMAXMP	1DC	1
CVTMCHPR	134	
CVTMDL	-6	
CVTMDLDS	D0	
CVTMFACT	320	80
CVTMFCTL	31C	



Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTMFRTR	320	
CVTMLPAE	C	
CVTMLPAS	8	
CVTMMVT	10	
CVTMSER	94	
CVTMSFCB	428	
CVTMSFRM	33C	
CVTMSLT	3C	
CVTMULNF	5C	80
CVTMULTICPUSPERCORE	17A	20
CVTMVPG	178	10
CVTMVSE	74	80
CVTMVS2	74	1
CVTMZ00	A4	7FFFFFFF
CVTNCPV	14	
CVTNIP	B6	10
CVTNLOG	B7	10
CVTNPAF	F4	20
CVTNOMP	18E	4
CVTNPE	5	80
CVTNUCB	80	
CVTNUCLS	4	40
CVTNUCLU	4EC	
CVTNUCMP	4B0	
CVTNUMB	-4	
CVTNVT0	4C4	
CVTNWTCM	F0	20
CVTOLTEP	1C	
CVTOLT0A	1C4	
CVT00CP	4F8	10
CVTOPCTP	25C	
CVTOPTA	B6	B2
CVTOPTB	B7	30
CVTOPTE	218	
CVT0SEXT	74	8
CVT0SLVA	4FA	0
CVT0SLVB	4FB	0
CVT0SLVC	4FC	0
CVT0SLVD	4FD	0

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTOSLVE	4FE	0
CVTOSLVF	4FF	0
CVTOSLVL	4F0	
CVTOSLV0	4F0	FF
CVTOSLV1	4F1	FF
CVTOSLV2	4F2	FF
CVTOSLV3	4F3	FF
CVTOSLV4	4F4	AF
CVTOSLV5	4F5	7F
CVTOSLV6	4F6	FF
CVTOSLV7	4F7	0
CVTOSLV8	4F8	F7
CVTOSLV9	4F9	0
CVTOS390	4F1	2
CVTOS390_R1	4F1	2
CVTOS390_R10	4F2	1
CVTOS390_R2	4F1	1
CVTOS390_R3	4F1	4
CVTOS390_R4	4F2	80
CVTOS390_R5	4F2	40
CVTOS390_R6	4F2	20
CVTOS390_R7	4F2	8
CVTOS390_R8	4F2	4
CVTOS390_R9	4F2	2
CVTOS390_010100	4F1	2
CVTOS390_010200	4F1	1
CVTOS390_010300	4F1	4
CVTOS390_010400	4F2	80
CVTOS390_020400	4F2	80
CVTOS390_020500	4F2	40
CVTOS390_020600	4F2	20
CVTOS390_020700	4F2	8
CVTOS390_020800	4F2	4
CVTOS390_020900	4F2	2
CVTOS390_021000	4F2	1
CVTOVER	178	8
CVTPARMC	4F1	1
CVTPARRL	274	
CVTPARS	20C	

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTPATCH	DC	
CVTPAUSE	4F3	80
CVTPAUSEMULTIPLE	4F6	40
CVTPAUS2	4F3	40
CVTPAUS3	4F8	80
CVTPAUS4	4F8	40
CVTPCCAT	2FC	
CVTPCNVT	1C	
CVTPERFM	2D8	
CVTPER2	179	8
CVTPGSR	45C	
CVTPLPAE	1C	
CVTPLPAS	18	
CVTPPGMX	1A8	
CVTPQAP	F7	80
CVTPRDED	4F1	2
CVTPRLTV	20	
CVTPROCASCORE	17A	40
CVTPROD	-28	
CVTPRODI	-20	40404040
CVTPRODN	-28	40404040
CVTPROT	B7	80
CVTPSXM	430	
CVTPTGT	1D0	
CVTPTRV	120	
CVTPTRV3	13C	
CVTPUMA	4F5	80
CVTPURG	104	
CVTPURGA	105	
CVTPUTL	1BC	
CVTPVBP	318	
CVTPVTP	164	
CVTPWI	314	
CVTP001A	17B	40
CVTP001I	17B	80
CVTP002	17B	8
CVTP002C	17B	4
CVTQABST	D4	
CVTQCDSR	B8	

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTQCS01	27C	
CVTQID	18	
CVTQIDA	19	
CVTQLPAQ	BC	
CVTQMSG	10C	
CVTQMWR	B0	
CVTQOCR	AC	
CVTQSAS	200	
CVTQTD00	6C	
CVTQTE00	68	
CVTQUIS	210	
CVTQV1	3A8	
CVTQV2	3AC	
CVTQV3	3B0	
CVTRAC	3E0	
CVTRBCB	16C	
CVTRCEP	490	
CVTRCP2B	152	0
CVTRDE	34	80
CVTREAL	11C	
CVTRECON	380	
CVTRECRQ	2BC	
CVTRELNO	-4	
CVTRENQ	288	
CVTRI	17B	2
CVTRLSTG	358	0
CVTRMBR	348	
CVTRMPMT	394	
CVTRMPTT	390	
CVTRMS	E0	
CVTRMT	408	
CVTRNIO	EC	2
CVTRONE	2C	
CVTRONS	28	
CVTRPOST	29C	
CVTRSCN	194	
CVTRSMWD	178	80
CVTRSPIE	28C	
CVTRSTCI	2CC	0

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTRSTRI	2CE	0
CVTRSTRS	151	0
CVTRSTWD	2CC	
CVTRSTW2	150	
CVTRSUME	400	
CVTRS1D7	1D7	0
CVTRS12C	12C	0
CVTRS150	150	0
CVTRS170	170	0
CVTRS180	180	0
CVTRS26C	26C	0
CVTRS3A4	3A4	0
CVTRS3C0	3C0	0
CVTRS360	360	0
CVTRS374	374	0
CVTRS4A1	4A1	0
CVTRS4B5	4B5	0
CVTRS444	444	
CVTRTLS	4F2	80
CVTRTMCT	23C	
CVTRTMS	250	
CVTRT03	17C	
CVTRWNE	24	
CVTRWNS	20	
CVTSAF	F8	
CVTSCAN	1E0	
CVTSCBP	248	
CVTSCPIN	340	
CVTSDBF	24C	80000000
CVTSDMP	244	
CVTSDRM	21C	
CVTSDTRC	18E	1
CVTSDUMP	110	80
CVTSFR	114	
CVTSFV	418	
CVTSHRVM	1A0	
CVTSIC	258	
CVTSIGPT	1D4	1E
CVTSKTA	2C	

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTSLID	175	
CVTSLIDA	174	
CVTSLIST	474	
CVTSLLOC	47C	
CVTSMCA	C4	
CVTSMEXT	4AC	
CVTSMFEX	1C8	
CVTSMFSP	424	
CVTSMF83	420	
CVTSNAME	154	40404040
CVTSNCTR	B4	0
CVTSOLED	F5	80
CVTSOLN	4A8	80
CVTSOPF	179	20
CVTSPD	EC	40
CVTSPDMC	1D5	0
CVTSPDME	E4	
CVTSPFRR	35C	
CVTSPIP	4BC	
CVTSPOST	2C8	
CVTSPSA	2AC	
CVTSPVLK	18D	
CVTSRBRT	1C0	
CVTSREGN	478	
CVTSRM	3E8	
CVTSSAP	2E8	
CVTSSCR	3C8	
CVTSSRB	3A0	
CVTSTB	70	
CVTSTCK	1D8	
CVTSTXU	214	
CVTSUBSP	178	2
CVTSUP	4F4	2
CVTSUSP	410	
CVTSVDCB	54	
CVTSVPRC	178	40
CVTSVT	364	
CVTSV60	240	
CVTSV76M	78	0

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTSYLK	174	0
CVTSYLKR	174	0
CVTSYLKS	174	FF
CVTSYSAD	30	
CVTSYSPLEXZAAPONLINE	F6	80
CVTSYSPLEXZCBPONLINE	F6	40
CVTS1EE	208	
CVTTAS	198	
CVTTCASP	3F4	
CVTTCBP	0	
CVTTCMFG	F0	0
CVTTCRDY	F0	80
CVTTCTL	404	
CVTTOCP	4F4	4
CVTTOD	B7	20
CVTTPC	58	
CVTTPIO	3BC	
CVTTPIOS	254	
CVTTPUR	438	
CVTTRCRM	19C	
CVTTRPOS	440	
CVTTSCE	D8	
CVTTSKI	460	
CVTTVT	9C	
CVTTX	17B	8
CVTTXC	17B	4
CVTTXJ	18E	80
CVTTXTE	18E	80
CVTTZ	130	0
CVTT6SVC	40C	
CVTUCBA	4E4	
CVTUCBSC	434	
CVTUCBSV	4F0	8
CVTUDUMP	110	40
CVTUNDALTM	5	2
CVTUNDVM	5	10
CVTUNDZVM	5	10
CVTUNICS	4F4	40
CVTUSER	CC	

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTUSR	EC	4
CVTVACR	37C	
CVTVEAC0	2A4	
CVTVEF	F4	80
CVTVRID	-18	40404040
CVTVESTU	4E8	
CVTVFCB	458	
CVTVFGET	450	
CVTVFIND	44C	
CVTVFMEM	454	
CVTVICOM	5	8
CVTVME	5	40
CVTMENV	5	1
CVTVPRM	4C	
CVTVPSA	38C	
CVTVPSIB	324	
CVTVPSM	4E	0
CVTVSI	328	
CVTVSS	4C	0
CVTVSTGX	0	
CVTVSTOP	388	
CVTVS1A	B7	2
CVTVS1B	B7	1
CVTVTAM	4B8	
CVTVVMDI	228	
CVTVWAIT	270	
CVTV64	4F3	10
CVTWARNUND	B6	8
CVTWLM	4F1	40
CVTWSAC	2B8	
CVTWSAG	2B4	
CVTWSAL	2B0	
CVTWSMA	344	
CVTWSPR	EC	20
CVTWTCB	378	
CVTXAFL	4B4	0
CVTXAPG	14	
CVTXAX	4F0	80
CVTXCPCT	448	1F4



Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTXEXTR	338	
CVTXITP	44	
CVTXPPF	B6	1
CVTXSFT	304	
CVTXSTKN	30C	
CVTXSTKS	308	
CVTXTLER	2C	
CVTXTNT1	0	
CVTXTNT2	0	
CVTXUNSN	330	
CVTXUNSS	310	
CVTZARCH	17A	80
CVTZCBP	17A	2
CVTZDTAB	40	
CVTZIIP	4F4	2
CVTZNALC	17B	20
CVTZOS	4F3	20
CVTZOS_V1R1	4F3	20
CVTZOS_V1R10	4F5	8
CVTZOS_V1R11	4F5	4
CVTZOS_V1R12	4F5	2
CVTZOS_V1R13	4F5	1
CVTZOS_V1R13_JBB778H	4F6	20
CVTZOS_V1R2	4F3	10
CVTZOS_V1R3	4F3	8
CVTZOS_V1R4	4F3	4
CVTZOS_V1R5	4F3	2
CVTZOS_V1R6	4F3	1
CVTZOS_V1R7	4F5	40
CVTZOS_V1R8	4F5	20
CVTZOS_V1R9	4F5	10
CVTZOS_V2R1	4F6	80
CVTZOS_V2R2	4F6	40
CVTZOS_V2R3	4F6	10
CVTZOS_V2R4	4F6	8
CVTZOS_V2R5	4F6	4
CVTZOS_V3R1	4F6	2
CVTZOS_V3R2	4F6	1
CVTZOS_010100	4F3	20

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVTZOS_010200	4F3	10
CVTZOS_010300	4F3	8
CVTZOS_010400	4F3	4
CVTZOS_010500	4F3	2
CVTZOS_010600	4F3	1
CVTZOS_010700	4F5	40
CVTZOS_010800	4F5	20
CVTZOS_010900	4F5	10
CVTZOS_011000	4F5	8
CVTZOS_011100	4F5	4
CVTZOS_011200	4F5	2
CVTZOS_011300	4F5	1
CVTZOS_011300_JBB778H	4F6	20
CVTZOS_020100	4F6	80
CVTZOS_020200	4F6	40
CVTZOS_020300	4F6	10
CVTZOS_020400	4F6	8
CVTZOS_020500	4F6	4
CVTZOS_030100	4F6	2
CVTZOS_030200	4F6	1
CVTZOSAS	4F5	80
CVTZOSE	4F5	80
CVTZPDT	5	4
CVTZ1	F4	80
CVT0DS	88	
CVT0EF00	4	
CVT0EF01	48	
CVT0PT0E	3EC	
CVT0PT01	98	
CVT0PT02	370	
CVT0PT03	3F0	
CVT0SCR1	E8	
CVT0TC0A	354	
CVT0VL00	18	
CVT0VL01	1A4	
CVT040ID	A0	0
CVT044R2	2D4	
CVT062R1	2A0	
CVT1EF00	A8	

Table 15. Cross Reference for CVT (continued)

Name	Offset	Hex Tag
CVT1R004	4	0
CVT1SSS	74	40
CVT121TR	B6	4
CVT2R000	0	0
CVT2R020	20	0
CVT2R035	35	0
CVT2R04C	4C	0
CVT2R044	44	0
CVT2R058	58	0
CVT2SPS	74	20
CVT4MPS	74	4
CVT4MS1	74	10
CVT6DAT	74	2
CVT8A0S2	74	12

## IFAWICCB information

### IFAWICCB programming interface information

IFAWICCB is a programming interface.

### IFAWICCB heading information

**Common name:** SMF Workload Interaction Correlator (WIC) constants and exploiter exit control blocks

**Macro ID:** IFAWICCB

**DSECT name:** WICPARM

**Owning component:** System Management Facilities (SC100)

**Eye-catcher ID:** None

**Storage attributes:** Subpool: 245  
Key: 0  
Residency: ESQA

**Size:** Size of some DSECTs may vary from release to release.  
WicWR\_AggBucketParmList -- X'0018' bytes  
WicWR\_AggBucket -- X'0002' bytes  
WicWR\_ExcBucket -- X'000A' bytes  
Wic\_Warning\_Rsn -- X'0004' bytes  
WicEnf -- X'0010' bytes  
WicEnfQual -- X'0004' bytes  
WicNB\_ParmList -- X'0030' bytes  
WicCB\_ParmList -- X'0014' bytes  
WicWR\_ParmList -- X'0068' bytes  
WicParm -- X'0030' bytes

**Created by:** SMF (IFAHFTSK)

**Pointed to by:** R1 passed to WIC exploiter exits

**Serialization:** None.

**Function:** This macro defines data structures for the SMF WIC exploiter exit, parameters and constants for the WIC exit services, and constants for the IFAWIC service.

## IFAWICCB mapping

Table 16. Structure WICPARM

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	WICPARM	
0	(0)	SIGNED	2	WICPARM_VERSION	Version of WicParm
2	(2)	SIGNED	2	WICPARM_LENGTH	Length of WicParm
4	(4)	SIGNED	2	WICPARM_SUBTYPE	Subtype number
6	(6)	SIGNED	2	WICPARM_REQUEST	Request for the WIC exploiter exit. If called with an unrecognized request number, the exit routine should return without any processing.
8	(8)	ADDRESS	8	WICPARM_DYNAREAPTR	Dynamic area for WIC exploiter exit. Must be accessed in amode 64
16	(10)	SIGNED	4	WICPARM_DYNAREALEN	Length of storage pointed to by WicParm_DynAreaPtr. At least 1M in size.
20	(14)	CHARACTER	4		Reserved for alignment

Address of WIC exit services  
Use these addresses to perform "common" WIC exit services.  
AMODE: 64-bit  
ASC Mode: Primary  
Input Registers: GR0 Irrelevant  
                  GR1 Address of the parameter list  
                                as shown below  
                  GR2-GR12 Irrelevant  
                  GR13 Address of 144 byte save area  
                  GR14 Return address  
                  GR15 Irrelevant  
Input Parameter List: Either WicNB\_Parmlist,  
                                WicCB\_Parmlist, or WicWR\_Parmlist.  
                                Pointed to by GR1, each WIC exit  
                                service has three parameters as  
                                shown.

0 63  
+-----+  
| Double Word 1 Address of WIC exit service parameter |  
| list (see Wic\_ParmList for the given WIC exit |  
| service) |  
+-----+  
| Double Word 2 Address of Return Code |  
+-----+  
| Double Word 3 Address of Reason Code |  
+-----+

Output Registers: GR0 Used by WIC exit services  
                  GR1-GR14 Unchanged  
                  GR15 Used by WIC exit services  
Recovery Operation: Provided by the caller

24	(18)	ADDRESS	8	WICPARM_WICSRV_GETNEXTBUFFERPTR	Address of the get next buffer WIC exit service IFAWGetNextBuffer
32	(20)	ADDRESS	8	WICPARM_WICSRV_CLEARBUFFERPTR	Address of the clear buffer WIC exit service IFAWClearBuffer
40	(28)	ADDRESS	8	WICPARM_WICSRV_WRITESMF98PTR	Address of the build and write SMF98.x record WIC exit service IFAWBuildAndWriteSMF98

Table 16. Structure WICPARM (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
40	(28)	X'1'	0	WICPARM_REQPRIME	"1" Request for the WIC exploiter exit to prime. Prepare for writing a record on the next HFTSINTVL by clearing the WIC instrumentation buffer data. Do not produce a record this interval.
40	(28)	X'2'	0	WICPARM_REQWRITERECORD	"2" Request for the WIC exploiter exit to write an SMF 98 record for it's subtype.
40	(28)	X'1'	0	WICPARM_KVERSIONFIRST	"1"
40	(28)	X'1'	0	WICPARM_KVERSIONCURRENT	"1"
Miscellaneous constants					
40	(28)	BITSTRING	0	WIC_RSN_CODEMASK	"X'0000FFFF'" Mask to locate the non-diagnostic portion of the reason code
40	(28)	X'10'	0	WIC_KMAX4KPAGES	"16" Max allowed value for BUFFER4KPAGES
40	(28)	X'2000'	0	WIC_KDYNAMICAREALENGTH	"8192" Size of common dynamic storage (SP 245) allocated for the use of WIC trend exploiter exit routines. This area is pointed to by R0 on entry to the WIC exploiter exit
40	(28)	X'2000'	0	WIC_KFIRSTTESTSUBTYPE	"8192" First test subtype number
40	(28)	X'2007'	0	WIC_KLASTTESTSUBTYPE	"8199" Last test subtype number
IFAWIC Return and Reason Code definitions					
		.... ..		IFAWIC_RC_SUCCESS	"X'00000000'" Meaning: IFAWIC request successful. For REQUEST=REGISTER a BufferPtr has been provided to the current primary address. If an ExitVersion has been specified, and this is the first REQUEST=REGISTER on the system for the input subtype, or the input ExitVersion is larger than the current ExitVersion the input subtype, the ExitRoutine has been registered, and the system will switch to calling the new exit routine at the next WIC interval. The system issues ENF 85 signals to inform IFAWIC callers to start or stop writing instrumentation data to their instrumentation buffer. See macro IFAWICCB for additional information. For REQUEST=Deregister the buffer is made unavailable to the current primary address. By invoking REQUEST=Deregister, the program certifies that the exploiter has stopped instrumenting into its buffer. The program does not have to certify its WIC exit is not accessing the buffer. Once REQUEST=Deregister is invoked, the system is responsible for preventing the WIC exit from accessing the the buffer If there are no more address spaces registered for the input subtype, the system will stop calling the exit routine. Action: For REQUEST=REGISTER, the program should begin instrumenting. For REQUEST=Deregister, none.

Table 16. Structure WICPARM (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.... .1..		IFAWIC_RC_WARNING	"X'00000004'" Meaning: IFAWIC request completed with a warning. The high half word of the reason code may contain indications of other warnings that have occurred besides the reason code portion in the lower half. See Wic_Warning_Rsn in IFAWICCB for a mapping of this area. For REQUEST=REGISTER, a BufferPtr has been provided to the current primary address. If an ExitVersion has been specified, and this is the first REQUEST=REGISTER on the system for the input subtype, or the input ExitVersion is larger than the current ExitVersion the input subtype, the ExitRoutine has been registered, and the system will switch to calling the new exit routine at the next WIC interval. For REQUEST=Deregister, the buffer is made unavailable to the current primary address. If there are no more address spaces registered for the input subtype, the system will stop calling the exit routine. Action: Refer to the action provided with the specific reason code. For REQUEST=REGISTER, the program should check the high half of the reason code for any conditions indicating the WIC services are not fully available by testing for any bits being on in the first byte using the IFAWICCB field Wic_NotFullyAvail_LowHighMask. When no bits in the first byte of the reason code are on, the program should start writing instrumentation data to the instrumentation buffer because the program's WIC exit will be called at the next WIC interval. When any bit in the first byte of the reason code is on, the program should not start writing instrumentation data to the instrumentation buffer because the WIC exit will not be called at the next WIC interval. The system issues ENF 85 signals to inform IFAWIC callers to start or stop writing instrumentation data to their instrumentation buffer. See macro IFAWICCB for additional information. For REQUEST=Deregister, none.
40		(28) BITSTRING	0	IFAWIC_RSN_BUFFER4KPAGESSMALLER	

Table 16. Structure WICPARM (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
					"X'00000401'" Meaning: A smaller Buffer4kPages value was requested than the value specified on a previous REQUEST=REGISTER for the same subtype. The system reserves the requested buffer size for this address space. Action: It is desirable for the Buffer4kPages to be the same for all register requests for the input subtype, however, the system will allow the value to change. When specifying different Buffer4kPages values, it is up to the caller to remember how much data in each address space WIC instrumentation buffer is available for the WIC exit routine to use. Also, keep in mind when aggregating data that new data fields may not be available for aggregation from older program versions. Record data in the WIC instrumentation buffer to identify to the WIC ExitRoutine what data it should access and what fields it can aggregate.
40	(28)	BITSTRING	0	IFAWIC_RSN_BUFFER4KPAGESLARGER	"X'00000402'" Meaning: A larger Buffer4kPages value was requested than the value specified on a previous REQUEST=REGISTER for the same subtype. The system reserves the requested buffer size for this address space. Action: It is desirable for the Buffer4kPages to be the same for all register requests for the input subtype, however, the system will allow the value to change. When specifying different Buffer4kPages values, it is up to the caller to remember how much data in each address space WIC instrumentation buffer is available for the WIC exit routine to use. Also, keep in mind when aggregating data that new data fields may not be available for aggregation from older program versions. Record data in the WIC instrumentation buffer to identify to the WIC ExitRoutine what data it should access and what fields it can aggregate.
40	(28)	BITSTRING	0	IFAWIC_RSN_ALREADYREGISTERED	"X'00000403'" Meaning: A REQUEST=REGISTER was requested from an address space for a subtype that was already registered. Action: none needed.
40	(28)	BITSTRING	0	IFAWIC_RSN_NOTREGISTERED	"X'00000404'" Meaning: A REQUEST=DEREGISTER was requested from an address space for a subtype that was not currently registered. Action: none needed.
40	(28)	BITSTRING	0	IFAWIC_RSN_EXITVERSIONIGNORED	"X'00000405'" Meaning: A REQUEST=REGISTER specified an ExitVersion that was less than the current ExitVersion. The ExitVersion and ExitRoutine parameters are ignored and the system continues using the current highest versioned exit routine. Action: none needed.

Table 16. Structure WICPARM (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
40	(28)	BITSTRING	0	IFAWIC_RSN_WICFEATURENOTENABLED	"X'00000481" Meaning: The WorkloadIntCorr feature was not enabled to product registry. Action: Wait for ENF 85 signal and verify the WicEnf bit WicEnf_InstrumentationRequested is on before writing to the instrumentation buffer.
40	(28)	BITSTRING	0	IFAWIC_RSN_SUBTYPENOTCOLLECTED	"X'00000482" Meaning: SMF has not been configured to collect the input SMF 98 subtype record. Action: Wait for ENF85 signal and verify the WicEnf bit WicEnf_InstrumentationRequested is on before writing to the instrumentation buffer.
	.... 1...			IFAWIC_RC_USERERROR	"X'00000008" Meaning: The IFAWIC request specified parameters that are not valid or the request is issued in an user-controllable environment that is not valid. The IFAWIC service did not complete successfully. For REQUEST=REGISTER, no WIC instrumentation buffer was provided to the caller. Action: Refer to the action provided with the specific reason code.
40	(28)	BITSTRING	0	IFAWIC_RSN_PARMLISTALET	"X'00000801" Meaning: Unable to use ALET of the IFAWIC parameter list. Action: Provide a valid ALET for the parameter list. The access register might not have been set up correctly.
40	(28)	BITSTRING	0	IFAWIC_RSN_BADVERSION	"X'00000802" Meaning: The provided PLISTVER is not valid. This suggests the parameter list used to call IFAWIC was overlaid, or was not generated by the IFAWIC macro. Action: Check for possible storage overlay of the parameter list. Use the IFAWIC service to produce the parameter list.
40	(28)	BITSTRING	0	IFAWIC_RSN_RESERVEDNOT0	"X'00000803" Meaning: The parameter list contains non-0 input in reserved fields. This suggests the parameter list used to call IFAWIC was overlaid, or was not generated by the IFAWIC macro. Action: Check for possible storage overlay of the parameter list. Use the IFAWIC service to produce the parameter list.
40	(28)	BITSTRING	0	IFAWIC_RSN_BADREQUESTTYPE	"X'00000804" Meaning: The request type is not valid. This suggests the parameter list used to call IFAWIC was overlaid, or was not generated by the IFAWIC macro. Action: Check for possible storage overlay of the parameter list. Use the IFAWIC service to produce the parameter list.
40	(28)	BITSTRING	0	IFAWIC_RSN_BADBUFFERKEY	"X'00000805" Meaning: The buffer key type is not valid. This suggests the parameter list used to call IFAWIC was overlaid, or was not generated by the IFAWIC macro. Action: Check for possible storage overlay of the parameter list. Use the IFAWIC service to produce the parameter list.



Table 16. Structure WICPARM (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
40	(28)	BITSTRING	0	IFAWIC_RSN_PARMLISTFETCH	"X'00000807'" Meaning: An error was encountered when fetching the supplied parameter list. Action: Call IFAWIC with a parameter list properly addressable.
40	(28)	BITSTRING	0	IFAWIC_RSN_PARMLISTWRITE	"X'00000808'" Meaning: An error was encountered when writing to the supplied parameter list. Action: Call IFAWIC with a parameter list properly addressable and able to be written to.
40	(28)	BITSTRING	0	IFAWIC_RSN_NOTENABLED	"X'00000810'" Meaning: The IFAWIC caller was not enabled. Action: Call IFAWIC only when enabled.
40	(28)	BITSTRING	0	IFAWIC_RSN_LOCKED	"X'00000811'" Meaning: The IFAWIC caller was locked. Action: Call IFAWIC without holding locks.
40	(28)	BITSTRING	0	IFAWIC_RSN_HOMENOTPRIMARY	"X'00000812'" Meaning: The IFAWIC caller's primary home space was not equal to the caller's primary address space. Action: Call IFAWIC when home equals primary address space.
40	(28)	BITSTRING	0	IFAWIC_RSN_CALLERFRR	"X'00000813'" Meaning: The task mode IFAWIC REQUEST=REGISTER caller had FRR recovery established. Action: Call IFAWIC in task mode without FRRs established.
40	(28)	BITSTRING	0	IFAWIC_RSN_NOTAUTHORIZED	"X'00000814'" Meaning: The IFAWIC caller is not authorized. Action: Call IFAWIC only when authorized.
40	(28)	BITSTRING	0	IFAWIC_RSN_SUBTYPEINPUT	"X'00000820'" Meaning: The provided SUBTYPE is not supported. Action: Supply the IBM provided program subtype.
40	(28)	BITSTRING	0	IFAWIC_RSN_EXITVERSIONZERO	"X'00000821'" Meaning: The ExitVersion input must not be 0. Action: Specify an ExitVersion that is not 0.
40	(28)	BITSTRING	0	IFAWIC_RSN_BUFFER4KPAGESZERO	"X'00000822'" Meaning: The Buffer4kPages must not be 0. Action: Specify Buffer4kPages that is not 0.
40	(28)	BITSTRING	0	IFAWIC_RSN_BUFFER4KPAGESTOOLARGE	"X'00000823'" Meaning: The Buffer4kPages input must be less than or equal to 16. Action: Specify Buffer4kPages as less than or equal to 16.
40	(28)	BITSTRING	0	IFAWIC_RSN_BUFFERKEYMISMATCH	"X'00000824'" Meaning: A previous REQUEST=REGISTER requested and received a buffer for a BUFFERKEY value that was different that the BUFFERKEY specified on this request. Action: Update the program to request the same BUFFERKEY for each call to IFAWIC, even across different program versions.
40	(28)	BITSTRING	0	IFAWIC_RSN_SRBMODE	"X'00000825'" Meaning: IFAWIC was issued in SRB mode Action: Do not issue IFAWIC in SRB mode
40	(28)	BITSTRING	0	IFAWIC_RSN_SAMEVEREXITROUTINEMISMATCH	

Table 16. Structure WICPARM (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
					"X'00000830'" Meaning: A ExitRoutine and ExitVersion was specified such that the ExitVersion is the same as a previous specification, however, the ExitRoutine is different. Action: Specify the same ExitRoutine to correspond with the same ExitVersion. If the program introduces a new exit routine, the ExitVersion must be incremented.
40	(28)	BITSTRING	0	IFAWIC_RSN_EXITRTNNOTFOUND	"X'00000831'" Meaning: The exit routine could not be found. Action: Ensure the exit routine is located in either the invoking program's joblib, steplib, tasklib, in the system's LNKLIST concatenation or in LPA.
40	(28)	BITSTRING	0	IFAWIC_RSN_EXITRTNNOTINAPFLIB	"X'00000832'" Meaning: IFAWIC was unable to load the provided exit routine because did not reside in an APF authorized library. The "xxxx" portion of the reason code contains an informational code as documented with ABEND 306 reasons. No abend was generated. Action: Refer to "MVS System Codes", sections "System completion codes", "306", with the "xxxx" portion of the reason code to see what action is needed to resolve the condition. Ensure the exit routine resides in an APF authorized library and is re-entrant.
40	(28)	BITSTRING	0	IFAWIC_RSN_EXITRTNAMODE24	"X'00000833'" Meaning: The ExitRoutine must not be in AMODE 24. Action: Provide an exit routine with an AMODE of 31 or 64.
40	(28)	BITSTRING	0	IFAWIC_RSN_EXITRTNNOTREENTRANT	"X'00000834'" Meaning: The ExitRoutine must be re-entrant. Action: Provide an exit routine that is linked as re-entrant.
40	(28)	BITSTRING	0	IFAWIC_RSN_EXITRTNNAMENOTUNIQUE	"X'00000835'" Meaning: Another subtype is using the same exit routine name. The exit routine name must be unique per subtype. Action: Provide a unique exit routine name.
	.... 11..			IFAWIC_RC_ENVERROR	"X'0000000C'" Meaning: Environmental error The IFAWIC service did not complete successfully. For REQUEST=REGISTER, no WIC instrumentation buffer was provided to the caller. Action: Refer to the action provided with the specific reason code.
40	(28)	BITSTRING	0	IFAWIC_RSN_SYSTEMNOTREADY	"X'00000C01'" Meaning: IFAWIC was issued before the system is ready to start processing requests. Action: Wait for the system to be ready to accept IFAWIC requests and re-issue the IFAWIC request.
40	(28)	BITSTRING	0	IFAWIC_RSN_UNSUPPORTEDMACHINE	"X'00000C02'" Meaning: IFAWIC was issued from an unsupported machine. WIC requires an IBM z14 or later hardware. Action: Issue IFAWIC on a supported machine.

Table 16. Structure WICPARM (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
40	(28)	BITSTRING	0	IFAWIC_RSN_ENVNOEXIT	"X'00000C03'" Meaning: IFAWIC REQUEST=REGISTER requests omitting ExitVersion and ExitRoutine, require a WIC exit routine to have been already established by a previous IFAWIC REQUEST=REGISTER call specifying the ExitVersion and ExitRoutine parameters. Action: Specify ExitVersion and ExitRoutine or ensure an IFAWIC REQUEST=REGISTER call has already completed specifying ExitVersion and ExitRoutine before issuing IFAWIC REQUEST=REGISTER.
40	(28)	BITSTRING	0	IFAWIC_RSN_SMFNOWICSPECIFIED	"X'00000C04'" Meaning: SMF parameters specified NOWIC which prevents programs from issuing IFAWIC REQUEST=REGISTER. Action: Accept that WIC services will be unavailable for the program, or choose to handle WIC registration dynamically. If the program can support dynamic WIC registration, wait for ENF 85 signal and check the WicEnf bit WicEnf_RegisterIsAvailable. When WicEnf_RegisterIsAvailable is on, The ENF listener exit can cause a task to wake up to re-issue IFAWIC REQUEST=REGISTER for the program. The system issues ENF 85 signals to inform IFAWIC callers that IFAWIC Register is available when SMF parameter WIC is specified. See macro IFAWICCB for additional information.
40	(28)	BITSTRING	0	IFAWIC_RSN_EXITRTNNOSTORAGE	"X'00000C10'" Meaning: There was not sufficient storage for IFAWIC to process the exit routine. Action: Contact your system programmer. there is a shortage of common storage.
40	(28)	BITSTRING	0	IFAWIC_RSN_UNEXPECTEDLOADERROR	"X'00000C11'" Meaning: IFAWIC encountered errors trying to load the provided exit routine. The "xxxx" portion of the reason code contains a code from the load service of what would have been the ABEND code. No abend was generated. Action: Alert the system programmer. Refer to "MVS System Codes", sections "System completion codes", with the "xxxx" portion as the abend code to see what action is needed to resolve the condition. This may be accomonied with system log messages with the CSV prefix.
		...1 ....		IFAWIC_RC_COMPERROR	"X'00000010'" Meaning: Unexpected failure. Action: Refer to the action provided with the specific reason code.
40	(28)	BITSTRING	0	IFAWIC_RSN_COMPERROR	"X'00001001'" Meaning: Unexpected failure. The state of the request is unpredictable. Action: Contact your system programmer to report the problem to IBM service
40	(28)	X'30'	0	WICPARM_LEN	"*-WicParm"

Table 17. Structure WIC\_WARNING\_RSN

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	WIC_WARNING_RSN	
0	(0)	BITSTRING	2	WIC_WARNING_FLAGS	
0	(0)	BITSTRING	2	WIC_WARNING_FLAGSREGISTER	
0	(0)	BITSTRING	1	WIC_WARNING_WICNOTFULLYAVAIL	Flags indicating the WIC service is not fully available for processing. For IFAWIC REQUEST=REGISTER, when the return code is IFAWIC_Rc_Warning and bits 0-7 of the reason code contain any bit on, the WIC services are not fully available. The program should not write instrumentation data to its instrumentation buffer because the program's WIC exit routine is not likely to be called.
Bit definitions:					
		1... ....		WIC_WARNING_WICFEATURENOTENABLED	"X'80'" The WorkloadIntCorr product is not part of the product enablement policy, see IFAWIC_Rsn_WicFeatureNotEnabled
		.1.. ....		WIC_WARNING_SUBTYPENOTCOLLECTED	"X'40'" SMF parameters did not specify WIC, see IFAWIC_Rsn_SubtypeNotCollected
1	(1)	BITSTRING	1	WIC_WARNING_FLAGSREGISTERGENERAL	General flags for register requests
Bit definitions:					
		1... ....		WIC_WARNING_ALREADYREGISTERED	"X'80'" See IFAWIC_Rsn_AlreadyRegistered
		.1.. ....		WIC_WARNING_BUFFER4KPAGESDIFFERENT	"X'40'" A buffer 4k Pages larger or smaller than a previous register was detected, see IFA WIC_Rsn_Buffer4kPagesSmaller or IFAWIC_Rsn_Buffer4kPagesLarger
		..1. ....		WIC_WARNING_EXITVERSIONIGNORED	"X'20'" Exit version was less than current, see IFAWIC_Rsn_ExitVersionIgnored
0	(0)	BITSTRING	2	WIC_WARNING_FLAGSDEREGISTER	
0	(0)	BITSTRING	1	WIC_WARNING_FLAGSDEREGISTERBYTE1	
0	(0)	BITSTRING	1		Reserved for future use
1	(1)	BITSTRING	1	WIC_WARNING_FLAGSDEREGISTERGENERAL	
Bit definitions:					
		1... ....		WIC_WARNING_NOTREGISTERED	"X'80'" See IFAWIC_Rsn_NotRegistered
2	(2)	SIGNED	2	WIC_WARNING_CODE	
2	(2)	BITSTRING	0	WIC_NOTFULLYAVAIL_LOWHIGHMASK	

Table 17. Structure WIC\_WARNING\_RSN (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
					"X'FF00'" For IFAWIC REQUEST=REGISTER, when the return code is IFAWIC_Rc_Warning and bits 0-7 of the reason code contain any bit on, the WIC services are not fully available. The program should not write instrumentation data to its instrumentation buffer because the program's WIC exit routine is not likely to be called. Assembler programs can use 'TMLH 0,Wic_NotFullyAvail_LowH ighMask' to check for this condition when the return code is IFAWIC_Rc_Warning
2	(2)	X'4'	0	WIC_WARNING_RSN_LEN	"*-Wic_Warning_Rsn"

Table 18. Structure WICENFQUAL

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	WICENFQUAL	
0	(0)	BITSTRING	1	WICENFQUAL_CODE	Code for ENF85 request
1	(1)	CHARACTER	1		reserved for future use
2	(2)	SIGNED	2	WICENFQUAL_SUBTYPE	Subtype number
WIC ENF Qual Codes					
2	(2)	X'1'	0	WICENF_CODE_INSTRUMENTATIONCHANGE	"1" This ENF is issued when SMF requests a change in instrumentation status for the WIC exploiter. The listener should check the WicENF parameter list field WicEnf_InstrumentationRequested to verify if the program should start or stop instrumenting. This condition occurs when SMF noticed a system change that indicates that affects whether the WIC exit routine will or will not be called.
2	(2)	X'2'	0	WICENF_CODE_REGISTERAVAILABLE	"2" This ENF was issued when the SMF WIC parameter was specified and IFAWIC REQUEST=REGISTER may succeed. Check WicEnf parameter list field WicEnf_RegisterIsAvailable to verify the most current status. If possible, alert a task to issue IFAWIC REQUEST=REGISTER to begin using WIC services.
2	(2)	X'4'	0	WICENFQUAL_LEN	"*-WicEnfQual"

Table 19. Structure WICENF

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	WICENF	
0	(0)	CHARACTER	8	WICENF_EYE	'WICENF '
8	(8)	SIGNED	2	WICENF_VERSION	Version of WicEnf
10	(A)	SIGNED	2	WICENF_SUBTYPE	Subtype number
12	(C)	BITSTRING	1	WICENF_FLAGS	Status Flags
Bit definitions:					

Table 19. Structure WICENF (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
		1... ....		WICENF_INSTRUMENTATIONREQUESTED	"X'80'" When on, instrumentation is requested for this subtype. When off, instrumentation is not requested for this subtype.
		.1.. ....		WICENF_REGISTERISAVAILABLE	"X'40'" When on, SMF parameter WIC is specified and IFAWIC REQUEST=REGISTER is available to be re-tried. When off, SMF parameter NOWIC is specified and IFAWIC REQUEST=REGISTER is likely to fail with return code IFAWIC_Rc_EnvError reason code IFAWIC_Rsn_SMFNOWICSpecified
		..1. ....		WICENF_EXITADDEROR	"X'20'" When on, SMF encountered an error adding the exit for the subtype. This is likely because an exit of the same name was added to WIC processing for another subtype. Specify unique names for exit routines for each subtype. When this bit is on, WicEnf_InstrumentationRequeste d is off
13	(D)	CHARACTER	3		Reserved for future use
13	(D)	X'C9C3C5'	0	WICENF_EYECONST_0T03	"C'WICE'" This is the first 4-byte segment of an 8-byte constant. Eye catcher for WicEnf
13	(D)	X'C64040'	0	WICENF_EYECONST_4T07	"C'NF '" This is the second 4-byte segment of an 8-byte constant. Eye catcher for WicEnf
13	(D)	X'1'	0	WICENF_VERCONST	"1" Current version for WicEnf
13	(D)	X'1'	0	WICENF_VER1CONST	"1" Version 1 constant
13	(D)	X'10'	0	WICENF_LEN	"*-WicEnf"

Table 20. Structure WICNB\_PARMLIST

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	WICNB_PARMLIST	Parameter list for the GetNextBuffer Service
0	(0)	SIGNED	2	WICNB_VERSION	Version identification
2	(2)	SIGNED	2	WICNB_INOUTASID	Address space ID. Caller must initialize this to zero for the initial invocation of the GetNextBuffer service. When the GetNextBuffer service completes with a return code of WicSrv_Rc_Success this will contain the address space id for the WIC buffer returned in WicNB_outBuffer@. For subsequent invocations of the GetNextBuffer service the caller should not change the content of WicNB_inoutAsid in order to obtain the next WIC buffer. When there are no remaining buffers, the service completes with an appropriate return and reason code.
4	(4)	CHARACTER	20	WICNB_RESERVED1	Reserved
24	(18)	CHARACTER	24	WICNB_OUTAREA	Output area
24	(18)	CHARACTER	16	WICNB_ASINFO	Address information
24	(18)	CHARACTER	8	WICNB_OUTJOBNAME	Jobname for Address Space Id (WicNB_inoutAsid)
32	(20)	CHARACTER	8	WICNB_OUTJOBPRIORITYAREA	Job priority area

Table 20. Structure WICNB\_PARMLIST (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
32	(20)	SIGNED	2	WICNB_OUTJOBPRIORITYBUCKET	Priority bucket - describes relative priority of the address space. When: 1 - SVT_kHiPriorityBucketIndex 2 - SVT_kMedPriorityBucketIndex 3 - SVT_kLowPriorityBucketIndex 4 - SVT_kDiscPriorityBucketIndex See IHASVT for description of the above constants
34	(22)	SIGNED	2	WICNB_OUTJOBSIZEBUCKET	Sub buckets for each of the priority buckets for the address space.
40	(28)	ADDRESS	8	WICNB_OUTBUFFERADDR	Starting address of the WIC buffer for this ASID (WicNB_inoutASID)
Constants for GetNextBuffer Service					
40	(28)	X'1'	0	WICNB_KVER	"1" Current version
40	(28)	X'1'	0	WICNB_KMAXVER	"1"
40	(28)	X'1'	0	WICNB_KVER1	"1"
40	(28)	X'1'	0	WICNB_#VER	"1" Current version
40	(28)	X'1'	0	WICNB_#MAXVER	"1"
40	(28)	X'1'	0	WICNB_#VER1	"1"
40	(28)	X'0'	0	WIC_KPROCCLASSINDEX_CP	"0" Local constant for CP proc class index to avoid having to include IHAPSA
40	(28)	X'2'	0	WIC_KPROCCLASSINDEX_ZIIP	"2" Local constant for zIIP proc class index to avoid having to include IHAPSA
40	(28)	X'30'	0	WICNB_PARMLIST_LEN	"*-WicNB_ParmList"

Table 21. Structure WICCB\_PARMLIST

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	WICCB_PARMLIST	Input parameter list for the Clear Buffer Service
0	(0)	SIGNED	2	WICCB_VERSION	Version identification
2	(2)	SIGNED	2	WICCB_INASID	Address space id
4	(4)	CHARACTER	4	WICCB_RESERVED1	Reserved for alignment
8	(8)	ADDRESS	8	WICCB_INSTARTADDRBUFTOCLEAR	Address within the WIC buffer to clear for the subtype and the given address space id (WicCB_inAsid). The WIC buffer address is obtained by calling the get next buffer service
16	(10)	SIGNED	4	WICCB_INBUFLENTOCLEAR	Length of the WIC buffer to be cleared from the starting WIC buffer address as specified by WicCB_inStartAddrBufToClear
Constants for ClearBuffer Service					
16	(10)	X'1'	0	WICCB_KVER	"1" Current version
16	(10)	X'1'	0	WICCB_KMAXVER	"1"
16	(10)	X'1'	0	WICCB_KVER1	"1"
16	(10)	X'1'	0	WICCB_#VER	"1" Current version
16	(10)	X'1'	0	WICCB_#MAXVER	"1"
16	(10)	X'1'	0	WICCB_#VER1	"1"

Table 21. Structure WICCB\_PARMLIST (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
16	(10)	X'14'	0	WICCB_PARMLIST_LEN	"*-WicCB_ParmList"

Table 22. Structure WICWR\_PARMLIST

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
0	(0)	STRUCTURE	0	WICWR_PARMLIST	Input parameter list for the Write SMF 98.x build and write service
0	(0)	SIGNED	2	WICWR_VERSION	Parameter list version
2	(2)	CHARACTER	4	WICWR_RSVD02	Reserved for alignment
6	(6)	BITSTRING	1	WICWR_FLAG	Build and write SMF 98.x Record flags

Bit definitions:

1... ....

WICWR\_FLAGEXCJOBSUNIQUEBYJOBSIZE

"X'80'" This controls whether multiple instances of the same job are compressed into a single instance always (off, default) or per job size (on). When off, the same job in any exception job bucket entry (WicWR\_ExcBucket\_Data) is compressed into a single exception job list entry because each job instance contains the same data (WicWR\_ExcBucket\_Data). When on, exception job bucket entries (WicWR\_ExcBucket) for the same job but with a different job size is unique, so it is added as an exception job list entry. When on, the same job in the same job size is compressed to a single entry because it contains the same data (WicWR\_ExcBucket\_Data). This bit may be set on when using either the default job sizes (when WicWR\_CustomJobSizeEntries=0) or custom job sizes (when WicWR\_CustomJobSizeEntries!=0). Setting this on requires no zIIP aggregate buckets (WicWR\_AggBucket) and no zIIP exception buckets (WicWR\_ExcBucket).

.111 1111

WICWR\_FLAGRSVD

"X'7F'" Reserved flags, must be 0

7 (7) BITSTRING

1 WICWR\_CUSTOMJOBSIZEENTRIES

This controls the job size entries and attributes. When zero, the build and write SMF 98.x record service applies the default job size bucket entries (WicWR\_kJobSizeDefNum) and bucket attributes (WicWR\_kJobSize\*). When non-zero, the required default job size bucket entries (WicWR\_kJobSizeDefNumReq) and attributes (WicWR\_kJobSizeAll) are applied, and the remaining job size bucket entries WicWR\_kJobSizeLarge (1) - WicWR\_kJobSizeTiny (4) are replaced with exploiter defined bucket entries from 1 to WicWR\_CustomJobSizeEntries with exploiter defined attributes. Job size bucket entry WicWR\_kJobSizeAll (0) is not replacable, so if it is not desired, set its bucket entry contents to binary zeros to prevent it from being written to the SMF 98 record. WicWR\_kCustomJobSizeEntriesMax is the maximum value that can be specified for WicWR\_CustomJobSizeEntries.



Table 22. Structure WICWR\_PARMLIST (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
<p>SMF 98 Record subtype information parameter list All SMF 98 subtype records (SMF 98.x) MUST be compatibly changed. The WicWR_ReleaseIndex (RI), WicWR_WithinReleaseIndex (WRI), and WicWR_PrototypeIndex (PI) form a macro version that is of the form RI.WRI.PI. Each SMF 98 subtype (SMF 98.x) has its own independent macro version. For products with multiple levels installed, the macro version ensures the latest mapping macro is used to analyze the SMF 98 subtype records. IBM expects all exploiters to follow these rules for setting RI, WRI, and PI: 1. For any shipped support, PI must be zero. 2. For the first delivery of an SMF 98 subtype, the indices MUST be set to RI=1, WRI=0, PI=0 across all levels. 3. When adding new data to an existing SMF 98 subtype record: a. For a new release or version RI is updated to RI+1. WRVI and PI should both be set to zero. b. For an existing release and version, RI is unchanged, WRVI is updated to WRI+1, and PI is set to 0. 4. For prototype testing (e.g. ++APAR, developer supplied link edit) that may or may not be delivered, RI and WRVI are to remain unchanged. PI is to be set to a non-zero value (e.g. PI=1). This mapping macro can be used to analyze SMF98.x records with matching RI.WRI.PI and earlier compatible records where PI=0 with the same or lower RI and/or WRI settings. Different PI values for the same RI and WRVI values are treated as incompatible test changes. Note: The WicWR_SubtypeInfo is a one to one mapping of SMF98_SubtypeInfo in IHAHR098. Changes to this</p>					
<p>WicWR_SubtypeInfo will require the same changes to the SMF98_SubtypeInfo mapping.</p>					
8	(8)	CHARACTER	24	WICWR_SUBTYPEINFO	SMF 98.x subtype information parameter list
8	(8)	SIGNED	2	WICWR_RELEASEINDEX	The current release version index value
10	(A)	SIGNED	2	WICWR_WITHINRELEASEINDEX	The current within release index value
12	(C)	SIGNED	2	WICWR_PROTOTYPEINDEX	The current prototype index value - IBM use only
14	(E)	CHARACTER	2	WICWR_RSVD0D	Reserved - IBM use only
16	(10)	CHARACTER	16	WICWR_PRODLEVEL	The program product level. - Bytes 0-7 contain the product name - Bytes 8-15 contain the FMID

Table 22. Structure WICWR\_PARMLIST (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
<p>Aggregate buckets parameter list (mapped by WicWR_AggregateBucketParmList) Up to two aggregate buckets are supported. IFAWIC exit exploiters may want to utilize two aggregate buckets, for example, one aggregate bucket can be used for global or non-CP related information, whereas, a second aggregate bucket can be used for CP related information. Notes: 1. For additional information on the structure of the aggregate bucket, see the definition of WicWR_AggBucket in this mapping macro. For an understanding of the output of the aggregate bucket in an SMF98.x record see the definition SMF98AggBucket in IHAHR098. 2. The number of aggregate bucket entries (instances) must be the product of the following:</p> <ul style="list-style-type: none"> <li>o the number of CPU types (1 for CP only, or WicWR_kCPUTypeDefNum (2) for CPs+zIIPs) times</li> <li>o the number of job priority buckets (WicWR_kJobPriorityDefNum (5)) times</li> <li>o one of the following: <ul style="list-style-type: none"> <li>- When WicWR_CustomJobSizeEntries is zero, the default number of job sizes (WicWR_kJobSizeDefNum (5)). In this case, there must be WicWR_kNumEntsOneCPUType (1 5 5=25) or WicWR_kNumEntsTwoCPUType (2 5 5=50) aggregate bucket entries.</li> <li>- When WicWR_CustomJobSizeEntries is non-zero, the exploiter replaces the job size bucket entries WicWR_kJobSizeLarge (1) to WicWR_kJobSizeTiny (4) with exploiter defined bucket entries from 1 to WicWR_CustomJobSizeEntries. In this case, there must be (1 5 (1+WicWR_CustomJobSizeEntries)) or (2 5 (1+WicWR_CustomJobSizeEntries)) aggregate bucket entries, where the 1+ is equivalent to WicWR_kJobSizeDefNumReq, which represents Job size bucket entry</li> </ul> </li> </ul> <p>WicWR_kJobSizeAll (0), which cannot be replaced.</p> <p>3. The aggregate bucket length defines the length of the exploiter's aggregate bucket for one entry (i.e. the length includes the length of the contributors, the record area and internal area). See WicWR_AggBucket definition. 4. The aggregate bucket write length defines the length of the aggregate bucket entry that will be written to the SMF98.x record. It includes the length of the contributors plus the length of the record area. See WicWR_AggBucket definition. 5. The write length of one aggregate bucket cannot be greater than the length of one aggregate bucket entry (instance).</p>					
32	(20)	CHARACTER	24	WICWR_AGGBUCKETPARMLIST1	First aggregate bucket parameter list
56	(38)	CHARACTER	24	WICWR_AGGBUCKETPARMLIST2	Second aggregate bucket parameter list

Table 22. Structure WICWR\_PARMLIST (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
<p>Exception bucket parameter list Only one exception bucket is supported. Note: 1. For additional information on the structure of the exception bucket, see the definition of WicWR_ExcBucket in this mapping macro. For an understanding of the output of the exception bucket in an SMF98.x record see the definition SMF98JobIdx and SMF98JobList in IHAHR098.</p> <p>2. The number of exception bucket entries (WicWR_ExcBucketNumOfEntries) must be the product of the following:</p> <ul style="list-style-type: none"> <li>o the number of exception activities (WicWR_ExcBucketNumOfActivities) times</li> <li>o the number of CPU types (1 for CP only, or WicWR_kCPUTypeDefNum (2) for CPs+zIIPs) times</li> <li>o the number of job priority buckets (WicWR_kJobPriorityDefNum (5)) times</li> <li>o one of the following: <ul style="list-style-type: none"> <li>- When WicWR_CustomJobSizeEntries is zero, the default number of job sizes (WicWR_kJobSizeDefNum (5)).</li> </ul> </li> </ul> <p>In this case, there must be WicWR_ExcBucketNumOfActivities CPU Types (1 or 2) Priority Buckets (5) Job Size Buckets (5) exception bucket entries.</p> <ul style="list-style-type: none"> <li>- When WicWR_CustomJobSizeEntries is non-zero, the exploiter replaces the job size bucket entries WicWR_kJobSizeLarge (1) to WicWR_kJobSizeTiny (4) with exploiter defined bucket entries from 1 to WicWR_CustomJobSizeEntries.</li> </ul> <p>In this case, there must be WicWR_ExcBucketNumOfActivities CPU Types (1 or 2) Priority Buckets (5) (1+WicWR_CustomJobSizeEntries) exception bucket entries, where the 1+ is equivalent to WicWR_kJobSizeDefNumReq, which represents Job size bucket entry WicWR_kJobSizeAll (0), which cannot be replaced.</p> <p>3. The exception bucket entry length defines the length of</p>					
<p>the exploiter's exception bucket for one entry (i.e. this includes the length of the ASID, Job Name, record area and internal area). See WicWR_ExpBucket definition. 4. The write length of one exception bucket cannot be greater than the length of one exception bucket entry (instance).</p>					
80	(50)	CHARACTER	24	WICWR_EXCEPTIONBUCKETPARMLIST	Exception bucket parameter list
80	(50)	SIGNED	4	WICWR_EXCCKETNUMOFENTRIES	The number of entries in the exception bucket. For value requirements of this field, see the previous block comment
84	(54)	SIGNED	4	WICWR_EXCCKETNUMOFACTIVITIES	The number of activities for the exception bucket
88	(58)	SIGNED	4	WICWR_EXCCKETENTRYLENGTH	The total length of one entry (instance) in the exception bucket exploiter area
92	(5C)	SIGNED	4	WICWR_EXCCKETENTRYWRITELENGTH	The write length one entry in the exception bucket exploiter area
96	(60)	ADDRESS	8	WICWR_EXCCKETSTARTADDR	Starting address of the exception bucket - see WicWR_ExcBucket
Constants for Write SMF 98.x Record Service					
96	(60)	X'1'	0	WICWR_KVER	"1" Current version

Table 22. Structure WICWR\_PARMLIST (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
96	(60)	X'1'	0	WICWR_KMAXVER	"1"
96	(60)	X'1'	0	WICWR_KVER1	"1"
96	(60)	X'1'	0	WICWR_#VER	"1" Current version
96	(60)	X'1'	0	WICWR_#MAXVER	"1"
96	(60)	X'1'	0	WICWR_#VER1	"1"
96	(60)	X'40'	0	WICWR_KCUSTOMJOBSIZEENTRIESMAX	"64"
96	(60)	X'68'	0	WICWR_PARMLIST_LEN	"*-WicWR_ParmList"

Table 23. Structure WICWR\_AGGBUCKETPARMLIST

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	WICWR_AGGBUCKETPARMLIST	Aggregate bucket parameter lists
0	(0)	SIGNED	4	WICWR_AGGBUCKETNUMOFENTRIES	The number of entries for this aggregate bucket
4	(4)	SIGNED	4	WICWR_AGGBUCKETENTRYLENGTH	The total length of one entry (instance) in the aggregate bucket exploiter area
8	(8)	SIGNED	4	WICWR_AGGBUCKETENTRYWRITELENGTH	The write length of one entry in the aggregate bucket exploiter area
12	(C)	CHARACTER	4	WICWR_RSVD2B	Reserved for alignment
16	(10)	ADDRESS	8	WICWR_AGGBUCKETSTARTADDR	Starting address of the aggregate bucket - see WicWR_AggBucket
16	(10)	X'18'	0	WICWR_AGGBUCKETPARMLIST_LEN	"*-WicWR_AggBucketParmList"

Table 24. Structure WICWR\_AGGBUCKET

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	0	WICWR_AGGBUCKET	Aggregate bucket
Aggregate bucket exploiter area					
0	(0)	CHARACTER	2	WICWR_AGGBUCKETEXPAREA	Aggregate bucket exploiter area
Contributors - the number of contributors for this bucket instance					
0	(0)	SIGNED	2	WICWR_AGGBUCKET_CONTRIBUTORS	Number of address spaces contributing to this bucket entry - supplied by the exploiter
Aggregate bucket data area					
2	(2)	CHARACTER	1	WICWR_AGGBUCKET_DATA(0)	Aggregate bucket data area End of Aggregate Bucket definition
2	(2)	X'2'	0	WICWR_AGGBUCKET_LEN	"*-WicWR_AggBucket"

Table 25. Structure WICWR\_EXCbucket

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
0	(0)	STRUCTURE	0	WICWR_EXCbucket	Exception bucket
Exception bucket exploiter area					
0	(0)	CHARACTER	10	WICWR_EXCbucketEXPAREA	Exception bucket exploiter area
ASID - the address space identifier for this bucket instance					
0	(0)	SIGNED	2	WICWR_EXCbucket_ASID	Address space id - supplied by the exploiter
Jobname - the jobname for this bucket instance					
2	(2)	CHARACTER	8	WICWR_EXCbucket_JOBNAME	Jobnamespace id - supplied by the exploiter
Exception bucket data area					
10	(A)	CHARACTER	1	WICWR_EXCbucket_DATA(0)	Exception bucket data area End of exception bucket definition
Aggregate bucket sizes for aggregate buckets that contain one processor type or two processor types.					
10	(A)	X'19'	0	WICWR_KNUMENTSONECPUTYPE	"25"
10	(A)	X'32'	0	WICWR_KNUMENTSTWOCPUTYPE	"50"
CPU Type constants					
10	(A)	X'0'	0	WICWR_KCPUTYPECP	"0"
10	(A)	X'1'	0	WICWR_KCPUTYPEZIIP	"1"
10	(A)	X'0'	0	WICWR_KCPUTYPEDEFFIRST	"0"
10	(A)	X'1'	0	WICWR_KCPUTYPEDEFLAST	"1"
10	(A)	X'2'	0	WICWR_KCPUTYPEDEFNUM	"2"
10	(A)	X'1'	0	WICWR_KCPUTYPEDEFNUMCPONLY	"1"
JobPriority constants					
10	(A)	X'0'	0	WICWR_KJOBPRIORITYALL	"0"
10	(A)	X'1'	0	WICWR_KJOBPRIORITYCRITICAL	"1"
10	(A)	X'2'	0	WICWR_KJOBPRIORITYHIGH	"2"
10	(A)	X'3'	0	WICWR_KJOBPRIORITYLOW	"3"
10	(A)	X'4'	0	WICWR_KJOBPRIORITYDISC	"4"
10	(A)	X'0'	0	WICWR_KJOBPRIORITYDEFFIRST	"0"
10	(A)	X'4'	0	WICWR_KJOBPRIORITYDEFLAST	"4"
10	(A)	X'5'	0	WICWR_KJOBPRIORITYDEFNUM	"5"
JobSize constants					
10	(A)	X'0'	0	WICWR_KJOBSIZEALL	"0"
10	(A)	X'1'	0	WICWR_KJOBSIZELARGE	"1"
10	(A)	X'2'	0	WICWR_KJOBSIZEMEDIUM	"2"
10	(A)	X'3'	0	WICWR_KJOBSIZESMALL	"3"
10	(A)	X'4'	0	WICWR_KJOBSIZETINY	"4"
10	(A)	X'0'	0	WICWR_KJOBSIZEDEFFIRST	"0"

Table 25. Structure WICWR\_EXCbucket (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
10	(A)	X'4'	0	WICWR_KJOBSIZEDEFLAST	"4"
10	(A)	X'5'	0	WICWR_KJOBSIZEDEFNUM	"5"
10	(A)	X'1'	0	WICWR_KJOBSIZEDEFNUMREQ	"1" Job size default number of required bucket entries. WicWR_kJobSizeAll is required.
Wic Exit Services Return and reason code constants Successful return and reason codes					
	....			WICSRV_RC_SUCCESS	"X'00000000" Meaning: The get next buffer service has returned the first or next WIC buffer or the clear buffer service has cleared the WIC buffer starting at the given offset for the given length, or the build and write service SMF98.x has successfully built and written the SMF98.x record. Action: None
Warning return and reason codes					
	....	.1..		WICSRV_RC_WARNING	"X'00000004" Meaning: The get next buffer service, the clear buffer, or the build and write SMF98.x service completed with a warning Action: See the issued reason code
10	(A)	BITSTRING	0	WICSRV_RSN_NOMOREBUFFERS	"X'00000401" Meaning: The get next buffer service was unable to find the first or "next" WIC buffer. Action: Callers of the WIC service should terminate current processing and move on to the next phase of exit processing
10	(A)	BITSTRING	0	WICSRV_RSN_CLEARSTADDRBUFLenOUTOFRANGE	"X'00000402" Meaning: The caller of the clear buffer service specified a buffer length to clear that when combined with the starting address within the WIC buffer extended beyond the buffer size requested when the IFAWIC REGISTER request was issued for the subtype. The WIC buffer area starting at the address to clear is clear (set to binary 0s). Action: To avoid this warning re-register the subtype specifying a larger buffer size.
10	(A)	BITSTRING	0	WICSRV_RSN_APPROACHINGMAXSMF98RECLen	"X'0000040A" Meaning: The build and write SMF98.x service has determined that the SMF98.x record is approaching the 32756 byte record length limit. The SMF98.x record is written. Action: Verify that the number of records for the aggregate buckets and the exception bucket and the write lengths for the buckets are correct. If the the number of records and write lengths are correct, consider reducing the amount of data recorded for each aggregate bucket record and/or exception bucket record.
User error return and reason codes					
	....	1...		WICSRV_RC_USERERROR	"X'00000008" Meaning: The called service completed with a user error Action: See the issued reason code

Table 25. Structure WICWR\_EXCbucket (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
10	(A)	BITSTRING	0	WICSRV_RSN_BADPARMLIST	"X'00000801" Meaning: An invalid parameter list was built and supplied to the called service. Action: Verify the parameter list for the called service is built and initialized properly
10	(A)	BITSTRING	0	WICSRV_RSN_BADCLEARSTADDRBUFLN	"X'00000802" Meaning: The caller of the clear buffer service specified a starting WIC buffer address and buffer length to clear that extends beyond the maximum size of a WIC buffer or is not within the range for the subtype. within The buffer area is not cleared. Action: Verify that either the starting buffer address is correct and/or the buffer length to clear is correct.
10	(A)	BITSTRING	0	WICSRV_RSN_AGGBUCKETSNOTSPECIFIED	"X'00000803" Meaning: The caller of the build and write SMF98.x service specified a zero address for both the aggregate bucket addresses within the aggregate bucket portion of the parameter list. Action: Verify that at least one of the aggregate bucket addresses specified and is not zero
10	(A)	BITSTRING	0	WICSRV_RSN_BADNUMOFAGGBUCKETENTRIES	"X'00000804" Meaning: The caller of the build and write SMF98.x service specified a bad number of aggregate bucket entries. Action: Specify a correct number of aggregate buckets based on the comments before WicWR_AggBucket
10	(A)	BITSTRING	0	WICSRV_RSN_AGGBUCKETWRLNGTONEBUCLN	"X'00000805" Meaning: The caller of the build and write SMF98.x service specified a write length for one aggregate bucket entry that is greater than one aggregate bucket entry length. Action: Verify the write length of one aggregate bucket entry is defined to be less than the length of one aggregate bucket entry.
10	(A)	BITSTRING	0	WICSRV_RSN_EXCbucketNOTSPECIFIED	"X'00000807" Meaning: The caller of the build and write SMF98.x service specified a zero address for the exception bucket address, or specified a zero for the number of entries contained in the exception bucket, or specified a zero for the number of activities, within the exception bucket portion of the parameter list. Action: Verify that the exception bucket address is specified and is not zero, and/or the number of entries is not zero, and/or the number of activities is not zero.
10	(A)	BITSTRING	0	WICSRV_RSN_BADNUMOFEXCbucketENTRIES	"X'00000808" Meaning: The caller of the build and write SMF98.x service specified a bad number exception bucket entries. Action: Specify a correct number of exception bucket entries based on the comments before WicWR_ExcBucket

Table 25. Structure WICWR\_EXCbucket (continued)

Offset Dec	Offset Hex	Type	Len	Name (Dim)	Description
10	(A)	BITSTRING	0	WICSRV_RSN_EXCbucketWRLengToneBUCLen	"X'00000809" Meaning: The caller of the build and write SMF98.x service specified a write length for one exception bucket entry that is greater than one exception bucket entry length. Action: Verify the write length of one exception bucket entry is defined to be less than the length of one exception bucket entry.
10	(A)	BITSTRING	0	WICSRV_RSN_SMF98RECORDTOOLONG	"X'0000080A" Meaning: The build and write SMF98.x service has determined that the SMF98.x record exceeds 32756 bytes in length. The SMF98.x record is not written. Action: Verify that the number of records for the aggregate buckets and the exception bucket and the write lengths for the buckets are correct. If the the number of records and write lengths are correct, consider reducing the amount of data recorded for each aggregate bucket record and/or exception bucket record.
10	(A)	BITSTRING	0	WICSRV_RSN_BADCUSTOMJOBSIZEENTRIES	"X'0000080B" Meaning: The build and write SMF98.x service has determined WicWR_CustomJobSizeEntries is greater than WicWR_kCustomJobSizeEntriesMax. The SMF98.x record is not written. Action: Set WicWR_CustomJobSizeEntries to a value less than or equal to WicWR_kCustomJobSizeEntries Max.
10	(A)	BITSTRING	0	WICSRV_RSN_ZIIPAGGBUCEXCUNIQUEBYJOBSIZE	"X'0000080C" Meaning: The build and write SMF98.x service has determined zIIP aggregate bucket(s) (WicWR_AggBucket) were provided when an exception job's activities with different job sizes are unique (WicWR_FlagExcJobsUniqueByJobSize is on). The SMF98.x record is not written. Action: Remove zIIP aggregate bucket(s) from WicWR_AggBucket or indicate exception job activities are the same for all job sizes by specifying WicWR_FlagExcJobsUniqueByJobSize off.
10	(A)	BITSTRING	0	WICSRV_RSN_ZIIPEXCCEXCUNIQUEBYJOBSIZE	"X'0000080D" Meaning: The build and write SMF98.x service has determined that zIIP exception job bucket (WicWR_ExcBucket) was provided when an exception job's activities with different job sizes are unique (WicWR_FlagExcJobsUniqueByJobSize is on). The SMF98.x record is not written. Action: Remove zIIP exception bucket(s) from WicWR_ExcBucket or indicate exception job activities are the same for all job sizes by specifying WicWR_FlagExcJobsUniqueByJobSize off.

Environment error return and reason codes



Table 25. Structure WICWR\_EXCbucket (continued)

Offset Dec	Offset Hex	Type	Len	Name(Dim)	Description
		.... 11..		WICSRV_RC_ENVERROR	"X'0000000C'" Meaning: The WIC service did not complete successfully due to an environmental error. Action: See the issued reason code
10	(A)	BITSTRING	0	WICSRV_RSN_UNABLETOWRITESMF98RECORD	"X'00000C01'" Meaning: The SMF98.x build and write service was unable to write the SMF98.x record. Action: This error is issued when the return code from the SMFEWTF macro invocation is non-zero. Refer to the high half (bits 0-15) of this reason code for the return code issued by the SMFEWTF macro. Documentation and corrective actions for the SMFEWTF return codes can be found in the z/OS MVS System Management Facilities (SMF) publication
10	(A)	X'A'	0	WICWR_EXCbucket_LEN	"*-WicWR_ExcBucket"

Table 26. Cross Reference for IFAWICCB

Name	Offset	Hex Tag
IFAWIC_RC_COMPERROR	28	10
IFAWIC_RC_ENVERROR	28	C
IFAWIC_RC_SUCCESS	28	0
IFAWIC_RC_USERERROR	28	8
IFAWIC_RC_WARNING	28	4
IFAWIC_RSN_ALREADYREGISTERED	28	403
IFAWIC_RSN_BADBUFFERKEY	28	805
IFAWIC_RSN_BADREQUESTTYPE	28	804
IFAWIC_RSN_BADVERSION	28	802
IFAWIC_RSN_BUFFERKEYMISMATCH	28	824
IFAWIC_RSN_BUFFER4KPAGESLARGER	28	402
IFAWIC_RSN_BUFFER4KPAGESSMALLER	28	401
IFAWIC_RSN_BUFFER4KPAGESTOOLARGE	28	823
IFAWIC_RSN_BUFFER4KPAGESZERO	28	822
IFAWIC_RSN_CALLERFRR	28	813
IFAWIC_RSN_COMPERROR	28	1001
IFAWIC_RSN_ENVNOEXIT	28	C03
IFAWIC_RSN_EXITRTNMODE24	28	833
IFAWIC_RSN_EXITRTNNAMENOTUNIQUE	28	835
IFAWIC_RSN_EXITRTNNOSTORAGE	28	C10
IFAWIC_RSN_EXITRTNNOTFOUND	28	831
IFAWIC_RSN_EXITRTNNOTINAPFLIB	28	832
IFAWIC_RSN_EXITRTNNOTREENTRANT	28	834
IFAWIC_RSN_EXITVERSIONIGNORED	28	405
IFAWIC_RSN_EXITVERSIONZERO	28	821

Table 26. Cross Reference for IFAWICCB (continued)

Name	Offset	Hex Tag
IFAWIC_RSN_HOMENOTPRIMARY	28	812
IFAWIC_RSN_LOCKED	28	811
IFAWIC_RSN_NOTAUTHORIZED	28	814
IFAWIC_RSN_NOTENABLED	28	810
IFAWIC_RSN_NOTREGISTERED	28	404
IFAWIC_RSN_PARMLISTALET	28	801
IFAWIC_RSN_PARMLISTFETCH	28	807
IFAWIC_RSN_PARMLISTWRITE	28	808
IFAWIC_RSN_RESERVEDNOT0	28	803
IFAWIC_RSN_SAMEVEREXITROUTINEMISMATCH	28	830
IFAWIC_RSN_SMFNOWICSPECIFIED	28	C04
IFAWIC_RSN_SRBMODE	28	825
IFAWIC_RSN_SUBTYPEINPUT	28	820
IFAWIC_RSN_SUBTYPENOTCOLLECTED	28	482
IFAWIC_RSN_SYSTEMNOTREADY	28	C01
IFAWIC_RSN_UNEXPECTEDLOADERROR	28	C11
IFAWIC_RSN_UNSUPPORTEDMACHINE	28	C02
IFAWIC_RSN_WICFEATURENOTENABLED	28	481
WIC_KDYNAMICAREALENGTH	28	2000
WIC_KFIRSTTESTSUBTYPE	28	2000
WIC_KLASTTESTSUBTYPE	28	2007
WIC_KMAX4KPAGES	28	10
WIC_KPROCCLASSINDEX_CP	28	0
WIC_KPROCCLASSINDEX_ZIIP	28	2
WIC_NOTFULLYAVAIL_LOWHIGHMASK	2	FF00
WIC_RSN_CODEMASK	28	FFFF
WIC_WARNING_ALREADYREGISTERED	1	80
WIC_WARNING_BUFFER4KPAGESDIFFERENT	1	40
WIC_WARNING_CODE	2	
WIC_WARNING_EXITVERSIONIGNORED	1	20
WIC_WARNING_FLAGS	0	
WIC_WARNING_FLAGSDEREGISTER	0	
WIC_WARNING_FLAGSDEREGISTERBYTE1	0	
WIC_WARNING_FLAGSDEREGISTERGENERAL	1	
WIC_WARNING_FLAGSREGISTER	0	
WIC_WARNING_FLAGSREGISTERGENERAL	1	
WIC_WARNING_NOTREGISTERED	1	80
WIC_WARNING_RSN	0	
WIC_WARNING_RSN_LEN	2	4

Table 26. Cross Reference for IFAWICCB (continued)

Name	Offset	Hex Tag
WIC_WARNING_SUBTYPENOTCOLLECTED	0	40
WIC_WARNING_WICFEATURENOTENABLED	0	80
WIC_WARNING_WICNOTFULLYAVAIL	0	
WICCB_#MAXVER	10	1
WICCB_#VER	10	1
WICCB_#VER1	10	1
WICCB_INASID	2	
WICCB_INBUFLENTOCLEAR	10	
WICCB_INSTARTADDRBUFTOCLEAR	8	
WICCB_KMAXVER	10	1
WICCB_KVER	10	1
WICCB_KVER1	10	1
WICCB_PARMLIST	0	
WICCB_PARMLIST_LEN	10	14
WICCB_RESERVED1	4	
WICCB_VERSION	0	
WICENF	0	
WICENF_CODE_INSTRUMENTATIONCHANGE	2	1
WICENF_CODE_REGISTERAVAILABLE	2	2
WICENF_EXITADDERROR	C	20
WICENF_EYE	0	
WICENF_EYECONST_0T03	D	C9C3C5
WICENF_EYECONST_4T07	D	C64040
WICENF_FLAGS	C	
WICENF_INSTRUMENTATIONREQUESTED	C	80
WICENF_LEN	D	10
WICENF_REGISTERISAVAILABLE	C	40
WICENF_SUBTYPE	A	
WICENF_VERCONST	D	1
WICENF_VERSION	8	
WICENF_VER1CONST	D	1
WICENFQUAL	0	
WICENFQUAL_CODE	0	
WICENFQUAL_LEN	2	4
WICENFQUAL_SUBTYPE	2	
WICNB_#MAXVER	28	1
WICNB_#VER	28	1
WICNB_#VER1	28	1
WICNB_ASINFO	18	

Table 26. Cross Reference for IFAWICCB (continued)

Name	Offset	Hex Tag
WICNB_INOUTASID	2	
WICNB_KMAXVER	28	1
WICNB_KVER	28	1
WICNB_KVER1	28	1
WICNB_OUTAREA	18	
WICNB_OUTBUFFERADDR	28	
WICNB_OUTJOBNAME	18	
WICNB_OUTJOBPRIORITYAREA	20	
WICNB_OUTJOBPRIORITYBUCKET	20	
WICNB_OUTJOBSIZEBUCKET	22	
WICNB_PARMLIST	0	
WICNB_PARMLIST_LEN	28	30
WICNB_RESERVED1	4	
WICNB_VERSION	0	
WICPARM	0	
WICPARM_DYNAREALEN	10	
WICPARM_DYNAREAPTR	8	
WICPARM_KVERSIONCURRENT	28	1
WICPARM_KVERSIONFIRST	28	1
WICPARM_LEN	28	30
WICPARM_LENGTH	2	
WICPARM_REQPRIME	28	1
WICPARM_REQUEST	6	
WICPARM_REQWRITERECORD	28	2
WICPARM_SUBTYPE	4	
WICPARM_VERSION	0	
WICPARM_WICSRV_CLEARBUFFERPTR	20	
WICPARM_WICSRV_GETNEXTBUFFERPTR	18	
WICPARM_WICSRV_WRITESMF98PTR	28	
WICSRV_RC_ENVERROR	A	C
WICSRV_RC_SUCCESS	A	0
WICSRV_RC_USERERROR	A	8
WICSRV_RC_WARNING	A	4
WICSRV_RSN_AGGBUCKETSNOTSPECIFIED	A	803
WICSRV_RSN_AGGBUCKETWRLONGTONEBUCLN	A	805
WICSRV_RSN_APPROACHINGMAXSMF98RECLN	A	40A
WICSRV_RSN_BADCLEARSTADDRBUFLN	A	802
WICSRV_RSN_BADCUSTOMJOBSIZEENTRIES	A	80B
WICSRV_RSN_BADNUMOFAGGBUCKETENTRIES	A	804

Table 26. Cross Reference for IFAWICCB (continued)

Name	Offset	Hex Tag
WICSRV_RSN_BADNUMOFEXCBUCKETENTRIES	A	808
WICSRV_RSN_BADPARMLIST	A	801
WICSRV_RSN_CLEARSTADDRBUFLenOUTOFRANGE	A	402
WICSRV_RSN_EXCBUCKETNOTSPECIFIED	A	807
WICSRV_RSN_EXCBUCKETWRLengTONEBUCLen	A	809
WICSRV_RSN_NOMOREBUFFERS	A	401
WICSRV_RSN_SMF98RECORDTOOLONG	A	80A
WICSRV_RSN_UNABLETOWRITESMF98RECORD	A	C01
WICSRV_RSN_ZIIPAGGBUCEXCUNIQUEBYJOBSIZE	A	80C
WICSRV_RSN_ZIIPEXCBUCEXCUNIQUEBYJOBSIZE	A	80D
WICWR_#MAXVER	60	1
WICWR_#VER	60	1
WICWR_#VER1	60	1
WICWR_AGGBUCKET	0	
WICWR_AGGBUCKET_CONTRIBUTORS	0	
WICWR_AGGBUCKET_DATA	2	
WICWR_AGGBUCKET_LEN	2	2
WICWR_AGGBUCKETENTRYLENGTH	4	
WICWR_AGGBUCKETENTRYWRITELENGTH	8	
WICWR_AGGBUCKETEXPAREA	0	
WICWR_AGGBUCKETNUMOFENTRIES	0	
WICWR_AGGBUCKETPARMLIST	0	
WICWR_AGGBUCKETPARMLIST_LEN	10	18
WICWR_AGGBUCKETPARMLIST1	20	
WICWR_AGGBUCKETPARMLIST2	38	
WICWR_AGGBUCKETSTARTADDR	10	
WICWR_CUSTOMJOBSIZEENTRIES	7	
WICWR_EXCBUCKET	0	
WICWR_EXCBUCKET_ASID	0	
WICWR_EXCBUCKET_DATA	A	
WICWR_EXCBUCKET_JOBNAME	2	
WICWR_EXCBUCKET_LEN	A	A
WICWR_EXCBUCKETENTRYLENGTH	58	
WICWR_EXCBUCKETENTRYWRITELENGTH	5C	
WICWR_EXCBUCKETEXPAREA	0	
WICWR_EXCBUCKETNUMOFACTIVITIES	54	
WICWR_EXCBUCKETNUMOFENTRIES	50	
WICWR_EXCBUCKETSTARTADDR	60	
WICWR_EXCEPTIONBUCKETPARMLIST	50	

Table 26. Cross Reference for IFAWICCB (continued)

Name	Offset	Hex Tag
WICWR_FLAG	6	
WICWR_FLAGEXCJOBSUNIQUEBYJOBSIZE	6	80
WICWR_FLAGRSVD	6	7F
WICWR_KCPUTYPECP	A	0
WICWR_KCPUTYPEDEFFIRST	A	0
WICWR_KCPUTYPEDEFLAST	A	1
WICWR_KCPUTYPEDEFNUM	A	2
WICWR_KCPUTYPEDEFNUMCPONLY	A	1
WICWR_KCPUTYPEZIIP	A	1
WICWR_KCUSTOMJOBSIZEENTRIESMAX	60	40
WICWR_KJOBPRIORITYALL	A	0
WICWR_KJOBPRIORITYCRITICAL	A	1
WICWR_KJOBPRIORITYDEFFIRST	A	0
WICWR_KJOBPRIORITYDEFLAST	A	4
WICWR_KJOBPRIORITYDEFNUM	A	5
WICWR_KJOBPRIORITYDISC	A	4
WICWR_KJOBPRIORITYHIGH	A	2
WICWR_KJOBPRIORITYLOW	A	3
WICWR_KJOBSIZEALL	A	0
WICWR_KJOBSIZEDEFFIRST	A	0
WICWR_KJOBSIZEDEFLAST	A	4
WICWR_KJOBSIZEDEFNUM	A	5
WICWR_KJOBSIZEDEFNUMREQ	A	1
WICWR_KJOBSIZELARGE	A	1
WICWR_KJOBSIZEMEDIUM	A	2
WICWR_KJOBSIZESMALL	A	3
WICWR_KJOBSIZETINY	A	4
WICWR_KMAXVER	60	1
WICWR_KNUMENTSONECPUTYPE	A	19
WICWR_KNUMENTSTWOCPUTYPE	A	32
WICWR_KVER	60	1
WICWR_KVER1	60	1
WICWR_PARMLIST	0	
WICWR_PARMLIST_LEN	60	68
WICWR_PRODLEVEL	10	
WICWR_PROTOTYPEINDEX	C	
WICWR_RELEASEINDEX	8	
WICWR_RSVD0D	E	
WICWR_RSVD02	2	

Table 26. Cross Reference for IFAWICCB (continued)

Name	Offset	Hex Tag
WICWR_RSVD2B	C	
WICWR_SUBTYPEINFO	8	
WICWR_VERSION	0	
WICWR_WITHINRELEASEINDEX	A	





---

## Appendix A. Accessibility

Accessible publications for this product are offered through [IBM Documentation for z/OS \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see [How to Send Feedback to IBM](#) to leave documentation feedback.



## Notices

---

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation  
Site Counsel  
2455 South Road*

Poughkeepsie, NY 12601-5400  
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

## Terms and conditions for product documentation

---

Permissions for the use of these publications are granted subject to the following terms and conditions.

### **Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

### **Personal use**

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

### **Commercial use**

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

## **Rights**

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## **IBM Online Privacy Statement**

---

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at [ibm.com/privacy](http://ibm.com/privacy) and IBM's Online Privacy Statement at [ibm.com/privacy/details](http://ibm.com/privacy/details) in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at [ibm.com/software/info/product-privacy](http://ibm.com/software/info/product-privacy).

## **Policy for unsupported hardware**

---

Various z/OS elements, such as DFSMSdfp, JES2, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

## **Minimum supported hardware**

---

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS \(www.ibm.com/software/support/systemsz/lifecycle\)](http://www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

## Trademarks

---

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [Copyright and Trademark information \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

UNIX is a registered trademark of The Open Group in the United States and other countries.

---

# Index

## A

accessibility  
    contact IBM [173](#)  
aggregate bucket  
    instance [33](#)  
assistive technologies [173](#)

## C

contact  
    z/OS [173](#)  
correlator records, SMF [12](#)

## D

data areas, MVS [94](#)  
Db2  
    statistics record [22](#)

## F

features  
    list of [1](#)

## H

hardware event data collection  
    start, configure, and stop [48](#)  
Hardware Instrumentation Services (HIS)  
    start, configure, and stop [48](#)  
HFTSINTVL parameter [7](#)

## I

IBM z/OS Workload Interaction Correlator  
    collecting records [10](#)  
    enabling [9](#)  
    optional feature [1](#), [4](#)  
    recording to an in-memory resource [11](#)  
    subtype records  
        aggregate bucket sections [19](#)  
        data section [18](#)  
        exceptional job sections [20](#)  
    writing records to the file system [11](#)  
IFAWIC service [79](#)

## K

keyboard  
    navigation [173](#)  
    PF keys [173](#)  
    shortcut keys [173](#)

## M

MODIFY command  
    for hardware event data collection [48](#)  
MVS data areas [94](#)  
MVS messages for z/OS Workload Interaction Correlator [41](#)

## N

navigation  
    keyboard [173](#)

## O

optional features  
    list of [1](#)

## P

parmlib member  
    SMFPRMxx [41](#)

## S

shortcut keys [173](#)  
SMF (system management facilities) [47](#)  
SMF records, correlator [12](#)  
SMFPRMxx parmlib member  
    description [41](#)  
    syntax format [42](#)

## T

trademarks [178](#)

## U

user interface  
    ISPF [173](#)  
    TSO/E [173](#)

## W

WIC parameter [8](#)  
workload interaction correlator [23](#), [79](#)  
    *See also* z/OS workload interaction correlator  
Workload Interaction Correlator  
    applications [23](#)  
    enabling [7](#)

## Z

z/OS Advanced Data  
    Gatherer  
    optional feature [1](#)

- z/OS workload interaction
  - correlator
    - aggregate buckets [32](#)
    - exception bucket [36](#)
    - exit routine services
      - build and write record [29](#)
      - clear buffer [28](#)
      - get next buffer [26](#)
    - exploiter exit routine [23](#)
    - what is [1](#)
- z/OS Workload Interaction Correlator [79](#)







Product Number: 5655-ZOS