CA MICS® Resource Management

Web Analyzer™ Option Guide

Release 12.9

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Contents

Chapter 1: OVERVIEW

The CA MICS Web Analyzer processes data collected by standard IBM SMF record type 103 and 120, as well as from the IBM HTTP Server for z/OS log files. Data from these sources can provide information (all sources do not have to be available) that helps you manage this critical web hosting IT service.

CA MICS automatically integrates Web Analyzer data with any other data that CA MICS maintains. This is possible because CA MICS stores data in a format that is common to all CA MICS products.

The CA MICS Web Analyzer processes SMF record type 103 Performance and Configuration data, SMF record type 120 IBM WebSphere Application Server (WAS) data, and IBM HTTP Server for z/OS log files containing: Server Access, FRCA (Cache Access), Proxy (Server and Cache), Agent (web browser) logs, Referer logs, and the Server Error logs.

The CA MICS Web Analyzer is one of the many Data Integration Analyzers of the CA MICS I/S Management Support System. Like the other Data Integration Analyzers, it processes raw data for analyzing and managing specific technologies.

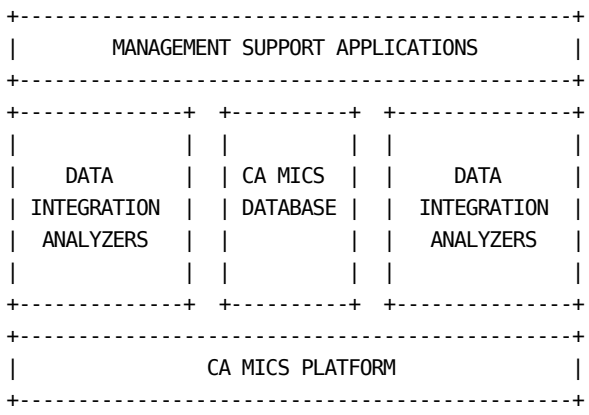


Figure 1-1. The CA MICS I/S Management Support System

The CA MICS I/S Management Support System is a comprehensive, flexible application system that applies standard management practices to the management of the I/S organization. The CA MICS I/S Management Support System, designed with a formal architecture, uses integrated applications that together provide a comprehensive tool for the effective management of your entire IT infrastructure resources.

Figure 1-2 illustrates the Web Analyzer data flow.

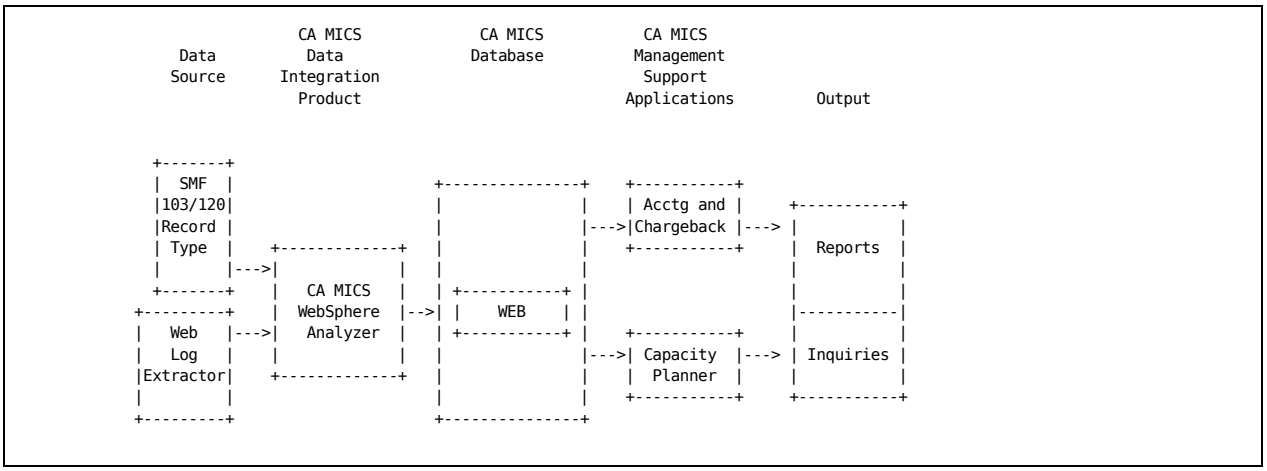


Figure 1-2. CA MICS Web Analyzer Data Flow

The CA MICS Web Analyzer integrates the SMF records generated by IBM HTTP Server for z/OS (SMF type 103), IBM WebSphere Application Server for z/OS (SMF type 120), and IBM HTTP Server Logs, into the CA MICS database. This process creates files in WEB and WLG Information Areas that can be accessed by other CA MICS products:

- CA MICS Q&R WorkStation - See Section 1.1
- CA MICS Accounting and Chargeback - See Section 2.2.2
- CA MICS Capacity Planner - See Section 2.2.1

This section contains the following topics:

1.1 Reporting and Inquiry Facilities

The CA MICS Web Analyzer's standard reports are produced by online inquiries and provide concise and comprehensive information to help manage the Web environment. The CA MICS Web Analyzer provides performance and usage reports that can either be used as delivered or tailored to your specific requirements. In addition to CA MICS Query and Reporting (discussed in Chapter 2), the CA MICS platform provides interactive capabilities that allow quick responses to complex requests. These capabilities are provided by:

- The CA MICS Information Center Facility (MICF) - MICF is a panel-oriented productivity tool that allows you to access information in the CA MICS database. MICF dynamically builds an inquiry program that executes in either interactive or batch mode. Depending on your specific requirements, MICF supports the ability to print the inquiry report, display the results at your terminal, catalog the results for later viewing, or generate a Comma Separated Variable (CSV) file that can be used for loading in the applications and as CA MICS Q&R Workstation.

- The standard SAS language interface - The CA MICS Workstation Facility (MWF) provides an online environment that supports the standard SAS language interface in both interactive and batch modes. SAS allows easy data manipulation, statistical analysis, and provides coding facilities for programmers who are conducting extensive analysis or designing new I/S business reports. CA MICS uses SAS as a data manager and as an advanced analysis language.

- CA MICS Q&R WorkStation - The CA MICS Q&R WorkStation is an optional CA MICS product that provides desktop client server based access to all data stored in the CA MICS mainframe repository. The product uses a wizard interface providing for the rapid development of charting queries with little effort and provides for automated publishing of the reports on your corporate intranet. CA MICS Q&R WorkStation also interfaces with the CA MICS Information Center Facility (MICF), which is a panel-oriented and mainframe based reporting tool included with the CA MICS platform.

Each type of report is discussed in detail in the following subsections:

1 - Management Control Reports
2 - Exception Analyzer Reports

1.1.1 Management Control Reports

MWF production reporting provides you with the ability to select any MICF inquiries, including those distributed with the CA MICS Web Analyzer, in a production mode.

1.1.2 Exception Analyzer Reports

The Exception Analyzer Reports are not available for the CA MICS Web Analyzer.

1.2 Files Overview

This section describes the source and content of each CA MICS file in the CA MICS Web Analyzer information areas.

The CA MICS Web Analyzer's data is divided into the following information areas:

- WEB Information Area (WEB)

- WLG Log Files Information Area (WLG)

The following files are contained in the WEB Information Area of the CA MICS database:

- HTTP Server Configuration File (WEBWCF)
- HTTP Server Performance File (WEBWPR)
- HTTP Server Performance Suspend File (_WPWPR)
- JVM Heap Interval File (WEBHPI)
- HTTP Session Manager Interval File (WEBHTP)
- WAS Server Activity File (WEBSAF)
- WAS Server Interval File (WEBSIF)
- J2EE Method Interval File (WEBMIJ)
- WebContainer Servlet Interval File (WEBSRV)
- Web Request Activity File (WEBWRA)
- WRA CPU Delta Suspend File (WEBWRS)
- Web CPU Usage Breakdown File (WEBWCB)
- Web User Data File (WEBWUD)

The following files are contained in the WLG Log Files Information Area of the CA MICS database:

- Server Access Log File (WLGSAL)
- Agent Log File (WLGAGL)
- Referer Log File (WLGRFL)
- Error Messages Log File (WLGERL)
- WLE Configuration File (WLGCFG)
- Access Method Summary File (WLGAMS)
- Access Path Summary File (WLGAPS)

The files and data elements that comprise the WEB and WLG information areas are described in detail in Chapter 5.

Figure 1-3 illustrates the data sources and files of the CA MICS Web Analyzer's Web Information Area and the Log Files Information Area.

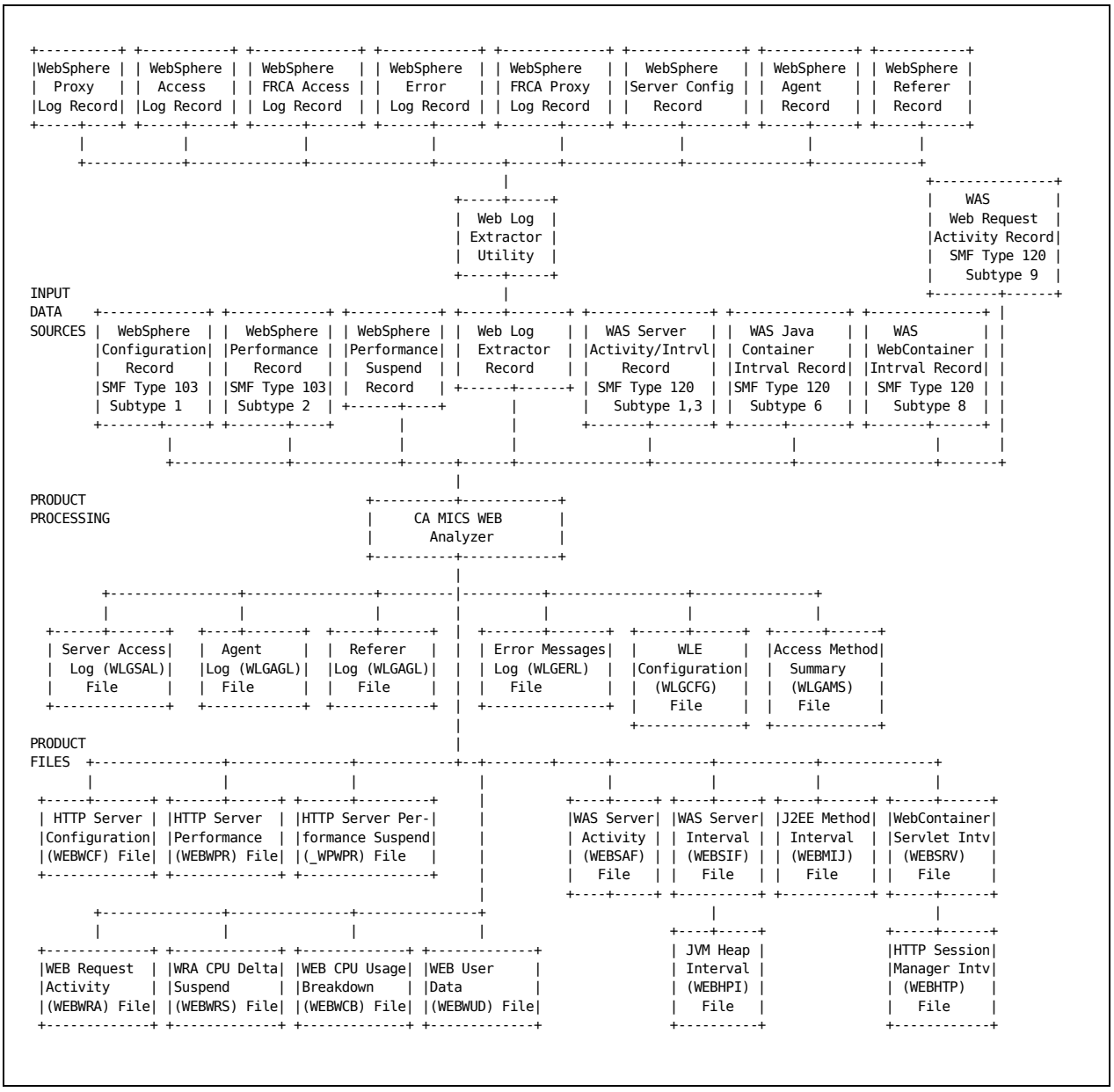


Figure 1-3. Performance & Log Files Information Area Structure

1.3 Product Prerequisites

The CA MICS Web Analyzer runs in a CA MICS environment operating under OS/390 or z/OS.

The CA MICS Web Analyzer supports the WebSphere Application Server and the IBM HTTP Server.

To ensure CA MICS Web Analyzer functionality, you must provide at least one of the following types of data:

- SMF type 103
- SMF type 120
- WLE generated records built from HTTP log files

Procedures for log collection are found in Appendix C, Web Log Extractor (WLE) Overview. Instructions for SMF type 103 record activation are discussed in Chapter 6, Data Sources. Additional information is also available in the IBM documentation for HTTP Server for OS/390.

Chapter 2: USAGE GUIDELINES

Data from the CA MICS Web Analyzer lends significant value to the use of other CA MICS products. These uses are described briefly in the following sections. For more detailed information refer to the individual product guides.

The following sections describe methods for using the Analyzer's data and the interfaces between the CA MICS Web Analyzer and other CA MICS products.

This section contains the following topics:

2.1 Data Analysis

Following are resource management functions that can be pursued using the CA MICS Web Analyzer's data:

Capacity

The CA MICS Web Analyzer database files provides utilization information that can be used for determining current and future capacity requirements.

Performance

The CA MICS Web Analyzer can be used to evaluate the performance of IBM's WebSphere web servers. Cache utilization and error analysis are two important performance issues to consider. The relationship between cache performance and response time can be tracked historically to assist in ensuring better response time. In addition, rich error metrics allow the discovery and addressing of internal problems.

Workload Characterization

A necessary task in any capacity evaluation effort is workload characterization. The CA MICS Web Analyzer can provide data to determine a system's primary application and usage patterns.

Trends

The CA MICS database structure is ideal for trending analysis--identifying behavior patterns over time.

2.2 CA MICS Product Interfaces

The CA MICS Web Analyzer processes, validates, and organizes raw data into meaningful information areas and files on a daily basis. This data repository is integrated with other CA MICS products to provide a comprehensive management solution. The following sections describe these interfaces in detail:

1 - CA MICS Capacity Planner Interface
2 - CA MICS Accounting Interface
3 - Additional Interfaces

2.2.1 CA MICS Capacity Planner Interface

The CA MICS Capacity Planner provides an interface to assist clients in performing effective workload forecasting of the WebSphere Application Server.

The WEB Planning Standard Application uses a capacity planning database file with the data elements required for usage reporting and forecasting of the WebSphere Application Server workload. It uses the Application Unit (CAPAPU) feature of the CA MICS Capacity Planner to group the individual WEB servers into logically similar workload groups for more meaningful analysis and reporting. The three methods for grouping WEB servers together are:

- defining each WEB server as its own workload group
- combining all WEB servers (no workload group)
- defining customized workload groupings.

Each WEB server is uniquely identified by a combination of its TCP/IP address and port number. The CA MICS Web Analyzer includes this information in the key structure for all timespans of the WebSphere Performance file. The CA MICS Web Analyzer also provides additional data elements that can be used to categorize and identify individual web servers, or groups of web servers, by their association with workloads, applications, or other meaningful business or installation related names. These data elements can be used to populate keys in the Capacity Planning database, such as the Capacity Planning Application Unit (CAPAPU), allowing studies and forecasts that focus on the business views of the web servers deployed in your environment. The following data elements provided by the CA MICS Web Analyzer should be considered as candidates for development of keys in a Capacity Planning database:

WEBSTYPE - Web Server Type
WEBSYSNM - System Name
WEBSRVVR - Server Version
WEBNAME - Web Server Long Name (User Defined)
WEBSNAME - Web Server Short Name (User Defined)
WEBGROUP - Web Group Name (User Defined)

The CA MICS Web Analyzer provides the following additional data elements from the CA MICS WebSphere Performance file for the CA MICS Capacity Planner:

- Data elements common to all CA MICS Capacity Planner database files:

  SYSID
  ZONE or HOUR
  YEAR
  MONTH or WEEK
  DATE

o Data elements unique to the WEB Planning Standard Application database file:

WPRREQR - Requests Received
WPRREQS - Requests Sent
WPRBYTR - Bytes Received
WPRBYTS - Bytes Sent

Overall, the integration of the CA MICS Web Analyzer and the CA MICS Capacity Planner provides a set of tools for addressing capacity management in the WEB application environment.

2.2.2 Accounting and Chargeback Interface

The CA MICS Web Analyzer provides data elements that CA MICS Accounting and Chargeback uses to bill internet traffic. Data provided to CA MICS Accounting and Chargeback supports accounting for consumed resources (e.g., bytes transferred) and almost any other measure of resource utilization. The elements that can be charged for are:

CHARGING ELEMENT	CHARGING ELEMENT DESCRIPTION
SAFBYTET	Total Bytes Transferred
SAFKBYTT	Total Bytes Transferred (Kilobytes)
SAFMBYTT	Total Bytes Transferred (Megabytes)
WRASUPTM	zIIP CPU Time
WRASPNTM	Normalized zIIP CPU Time
WRASUCTM	zIIP Eligible CPU Time on CP
WRAZAPTM	zAAP CPU Time
WRAZPNTM	Normalized zAAP CPU Time
WRAZACTM	zAAP Eligible CPU Time on CP
WRAXCSTM	CP CPU Time Minus zIIP/zAAP Eligible
WRACPUTM	CP CPU Time
SALBYTET	Bytes Transferred
SALKBYTT	Kilobytes Transferred
SALMBYTT	Megabytes Transferred

2.2.3 Additional Interfaces

The CA MICS Information Center Facility (MICF), accessible from the CA MICS Workstation Facility (MWF), is a powerful tool that enables you to create, view, and modify inquiries using data from the CA MICS Web Analyzer. In addition, a standard set of inquiries pertaining to the CA MICS Web Analyzer are provided. They are organized under the reporting group PERFORM.

Using the Production Reporting feature of MICF, you can schedule MICF inquiries to run as part of the CA MICS processing job streams (e.g., DAILY, WEEKLY, MONTHLY). These same queries can interface to CA MICS Q&R WorkStation, and are documented in Chapter 3 of this guide.

2.3 General Ad Hoc Management Reporting

Both the CA MICS Information Facility (MICF) and the CA MICS Q&R WorkStation Facility are ideally suited to satisfy any ad hoc data analysis requests.

Chapter 3: REPORTS

The CA MICS Web Analyzer produces reports using the CA MICS Information Center Facility (MICF), which is part of the CA MICS Workstation Facility (MWF).

MICF inquiries produce meaningful reports from the CA MICS database and provide you with the flexibility to code and save your own reports. You can execute and modify distributed MICF inquiries using the Database Inquiries option from the MICF main menu.

This section contains the following topics:

3.1 MICF Inquiries for the CA MICS Web Analyzer

The table below lists the MICF inquiries distributed with the CA MICS Web Analyzer. The remainder of this chapter follows this category listing. Each section contains a brief description of the inquiry and the panels used to generate it.

The CA MICS Web Analyzer reports are initially delivered with defaults. These defaults will change based on the settings of the last execution of the query. For example, WEBCSA is delivered with a hard-coded default value of DAYS for the timespan. If the user changes the timespan so that WEBCSA will run at the WEEKS timespan, then the next execution of WEBCSA will default to the WEEKS timespan.

The CA MICS Web Analyzer's graphic reports can be produced in color or non-color modes. The line graphics are illustrated in this guide. Data used in the graphic analysis queries can also generate CSV (Comma Separated Value) files for subsequent use in the CA MICS Q&R WorkStation product.

Graphs and reports can be produced at multiple summarization levels. Dates and titles will be adjusted appropriately based on the timespan and summarization level chosen for the execution. After data has been collected for a time period, this summarization capability can be used to "drill down" to a specific time frame. For example, you may wish to analyze trends at a weekly level for the last 6 weeks as a starting point. If last week's data showed erratic behavior at this level you may drill down to the days level, days by hour and/or detail levels, to pinpoint potential causes of the behavior.

CA MICS MICF Category	Report Name	Inquiry ID	Use	For additional information, use ...
Performance Analysis	Performance Summary	WEBSL1	General overview of WebSphere utilization and activity	WEBCSA-WEBCSE
	Top Ten Pages Hit	WEBSL2	Overview of Top Ten Pages Hit	WEBCSE
	Bytes Sent and Received	WEBCSA	Graph of Bytes Sent and Received (HTTP Server)	WEBSL1
	Error Level Response Messages	WEBCSB	Graph of Error Level Response Messages	WEBSL1
	Caching Overview	WEBCSC	Graph of Responses Rec. compared to Cache Hits	WEBSL1
	Average Response Times	WEBCSD	Graph of Average Response Times	WEBSL1
	Method Utilization	WEBCSE	Graph of Total Hits per specific Request Method	WEBSL2
	Bytes Sent and Received	WEBCSF	Graph of Bytes Sent and Received (WAS server)	
	WAS Server HTTP Sessions Statistics	WEBCSG	Graph of Created, Active, Invalidated, Finalized, and Live Sessions	

The following sections describe the individual inquiries:

1 - WEB Tabular Inquiries
2 - WEB Graphic Inquiries

3.1.1 WEB Tabular Inquiries

The CA MICS Web Analyzer provides the following tabular report inquiries:

1 - WEB Performance Summary (WEBSL1)
2 - WEB Top Ten Pages Hit (WEBSL2)

3.1.1.1 WEB Performance Summary (WEBSL1)

The WEB Performance Summary uses measurements from the WebSphere Performance File (WEBWPR). It displays an overall view of the utilization and activity for each of the selected Web Names for the date and/or time ranges you select.

Each report produced displays categorized information on utilization, caching, and errors. The report categories and element labels are listed on the left side of the report pages. Individual columns on the report represent the element's value for the selected granularity level (hour, interval, or date) as denoted above the columns. The last column on the report represents the percent increase or decrease over the time periods represented on the report page.

The following table displays the relationship between report labels, originating file, and CA MICS database source elements for each performance category.

Report Label	File	Element
AVG BYTES PER CACHE HIT	WEBWPR	WPRAVBCH
% OF TOT REQ MET BY CACHE	WEBWPR	WPRPCCAH
AVG BYTES SENT PER SECOND	WEBWPR	WPRBYSPS
MAX BYTES SENT PER SECOND	WEBWPR	WPRMXBSS
AVG BYTES REC. PER SECOND	WEBWPR	WPRBYRPS
MAX BYTES REC. PER SECOND	WEBWPR	WPRMXBRS
MAX THREADS DEFINED	WEBWPR	WPRMXTRD
MAX THREADS USED	WEBWPR	WPRMXTRU
THREADS USED	WEBWPR	WPRTRDCT
REQUESTS RECEIVED	WEBWPR	WPRREQR
REQUESTS DISCARDED	WEBWPR	WPRREQDC
RESPONSES SENT	WEBWPR	WPRRSPS
RESPONSES DISCARDED	WEBWPR	WPRRSPD
BYTES RECEIVED	WEBWPR	WPRBYTR
BYTES SENT	WEBWPR	WPRBYTS
KBYTES READ FROM CACHE	WEBWPR	WPRKBCAC
CACHE HITS	WEBWPR	WPRCACHT
BYTES OF CACHE RAM IN USE	WEBWPR	WPRCACRM
ERROR LEVEL 200 RESPONSES	WEBWPR	WPRL200R
ERROR LEVEL 300 RESPONSES	WEBWPR	WPRL300R
ERROR LEVEL 400 RESPONSES	WEBWPR	WPRL400R
ERROR LEVEL 500 RESPONSES	WEBWPR	WPRL500R
TOTAL ERROR RESPONSES	WEBWPR	WPRERRCT

Note: The label associated to the field DATE in the column titles will vary based on timespan used.

To produce the report:

- Enter the CA MICS Workstation Facility (MWF)
- Choose option 2, CA MICS Information Center Facility (MICF)
- Choose option 2, Database Inquiries

o Select WEBSL1, Catalog Group: PERFORM

o Complete the MICF Execution panel

o Complete the specification panel illustrated below
  (Descriptions of the fields follow the panel)

The result displays on your terminal.

```
+-------------------------------------------------------------------------+
|-----------------------WEB Performance Summary ----------------------|
| Command ===>                                                            |
|                                                                         |
| Composing Inquiry: WEBSL1 - WEB Performance Summary                     |
|                                                                         |
| Database ===> ____________________________________________              |
| Cycle(s) ===> __ - __                                                   |
| Time-span ===> ______ (DETAIL/DAYS/WEEKS/MONTHS)                        |
| Report by hour if time-span is DAYS or WEEKS ===> _ (Y/N)               |
| Generate a report for each value of Zone     ===> _ (Y/N)               |
|                                                                         |
| Data Selection:                                                         |
|                      dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy      |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __       |
|   Hour         ===> __ - __    __ - __    __ - __                      |
|   Zone         ===> _ _ _ _ _ _ _ _ _                                   |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                       |
|   WEB Short Names(WEBSNAME) ===> ____ ____ ____ ____ ____ ____          |
|   WEB Name  ===> ____________________ ____________________              |
|                  ____________________ ____________________              |
|   WEB Group(WEBGROUP) ===> ________ ________ ________ ________          |
|                                                                         |
+-------------------------------------------------------------------------+
```

```
PARAMETER SPECIFICATIONS

_________________________

REQUIRED PARAMETERS:

The following parameters require specification for the
inquiry to execute.

DATABASE.  DEFAULT: None
         Specify up to six database unit IDs (the IDs of the
         CA MICS database units from which files are read).

         If CA MICS files exist in only one database unit,
         this parameter is set automatically.

         To display a selection list of valid database unit
         IDs, specify ?  (question mark).

         Duplicates IDs are not allowed.

CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).

TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)
```

```
REPORT BY HOUR.  DEFAULT: Y
         At the DAYS and WEEKS timespans, reports can be
         produced at an hourly, DAILY, or WEEKLY level.

         Y - (Yes) to Report by HOUR.
         The data is summarized and displayed by hour and
         separate reports are produced for each day or week.

         N - (No) to NOT report by HOUR.
         The data is summarized by day or week and a single
         report listing all days or weeks is produced.

GENERATE A REPORT FOR EACH VALUE OF ZONE. DEFAULT: Y
         Y - (Yes) to generate reports for each ZONE.

         N - (No)  to NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.
```

```
OPTIONAL PARAMETERS:

The following parameters do not require specification for the
inquiry to execute, they are optional.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.
```

```
          The hour ranges should not overlap, but you can
          specify the higher range before the lower range.

          For example, a valid hour range specification is:

            18-23 09-17 02-04

          An input observation is selected if its hour value
          is greater than or equal to the starting hour and
          less than or equal to the ending hour of any
          specified hour range.

ZONE.  DEFAULT: None
          Specify up to six ZONE values.

          Each ZONE a 1 byte character field.

          Duplicates are not allowed.

          Using ZONEs you can restrict the inquiry analysis
          to only those ZONE values specified.

SYSID.  DEFAULT: None
          Specify up to six SYSIDs.

          Each SYSID a 1-4 byte character field.

          Duplicates are not allowed.

          Using SYSIDs you can restrict the inquiry analysis
          to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
          Specify up to six Web Short Names.

          Each Web Short Name is a 1-4 byte character field.

          Duplicates are not allowed.

          Using WEBSNAMEs you can restrict the inquiry
          analysis to only those WEBSNAME values specified.

WEB NAME (WEBNAME).  DEFAULT: None
          Specify up to four Web Names.

          Each Web Name is a 1-20 byte character field.

          Duplicates are not allowed.
```

```
          Using WEBNAMEs you can restrict the inquiry analysis
          to only those WEBNAME values specified.

WEB GROUP (WEBGROUP).  DEFAULT: None
          Specify up to four Web Groups.

          Each Web Group is a 1-8 byte character field.

          Duplicates are not allowed.

          Using WEBGROUPs you can restrict the inquiry
          analysis to only those WEBGROUP values specified.
```

CA
CA MICS WebSphere Analyzer
DETAIL WEB PERFORMANCE SUMMARY
FOR ZONE = 1
SYSTEM = XAL1 WEB NAME = HELP DESK RUNDATE = Sep 10, 2005

INTVL-DAY (HR:MM-DD):	11:16-26	12:58-26	16:11-26	16:58-26	CHANGE %
WEBSPHERE ACTIVITY					
BYTES SENT	42730	78002	18145	94083	120.18 %
AVG BYTES SENT PER SECOND	1	13	2	34	3300.00 %
MAX BYTES SENT PER SECOND	1	13	2	34	3300.00 %
BYTES RECEIVED	3334	6107	1762	5525	65.72 %
AVG BYTES REC. PER SECOND	0	1	0	2	N.A.
MAX BYTES REC. PER SECOND	0	1	0	2	N.A.
THREADS USED	2	2	1	1	-50.00 %
MAX THREADS DEFINED	50	50	50	50	---
MAX THREADS USED	2	2	1	1	-50.00 %
REQUESTS RECEIVED	8	15	4	13	62.50 %
REQUESTS DISCARDED	0	0	0	0	N.A.
RESPONSES SENT	10	15	4	13	30.00 %
RESPONSES DISCARDED	0	0	0	0	N.A.
WEBSPHERE CACHE STATS					
CACHE HITS	5	8	2	6	20.00 %
% TOTAL REQ MET BY CACHE	62.50%	53.33%	50.00%	46.15%	-26.16 %
AVG BYTES PER CACHE HIT	175	81	348	79	-54.86 %

```
| KBYTES READ FROM CACHE          19       26        5       26     36.84 % |
| BYTES OF CACHE RAM IN USE    42090    42090    42090    42090     ---      |
|                                                                            |
|----------------------------------------------------------------------------|
|                                                                            |
| WEBSPHERE ERRORS                                                           |
| ----------------                                                           |
|                                                                            |
| ERROR LEVEL 200 RESPONSES        7       12        3       13     85.71 % |
| ERROR LEVEL 300 RESPONSES        1        3        1        0     N.A.     |
| ERROR LEVEL 400 RESPONSES        2        0        0        0     N.A.     |
| ERROR LEVEL 500 RESPONSES        0        0        0        0     N.A.     |
| TOTAL ERROR RESPONSES           10       15        4       13     30.00 % |
|                                                                            |
------------------------------------------------------------------------------
```

3.1.1.2 WEB Top Ten Pages Hit (WEBSL2)

The WEB Top Ten Pages Hit uses measurements from the WebSphere Access Path Summary File (WLGAPS). It displays a report that ranks top ten most hit web pages for each of the selected Web Names for the date and/or time ranges you select.

Each report produced displays categorized information on the top ten pages hit, broken down by hit category. At the DAYS timespan you should only see hits by either Server or Proxy categories, but not both. At higher (WEEKS, MONTHS) summarized timespans it will be possible to see total hits spread throughout the two access categories. The report categories and element labels are listed on the top of the report page. Individual columns on the report represent the element's value for the appropriate granularity level (hour, interval or date) as denoted above the columns. The last column on the report represents the percent increase or decrease over the time periods represented on the report page.

The following table displays the relationship between report labels, originating file, and CA MICS database source elements for each performance element category.

Report Label	File	Element
SERVER ACCESS LOG COUNT	WLGAPS	APSACNT
FRCA (CACHE) ACCESS LOG COUNT	WLGAPS	APSACCNT
PROXY ACCESS LOG COUNT	WLGAPS	APSPCNT
PROXY CACHE ACCESS LOG COUNT	WLGAPS	APSPCCNT
TOTAL HITS	WLGAPS	APSTOTHT

Note: The label associated to the field DATE in the column titles will vary based on timespan used.

To produce the report:

- Enter the CA MICS Workstation Facility (MWF)
- Choose option 2, CA MICS Information Center Facility (MICF)
- Choose option 2, Database Inquiries
- Select WEBSL2, Catalog Group: PERFORM
- Complete the MICF Execution panel
- Complete the specification panel illustrated below (Descriptions of the fields follow the panel)

```
+-------------------------------------------------------------------------+
|------------------------- WEB Top Ten Pages Hit ----------------------|
| Command ===>                                                            |
|                                                                         |
| Composing Inquiry: WEBSL2 - WEB Top Ten Pages Hit                       |
|                                                                         |
| Database ===> ____________________________________________              |
| Cycle(s) ===> __ - __                                                   |
| Time-span ===> ______ (DAYS/WEEKS/MONTHS)                               |
| Report by hour if time-span is DAYS or WEEKS ===> _ (Y/N)               |
| Generate a report for each value of Zone     ===> _ (Y/N)               |
|                                                                         |
| Data Selection:                                                         |
|                      dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy      |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __       |
|   Hour         ===> __ - __    __ - __    __ - __                      |
|   Zone         ===> _ _ _ _ _ _ _ _ _                                   |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                       |
|   WEB Short Names(WEBSNAME) ===> ____ ____ ____ ____ ____ ____          |
|   WEB Name  ===> ____________________ ____________________              |
|                  ____________________ ____________________              |
|   WEB Group(WEBGROUP) ===> ________ ________ ________ ________          |
|                                                                         |
+-------------------------------------------------------------------------+
```

PARAMETER SPECIFICATIONS

_____

REQUIRED PARAMETERS:

The following parameters require specification for the
inquiry to execute.

DATABASE. DEFAULT: None

Specify up to six database unit IDs (the IDs of the CA MICS database units from which files are read).

If CA MICS files exist in only one database unit, this parameter is set automatically.

To display a selection list of valid database unit IDs, specify ? (question mark).

Duplicates IDs are not allowed.

```
CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).

TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)

REPORT BY HOUR.  DEFAULT: Y
         At the DAYS and WEEKS timespans, reports can be
         produced at an hourly, DAILY, or WEEKLY level.

         Y - (Yes) to Report by HOUR.
         The data is summarized and displayed by hour and
         separate reports are produced for each day or week.

         N - (No) to NOT report by HOUR.
         The data is summarized by day or week and a single
         report listing all days or weeks is produced.

GENERATE A REPORT FOR EACH VALUE OF ZONE.  DEFAULT: Y
         Y - (Yes) to generate reports for each ZONE.

         N - (No)  to NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.
```

```
OPTIONAL PARAMETERS:

The following parameters do not require specification for the
inquiry to execute, they are optional.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.
```

```
          The hour ranges should not overlap, but you can
          specify the higher range before the lower range.

          For example, a valid hour range specification is:

            18-23 09-17 02-04

          An input observation is selected if its hour value
          is greater than or equal to the starting hour and
          less than or equal to the ending hour of any
          specified hour range.

ZONE.  DEFAULT: None
          Specify up to six ZONE values.

          Each ZONE a 1 byte character field.

          Duplicates are not allowed.

          Using ZONEs you can restrict the inquiry analysis
          to only those ZONE values specified.

SYSID.  DEFAULT: None
          Specify up to six SYSIDs.

          Each SYSID a 1-4 byte character field.

          Duplicates are not allowed.

          Using SYSIDs you can restrict the inquiry analysis
          to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
          Specify up to six Web Short Names.

          Each Web Short Name is a 1-4 byte character field.

          Duplicates are not allowed.

          Using WEBSNAMEs you can restrict the inquiry
          analysis to only those WEBSNAME values specified.

WEB NAME (WEBNAME).  DEFAULT: None
          Specify up to four Web Names.

          Each Web Name is a 1-20 byte character field.

          Duplicates are not allowed.
```

Using WEBNAMEs you can restrict the inquiry analysis
to only those WEBNAME values specified.

WEB GROUP (WEBGROUP). DEFAULT: None
Specify up to four Web Groups.

Each Web Group is a 1-8 byte character field.

Duplicates are not allowed.

Using WEBGROUPs you can restrict the inquiry
analysis to only those WEBGROUP values specified.

Daily WEB Top Ten Pages Hit
CA

---------- SYSID=XAL1 ZONE=1 WEBNAME=141.202.198.1/4080 Year of Century=04 Month of Year=11 Day of Month=13 Hour Of Day=8 ---------

Path to File	Total Hits	Server Access Log Count	FRCA (Cache) Access Log Count	Proxy Access Log Count	Proxy Cache Access Log Count
/bookmgr-cgi/bookmgr.exe/Shelves/DCFP	17	17	0	0	0
/bookmgr-cgi/bookmgr.exe/BOOKS/DCFP	5	5	0	0	0
/bookmgr/ftfc.gif	5	5	0	0	0
/bookmgr/ftfo.gif	5	5	0	0	0
/s390cs/share/images/template2/nav_off.gif	4	4	0	0	0
/s390cs/share/images/template2/nav_on.gif	4	4	0	0	0
/	3	3	0	0	0
/bookmgr-cgi/bookmgr.exe/cases/IBMMANX	2	2	0	0	0
/bookmgr-cgi/bookmgr.exe/cases/VENDMANX	2	2	0	0	0
/bookmgr-cgi/bookmgr.exe/BOOKS/EPHM2M00/CONTENTS	2	2	0	0	0

CA MICS WEB WEBSL2 30NOV05 BASED ON Information Summarized by Hour Within Day

3.1.2 WEB Graphic Inquiries

The CA MICS Web Analyzer provides the following graph
generating inquiries:

1 - WEB Bytes Sent and Received (WEBCSA)
2 - WEB Error Level Response Messages (WEBCSB)
3 - WEB Caching Overview (WEBCSC)
4 - WEB Average Response Times (WEBCSD)
5 - WEB Method Utilization (WEBCSE)
6 - WAS Bytes Sent and Received (WEBCSF)
7 - WAS HTTP Sessions Statistics (WEBCSG)

3.1.2.1 WEB Bytes Sent and Received (WEBCSA)

The WEB Bytes Sent and Received Report uses measurements from the WEBSPHERE Performance (WEBWPR) file to contrast the total bytes sent compared to the total bytes received over the date and/or time intervals you request.

The following CA MICS Web Analyzer elements will be selected:

KEY ELEMENTS:

- SYSID - System Identifier
- WEBSNAME - Web Server Short Name
- WEBNAME - Web Server Long Name
- WEBGROUP - Web Group Name
- HOUR - Hour Of Day

DATA ELEMENTS:

- WPRBYTS - Bytes Sent
- WPRBYTR - Bytes Received

Note that the label associated to the field DATE will vary based on timespan used.

To produce the report:

- Enter the CA MICS Workstation Facility (MWF)
- Choose option 2, CA MICS Information Center Facility (MICF)
- Choose option 2, Database Inquiries
- Select WEBCSA, Catalog Group: PERFORM
- Complete the MICF Execution panel
- Complete the specification panel illustrated below (Descriptions of the fields follow the panel)

The result will be displayed on your terminal.

```
+-------------------------------------------------------------------------+
|-----------------------  WEB Bytes Sent and Received ----------------|
| Command ===>                                                            |
|                                                                         |
| Composing Inquiry: WEBCSA - WEB Bytes Sent and Received                 |
|                                                                         |
| Database ===> ____________________________________________              |
| Cycle(s) ===> __ - __                                                   |
| Time-span ===> ______ (DETAIL/DAYS/WEEKS/MONTHS)                        |
| Plot by hour if time-span is DAYS or WEEKS   ===> _ (Y/N)               |
| Generate a plot for each value of Zone       ===> _ (Y/N)               |
| Generate color graphic plots                 ===> _ (Y/N)               |
| Generate CSV Extract for Query and Reporting ===> _ (Y/N)               |
|                                                                         |
| Data Selection:                                                         |
|                     dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy       |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __       |
|   Hour         ===> __ - __    __ - __    __ - __                      |
|   Zone         ===> _ _ _ _ _ _ _ _ _                                   |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                       |
|   WEB Short Names(WEBSNAME) ===> ____ ____ ____ ____ ____ ____          |
|   WEB Name  ===> ____________________ ____________________              |
|                  ____________________ ____________________              |
|   WEB Group(WEBGROUP) ===> ________ ________ ________ ________          |
+-------------------------------------------------------------------------+
```

```
PARAMETER SPECIFICATIONS

_________________________

REQUIRED PARAMETERS:

The following parameters require specification for the
inquiry to execute.

DATABASE.  DEFAULT: None
         Specify up to six database unit IDs (the IDs of the
         CA MICS database units from which files are read).

         If CA MICS files exist in only one database unit,
         this parameter is set automatically.

         To display a selection list of valid database unit
         IDs, specify ?  (question mark).

         Duplicates IDs are not allowed.

CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).

TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)
```

```
PLOT BY HOUR.  DEFAULT: Y
         At the DAYS and WEEKS timespans, reports can be
         produced at an hourly, DAILY, or WEEKLY level.

         Y - (Yes) to Report by HOUR.
         The data is summarized and displayed by hour and
         separate reports are produced for each day or week.

         N - (No) to NOT report by HOUR.
         The data is summarized by day or week and a single
         report listing all days or weeks is produced.

GENERATE A PLOT FOR EACH VALUE OF ZONE.  DEFAULT: Y
         Y - (Yes) to generate reports for each ZONE.

         N - (No)  to NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.


GENERATE COLOR GRAPHIC PLOTS.  DEFAULT: N
        Y - (Yes) generates color graphic plots.  SAS/GRAPH
            software is a prerequisite for color graphics.
        N - (No) Standard SAS plots and graphs will be
            produced (non-color, line formats).

GENERATE CSV EXTRACT FOR QUERY AND REPORTING.  DEFAULT: N
        Y - (Yes) generates an extract CSV file for
            subsequent use in CA MICS Query and Reporting.
            This option is only valid when the CA MICS Query
            and Reporting is installed.  Elements available
            for CSV extract are WPRBYTS and WPRBYTR.
        N - (No) CSV file not produced.
```

```
OPTIONAL PARAMETERS:

The following parameters do not require specification for the
inquiry to execute, they are optional.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.
```

```
          The hour ranges should not overlap, but you can
          specify the higher range before the lower range.

          For example, a valid hour range specification is:

            18-23 09-17 02-04

          An input observation is selected if its hour value
          is greater than or equal to the starting hour and
          less than or equal to the ending hour of any
          specified hour range.

ZONE.  DEFAULT: None
          Specify up to six ZONE values.

          Each ZONE a 1 byte character field.

          Duplicates are not allowed.

          Using ZONEs you can restrict the inquiry analysis
          to only those ZONE values specified.

SYSID.  DEFAULT: None
          Specify up to six SYSIDs.

          Each SYSID a 1-4 byte character field.

          Duplicates are not allowed.

          Using SYSIDs you can restrict the inquiry analysis
          to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
          Specify up to six Web Short Names.

          Each Web Short Name is a 1-4 byte character field.

          Duplicates are not allowed.

          Using WEBSNAMEs you can restrict the inquiry
          analysis to only those WEBSNAME values specified.

WEB NAME (WEBNAME).  DEFAULT: None
          Specify up to four Web Names.

          Each Web Name is a 1-20 byte character field.

          Duplicates are not allowed.
```

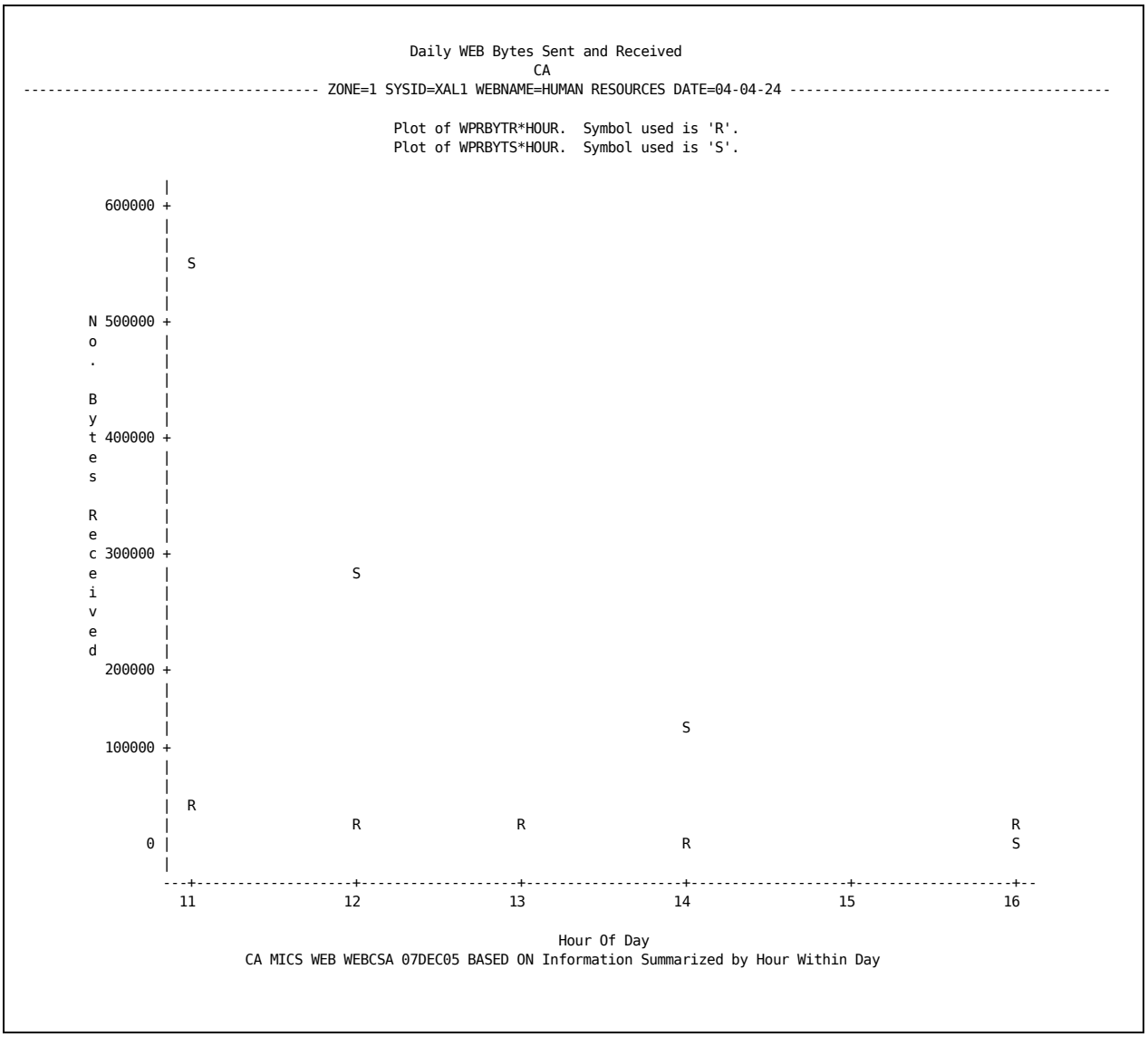
Using WEBNAMEs you can restrict the inquiry analysis
to only those WEBNAME values specified.

WEB GROUP (WEBGROUP). DEFAULT: None
Specify up to four Web Groups.

Each Web Group is a 1-8 byte character field.

Duplicates are not allowed.

Using WEBGROUPs you can restrict the inquiry
analysis to only those WEBGROUP values specified.



3.1.2.2 WEB Error Level Responses (WEBCSB)

The WEB Error Level Responses Report uses measurements from the Performance File (WEBWPR) to display an overall view of the various error response levels over the date and/or time ranges you select. The plot displays each Web Name based on the report summarization options you specify.

The following CA MICS WEB Analyzer elements will be selected:

KEY ELEMENTS:

- SYSID - System Identifier
- WEBSNAME - Web Server Short Name
- WEBNAME - Web Server Long Name
- WEBGROUP - Web Group Name
- HOUR - Hour Of Day

DATA ELEMENTS:

- WPRERRCT - Total Error Responses
- WPRL200R - Error Level 200 Response Count
- WPRL300R - Error Level 300 Response Count
- WPRL400R - Error Level 400 Response Count
- WPRL500R - Error Level 500 Response Count

Note: The label associated to the field DATE will vary based on timespan used.

To produce the report:

- Enter the CA MICS Workstation Facility (MWF)
- Choose option 2, CA MICS Information Center Facility (MICF)
- Choose option 2, Database Inquiries
- Select WEBCSB, Catalog Group: PERFORM
- Complete the MICF Execution panel
- Complete the specification panel illustrated below (Descriptions of the fields follow the panel)

The result displays on your terminal.

```
+-------------------------------------------------------------------------+
|------------------  WEB Error Level Responses  ----------------------|
| Command ===>                                                            |
|                                                                         |
| Composing Inquiry: WEBCSB - WEB Error Level Responses                   |
|                                                                         |
| Database ===> ____________________________________________              |
| Cycle(s) ===> __ - __                                                   |
| Time-span ===> ______ (DETAIL/DAYS/WEEKS/MONTHS)                        |
| Plot by hour if time-span is DAYS or WEEKS   ===> _ (Y/N)               |
| Generate a plot for each value of Zone       ===> _ (Y/N)               |
| Generate color graphic plots                 ===> _ (Y/N)               |
| Generate CSV Extract for Query and Reporting ===> _ (Y/N)               |
|                                                                         |
| Data Selection:                                                         |
|                     dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy       |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __       |
|   Hour         ===> __ - __    __ - __     __ - __                      |
|   Zone         ===> _ _ _ _ _ _ _ _ _ _                                 |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                       |
|   WEB Short Names(WEBSNAME) ===> ____ ____ ____ ____ ____ ____          |
|   WEB Name  ===> ____________________ ____________________              |
|                  ____________________ ____________________              |
|   WEB Group(WEBGROUP) ===> ________ ________ ________ ________          |
+-------------------------------------------------------------------------+
```

PARAMETER SPECIFICATIONS

REQUIRED PARAMETERS:

The following parameters require specification for the
inquiry to execute.

DATABASE. DEFAULT: None

Specify up to six database unit IDs (the IDs of the CA MICS database units from which files are read).

If CA MICS files exist in only one database unit, this parameter is set automatically.

To display a selection list of valid database unit IDs, specify ? (question mark).

Duplicates IDs are not allowed.

```
CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).


DATABASE.  DEFAULT: None
         Specify up to six database unit IDs (the IDs of the
         CA MICS database units from which files are read).

         If CA MICS files exist in only one database unit,
         this parameter is set automatically.

         To display a selection list of valid database unit
         IDs, specify ?  (question mark).

         Duplicates IDs are not allowed.

CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).
```

```
TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)


DATABASE.  DEFAULT: None
         Specify up to six database unit IDs (the IDs of the
         CA MICS database units from which files are read).

         If CA MICS files exist in only one database unit,
         this parameter is set automatically.

         To display a selection list of valid database unit
         IDs, specify ?  (question mark).

         Duplicates IDs are not allowed.

CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).
```

```
TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)

PLOT BY HOUR.  DEFAULT: Y
         At the DAYS and WEEKS timespans, reports can be
         produced at an hourly, DAILY, or WEEKLY level.

         Y - (Yes) to Report by HOUR.
         The data is summarized and displayed by hour and
         separate reports are produced for each day or week.

         N - (No) to NOT report by HOUR.
         The data is summarized by day or week and a single
         report listing all days or weeks is produced.

GENERATE A PLOT FOR EACH VALUE OF ZONE.  DEFAULT: Y
         Y - (Yes) to generate reports for each ZONE.

         N - (No)  to NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.

GENERATE COLOR GRAPHIC PLOTS.  DEFAULT: N
        Y - (Yes) generates color graphic plots.  SAS/GRAPH
            software is a prerequisite for color graphics.
        N - (No) Standard SAS plots and graphs will be
            produced (non-color, line formats).

GENERATE CSV EXTRACT FOR QUERY AND REPORTING.  DEFAULT: N
        Y - (Yes) generates an extract CSV file for
            subsequent use in CA MICS Query and Reporting.
            This option is only valid when the CA MICS Query
            and Reporting is installed.  The elements
            available for CSV extract are WPRERRCT, WPRL200R,
            WPRL300R, WPRL400R, and WPRL500R.
        N - (No) CSV file not produced.
```

```
OPTIONAL PARAMETERS:

The following parameters do not require specification for the
inquiry to execute, they are optional.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.

         The hour ranges should not overlap, but you can
         specify the higher range before the lower range.

         For example, a valid hour range specification is:

           18-23 09-17 02-04

         An input observation is selected if its hour value
         is greater than or equal to the starting hour and
         less than or equal to the ending hour of any
         specified hour range.
```

```
ZONE.  DEFAULT: None
         Specify up to six ZONE values.

         Each ZONE a 1 byte character field.

         Duplicates are not allowed.

         Using ZONEs you can restrict the inquiry analysis
         to only those ZONE values specified.

SYSID.  DEFAULT: None
         Specify up to six SYSIDs.

         Each SYSID a 1-4 byte character field.

         Duplicates are not allowed.

         Using SYSIDs you can restrict the inquiry analysis
         to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
         Specify up to six Web Short Names.

         Each Web Short Name is a 1-4 byte character field.

         Duplicates are not allowed.

         Using WEBSNAMEs you can restrict the inquiry
         analysis to only those WEBSNAME values specified.

WEB NAME (WEBNAME).  DEFAULT: None
         Specify up to four Web Names.

         Each Web Name is a 1-20 byte character field.

         Duplicates are not allowed.

         Using WEBNAMEs you can restrict the inquiry analysis
         to only those WEBNAME values specified.
```

```
WEB GROUP (WEBGROUP).  DEFAULT: None
         Specify up to four Web Groups.

         Each Web Group is a 1-8 byte character field.

         Duplicates are not allowed.

         Using WEBGROUPs you can restrict the inquiry
         analysis to only those WEBGROUP values specified.
DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.

         The hour ranges should not overlap, but you can
         specify the higher range before the lower range.

         For example, a valid hour range specification is:

           18-23 09-17 02-04

         An input observation is selected if its hour value
         is greater than or equal to the starting hour and
         less than or equal to the ending hour of any
         specified hour range.
```

```
ZONE.  DEFAULT: None
         Specify up to six ZONE values.

         Each ZONE a 1 byte character field.

         Duplicates are not allowed.

         Using ZONEs you can restrict the inquiry analysis
         to only those ZONE values specified.

SYSID.  DEFAULT: None
         Specify up to six SYSIDs.

         Each SYSID a 1-4 byte character field.

         Duplicates are not allowed.

         Using SYSIDs you can restrict the inquiry analysis
         to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
         Specify up to six Web Short Names.

         Each Web Short Name is a 1-4 byte character field.

         Duplicates are not allowed.

         Using WEBSNAMEs you can restrict the inquiry
         analysis to only those WEBSNAME values specified.

WEB NAME (WEBNAME).  DEFAULT: None
         Specify up to four Web Names.

         Each Web Name is a 1-20 byte character field.

         Duplicates are not allowed.

         Using WEBNAMEs you can restrict the inquiry analysis
         to only those WEBNAME values specified.
```

```
WEB GROUP (WEBGROUP).  DEFAULT: None
         Specify up to four Web Groups.

         Each Web Group is a 1-8 byte character field.

         Duplicates are not allowed.

         Using WEBGROUPs you can restrict the inquiry
         analysis to only those WEBGROUP values specified.
DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.

         The hour ranges should not overlap, but you can
         specify the higher range before the lower range.

         For example, a valid hour range specification is:

           18-23 09-17 02-04

         An input observation is selected if its hour value
         is greater than or equal to the starting hour and
         less than or equal to the ending hour of any
         specified hour range.
```

```
ZONE.  DEFAULT: None
         Specify up to six ZONE values.

         Each ZONE a 1 byte character field.

         Duplicates are not allowed.

         Using ZONEs you can restrict the inquiry analysis
         to only those ZONE values specified.

SYSID.  DEFAULT: None
         Specify up to six SYSIDs.

         Each SYSID a 1-4 byte character field.

         Duplicates are not allowed.

         Using SYSIDs you can restrict the inquiry analysis
         to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
         Specify up to six Web Short Names.

         Each Web Short Name is a 1-4 byte character field.

         Duplicates are not allowed.

         Using WEBSNAMEs you can restrict the inquiry
         analysis to only those WEBSNAME values specified.

WEB NAME (WEBNAME).  DEFAULT: None
         Specify up to four Web Names.

         Each Web Name is a 1-20 byte character field.

         Duplicates are not allowed.

         Using WEBNAMEs you can restrict the inquiry analysis
         to only those WEBNAME values specified.
```

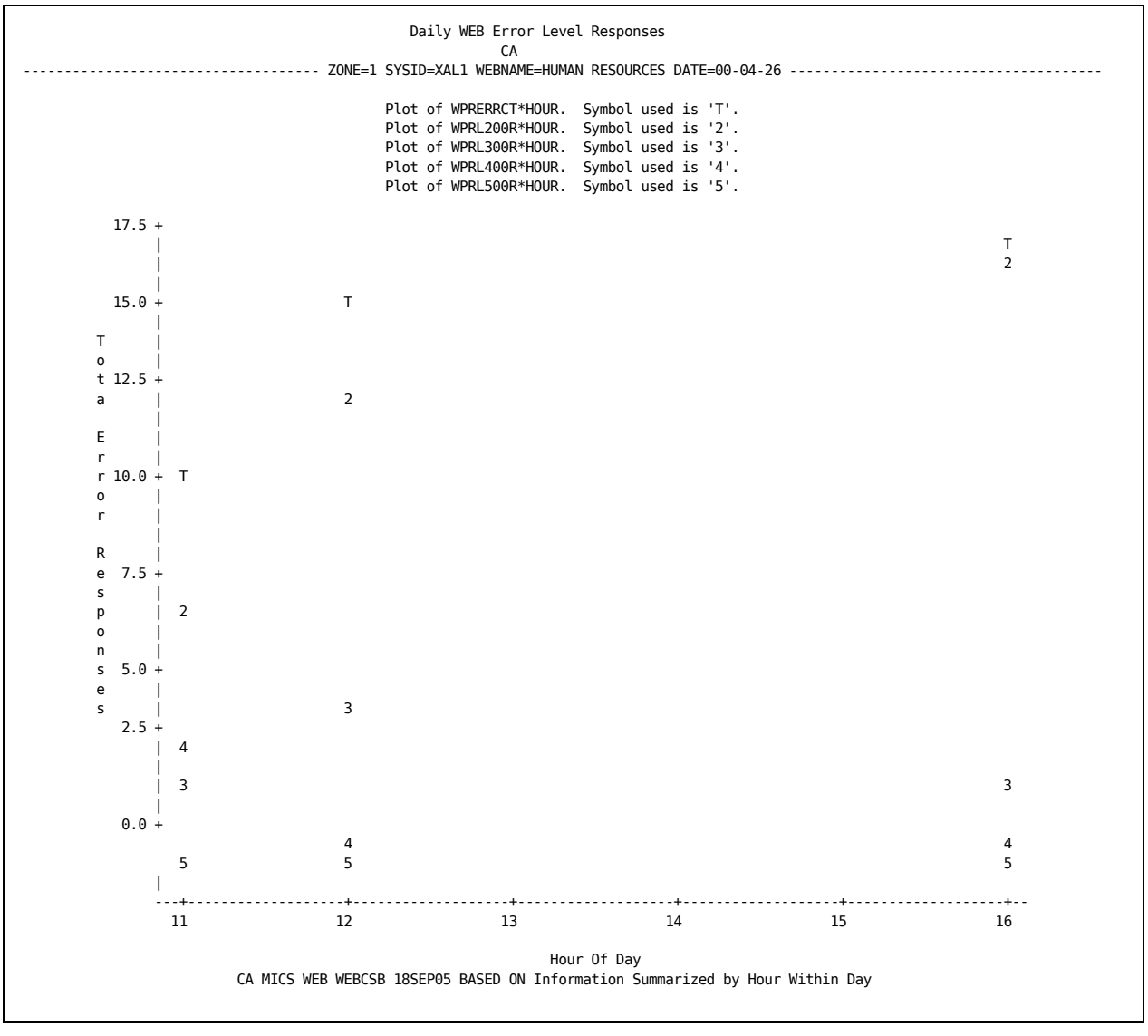
WEB GROUP (WEBGROUP). DEFAULT: None

Specify up to four Web Groups.

Each Web Group is a 1-8 byte character field.

Duplicates are not allowed.

Using WEBGROUPs you can restrict the inquiry analysis to only those WEBGROUP values specified.



3.1.2.3 WEB Caching Overview (WEBCSC)

The WEB Caching Overview Report uses measurements from the Performance File (WEBWPR) to display an overall view of the total requests received compared with the number of cache hits over the date and/or time ranges you select. The plot displays each Web Name based on the report summarization options you specify.

The following CA MICS Web Analyzer elements will be selected:

KEY ELEMENTS:

- o SYSID - System Identifier
- o WEBSNAME - Web Server Short Name
- o WEBNAME - Web Server Long Name
- o WEBGROUP - Web Group Name
- o HOUR - Hour Of Day

DATA ELEMENTS:

- o WPRREQR - Requests Received
- o WPRCACHT - Cache Hits

Note: The label associated to the field DATE will vary based on timespan used.

To produce the report:

- o Enter the CA MICS Workstation Facility (MWF)
- o Choose option 2, CA MICS Information Center Facility (MICF)
- o Choose option 2, Database Inquiries
- o Select WEBCSC, Catalog Group: PERFORM

o Complete the MICF Execution panel

o Complete the specification panel illustrated below
(Descriptions of the fields follow the panel)

The result displays on your terminal.

```
+-------------------------------------------------------------------------+
|--------------------------- WEB Caching Overview --------------------|
| Command ===>                                                            |
|                                                                         |
| Composing Inquiry: WEBCSC - WEB Caching Overview                        |
|                                                                         |
| Database ===> ____________________________________________              |
| Cycle(s) ===> __ - __                                                   |
| Time-span ===> ______ (DETAIL/DAYS/WEEKS/MONTHS)                        |
| Plot by hour if time-span is DAYS or WEEKS   ===> _ (Y/N)               |
| Generate a plot for each value of Zone       ===> _ (Y/N)               |
| Generate color graphic plots                 ===> _ (Y/N)               |
| Generate CSV Extract for Query and Reporting ===> _ (Y/N)               |
|                                                                         |
| Data Selection:                                                         |
|                     dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy       |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __       |
|   Hour         ===> __ - __     __ - __     __ - __                     |
|   Zone         ===> _ _ _ _ _ _ _ _ _                                   |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                       |
|   WEB Short Names(WEBSNAME) ===> ____ ____ ____ ____ ____ ____          |
|   WEB Name  ===> ____________________ ____________________              |
|                  ____________________ ____________________              |
|   WEB Group(WEBGROUP) ===> ________ ________ ________ ________          |
+-------------------------------------------------------------------------+
```

```
PARAMETER SPECIFICATIONS

_________________________

REQUIRED PARAMETERS:

The following parameters require specification for the
inquiry to execute.

DATABASE.  DEFAULT: None
         Specify up to six database unit IDs (the IDs of the
         CA MICS database units from which files are read).

         If CA MICS files exist in only one database unit,
         this parameter is set automatically.

         To display a selection list of valid database unit
         IDs, specify ?  (question mark).

         Duplicates IDs are not allowed.

CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).

TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)
```

```
PLOT BY HOUR.  DEFAULT: Y
         At the DAYS and WEEKS timespans, reports can be
         produced at an hourly, DAILY, or WEEKLY level.

         Y - (Yes) to Report by HOUR.
         The data is summarized and displayed by hour and
         separate reports are produced for each day or week.

         N - (No) to NOT report by HOUR.
         The data is summarized by day or week and a single
         report listing all days or weeks is produced.

GENERATE A PLOT FOR EACH VALUE OF ZONE.  DEFAULT: Y
         Y - (Yes) to generate reports for each ZONE.

         N - (No)  to NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.

GENERATE COLOR GRAPHIC PLOTS.  DEFAULT: N
        Y - (Yes) generates color graphic plots.  SAS/GRAPH
            software is a prerequisite for color graphics.
        N - (No) Standard SAS plots and graphs will be
            produced (non-color, line formats).

GENERATE CSV EXTRACT FOR QUERY AND REPORTING.  DEFAULT: N
        Y - (Yes) generates an extract CSV file for
            subsequent use in CA MICS Query and Reporting.
            This option is only valid when the CA MICS Query
            and Reporting is installed.  The elements
            available for CSV extract are WPRREQR and
            WPRCACHT.
        N - (No) CSV file not produced.
```

```
OPTIONAL PARAMETERS:

The following parameters do not require specification for the
inquiry to execute, they are optional.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.

         The hour ranges should not overlap, but you can
         specify the higher range before the lower range.

         For example, a valid hour range specification is:

           18-23 09-17 02-04

         An input observation is selected if its hour value
         is greater than or equal to the starting hour and
         less than or equal to the ending hour of any
         specified hour range.
```

```
ZONE.  DEFAULT: None
         Specify up to six ZONE values.

         Each ZONE a 1 byte character field.

         Duplicates are not allowed.

         Using ZONEs you can restrict the inquiry analysis
         to only those ZONE values specified.

SYSID.  DEFAULT: None
         Specify up to six SYSIDs.

         Each SYSID a 1-4 byte character field.

         Duplicates are not allowed.

         Using SYSIDs you can restrict the inquiry analysis
         to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
         Specify up to six Web Short Names.

         Each Web Short Name is a 1-4 byte character field.

         Duplicates are not allowed.

         Using WEBSNAMEs you can restrict the inquiry
         analysis to only those WEBSNAME values specified.

WEB NAME (WEBNAME).  DEFAULT: None
         Specify up to four Web Names.

         Each Web Name is a 1-20 byte character field.

         Duplicates are not allowed.

         Using WEBNAMEs you can restrict the inquiry analysis
         to only those WEBNAME values specified.
```

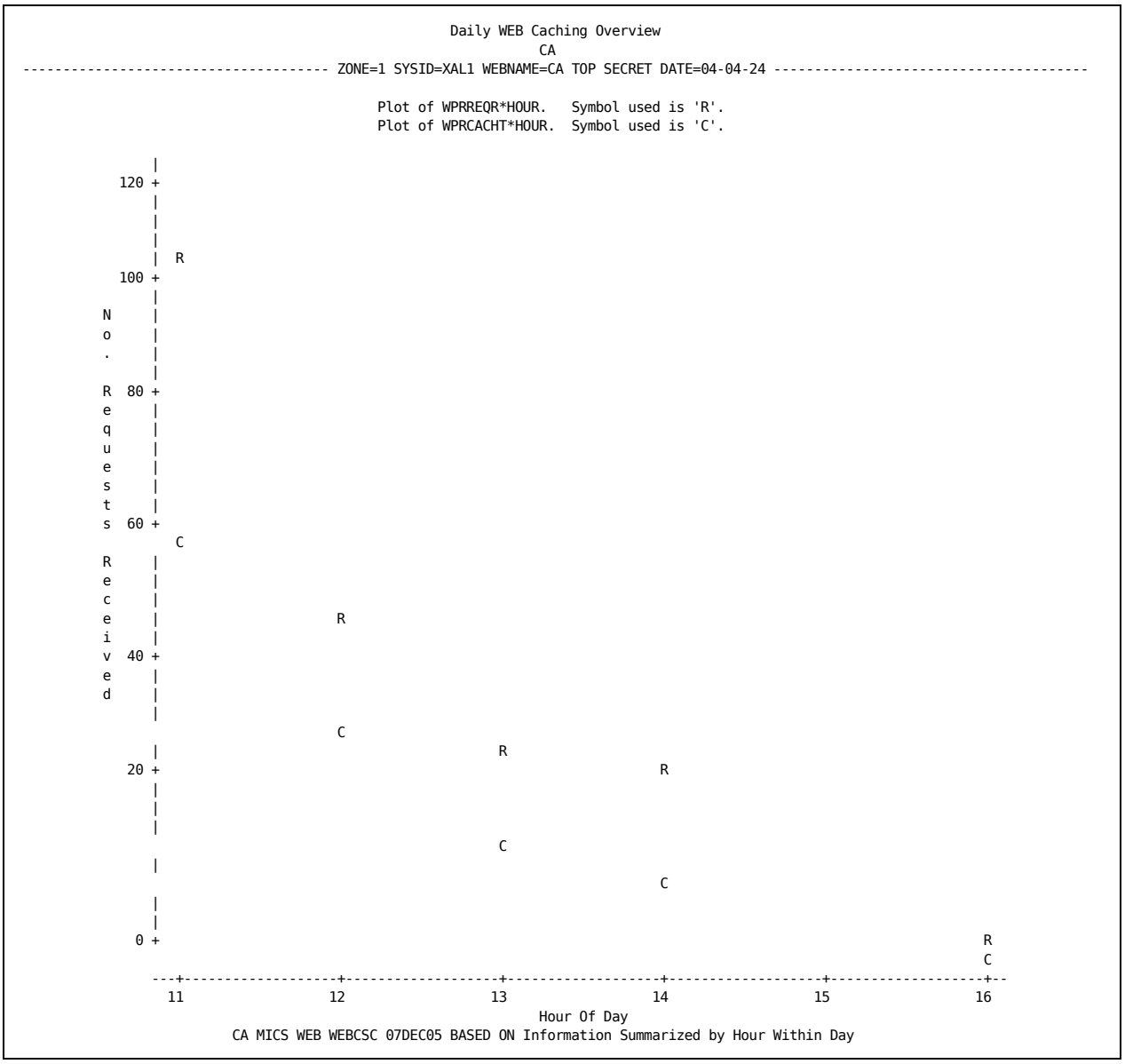
WEB GROUP (WEBGROUP). DEFAULT: None

Specify up to four Web Groups.

Each Web Group is a 1-8 byte character field.

Duplicates are not allowed.

Using WEBGROUPs you can restrict the inquiry analysis to only those WEBGROUP values specified.



3.1.2.4 Web Average Response Times (WEBCSD)

The Web Average Response Times Report uses measurements from the Performance File (WEBWPR) to display an overall view of various average response times, at the detail timespan, over the date and/or time ranges you select. The plot displays each web name based on the report summarization options you specify.

The following CA MICS Web Analyzer elements will be selected:

KEY ELEMENTS:

- SYSID - System Identifier
- WEBSNAME - Web Server Short Name
- WEBNAME - Web Server Long Name
- WEBGROUP - Web Group Name
- HOUR - Hour Of Day

DATA ELEMENTS:

- WPRAVDNS - Avg DNS Lookup Response Time
- WPRAVSPL - Avg Service Plugins Response Time
- WPRAVCGI - Avg CGI Response Time
- WPRAVSSH - Avg SSL Handshake Response Time
- WPRAVPRX - Avg Proxy Response Time

To produce the report:

- Enter the CA MICS Workstation Facility (MWF).
- Choose option 2, CA MICS Information Center Facility (MICF).
- Choose option 2, Database Inquiries.
- Select WEBCSD, Catalog Group: PERFORM.
- Complete the MICF Execution panel. Note that the timespan has been preset. (If you are not familiar with MICF, refer to the MICF User Guide.)
- Complete the specification panel illustrated below. (Descriptions of the fields follow the panel.)

The result displays on your terminal.

```
+------------------------------------------------------------------------+
|------------------------Web Average Response Times--------------------|
| Command ===>                                                           |
|                                                                        |
| Composing Inquiry: WEBCSD - Web Average Response Times                 |
|                                                                        |
| Database ===> ____________________________________________             |
| Cycle(s) ===> __ - __                                                  |
| Timespan ===> DETAIL                                                   |
| Generate a plot for each value of Zone       ===> _ (Y/N)              |
| Generate color graphic plots                 ===> _ (Y/N)              |
| Generate CSV Extract for Query and Reporting ===> _ (Y/N)              |
|                                                                        |
| Data Selection:                                                        |
|                     dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy      |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __      |
|   Hour         ===> __ - __     __ - __     __ - __                   |
|   Zone         ===> _ _ _ _ _ _ _ _ _                                  |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                      |
|   Web Short Names(WEBSNAME) ===> ____ ____ ____ ____ ____ ____         |
|   Web Name  ===> ____________________ ____________________             |
|                  ____________________ ____________________             |
|   Web Group(WEBGROUP) ===> ________ ________ ________ ________         |
+------------------------------------------------------------------------+
```

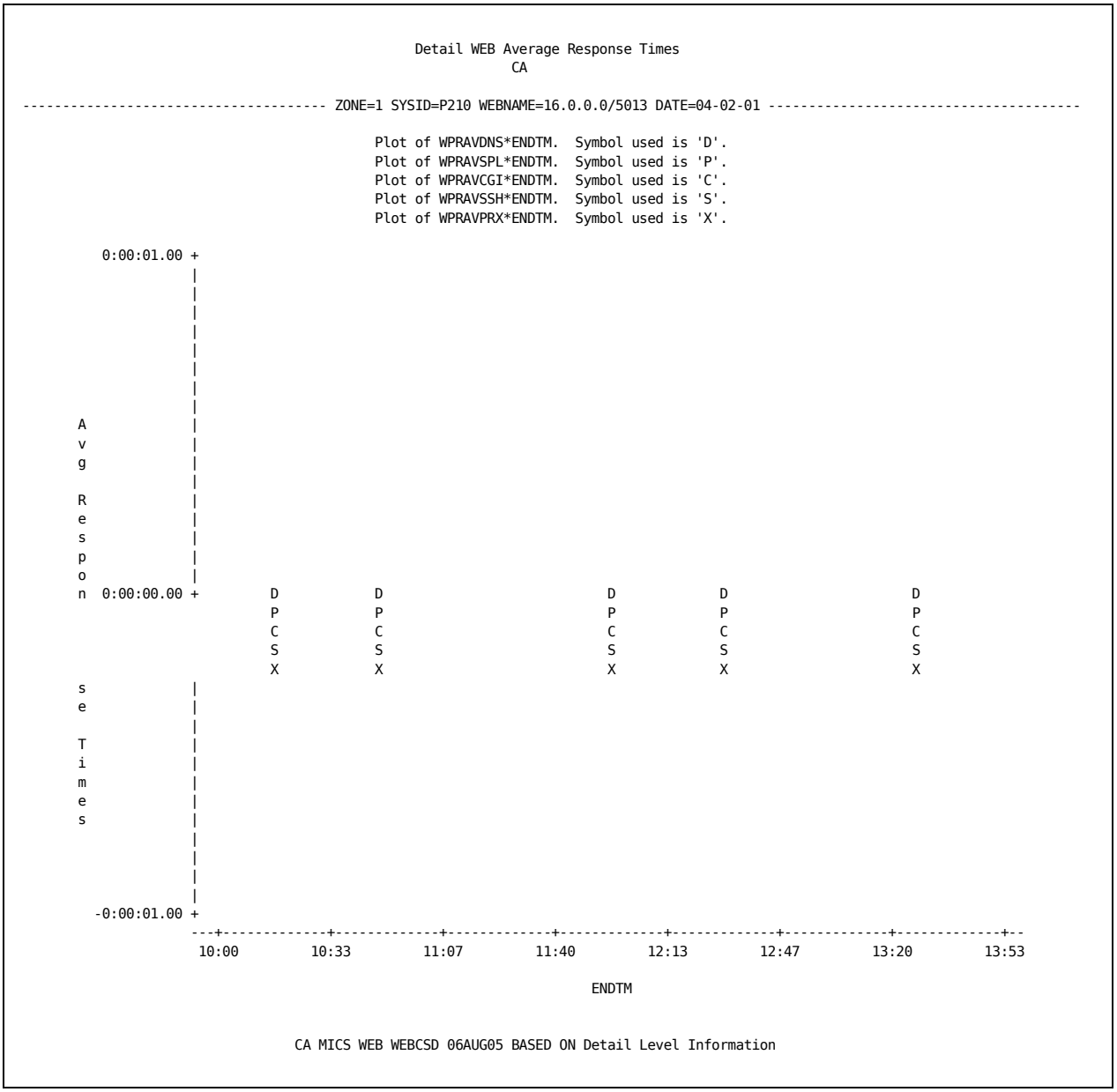
```
PARAMETER SPECIFICATIONS

_________________________

REQUIRED PARAMETERS:

The following parameters must be specified for the inquiry to
execute.

DATABASE.  DEFAULT: None
         Specify up to six database unit IDs (the IDs of the
         CA MICS database units from which files are read).

         If CA MICS files exist in only one database unit,
         this parameter is set automatically.

         To display a selection list of valid database unit
         IDs, specify ?  (question mark).

         Duplicates IDs are not allowed.

CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).
```

```
GENERATE A PLOT FOR EACH VALUE OF ZONE.  DEFAULT: Y

         Y - Yes generates reports for each ZONE.

         N - No does NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.

GENERATE COLOR GRAPHIC PLOTS.  DEFAULT: N
        Y - (Yes) generates color graphic plots.  SAS/GRAPH
            software is a prerequisite for color graphics.
        N - (No) Standard SAS plots and graphs will be
            produced (non-color, line formats).

GENERATE CSV EXTRACT FOR QUERY AND REPORTING.  DEFAULT: N

        Y - Yes generates an extract CSV file for
            subsequent use in CA MICS Query and Reporting.
            This option is only valid when CA MICS Query
            and Reporting is installed.  The elements
            available for CSV extract are WPRAVDNS,
            WPRAVSPL, WPRAVCGI, WPRABSSH, and WPRAVPRX.

        N - No does not generate a CSV file.
```

```
OPTIONAL PARAMETERS:

The following parameters need not be specified for the
inquiry to execute.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.

         The hour ranges should not overlap, but you can
         specify the higher range before the lower range.

         For example, a valid hour range specification is:

           18-23 09-17 02-04

         An input observation is selected if its hour value
         is greater than or equal to the starting hour and
         less than or equal to the ending hour of any
         specified hour range.
```

```
ZONE.  DEFAULT: None
         Specify up to six ZONE values.

         Each ZONE a 1 byte character field.

         Duplicates are not allowed.

         Using ZONEs you can restrict the inquiry analysis
         to only those ZONE values specified.

SYSID.  DEFAULT: None
         Specify up to six SYSIDs.

         Each SYSID a 1-4 byte character field.

         Duplicates are not allowed.

         Using SYSIDs you can restrict the inquiry analysis
         to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
         Specify up to six Web Short Names.

         Each Web Short Name is a 1-4 byte character field.

         Duplicates are not allowed.

         Using WEBSNAMEs you can restrict the inquiry
         analysis to only those WEBSNAME values specified.

WEB NAME (WEBNAME).  DEFAULT: None
         Specify up to four Web Names.

         Each Web Name is a 1-20 byte character field.

         Duplicates are not allowed.

         Using WEBNAMEs you can restrict the inquiry analysis
         to only those WEBNAME values specified.
```

```
WEB GROUP (WEBGROUP).  DEFAULT: None
        Specify up to four Web Groups.

        Each Web Group is a 1-8 byte character field.

        Duplicates are not allowed.

        Using WEBGROUPs you can restrict the inquiry
        analysis to only those WEBGROUP values specified.
```



3.1.2.5 WEB Method Utilization (WEBCSE)

The WEB Method Utilization Report uses measurements from the Access Method Summary File (WLGAMS) to display an overall view of page hits categorized by request methods over the date and/or time ranges you select. Each plot displays one separate request method based on the report summarization options you specify.

The following CA MICS Web Analyzer elements will be selected:

KEY ELEMENTS:

- SYSID - System Identifier
- WEBSNAME - Web Server Short Name
- WEBNAME - Web Server Long Name
- WEBGROUP - Web Group Name
- HOUR - Hour Of Day

DATA ELEMENTS:

- WLGREQME - Request Method (GET, POST, etc.)

Note: The label associated to the field DATE will vary based on timespan used.

To produce the report:

- Enter the CA MICS Workstation Facility (MWF)
- Choose option 2, CA MICS Information Center Facility (MICF)
- Choose option 2, Database Inquiries
- Select WEBCSE, Catalog Group: PERFORM
- Complete the MICF Execution panel
- Complete the specification panel illustrated below (Descriptions of the fields follow the panel)

The result displays on your terminal.

```
+-------------------------------------------------------------------------+
|---------------------------WEB Method Utilization--------------------|
| Command ===>                                                            |
|                                                                         |
| Composing Inquiry: WEBCSE - WEB Method Utilization                      |
|                                                                         |
| Database ===> ________________________________________                  |
| Cycle(s) ===> __ - __                                                   |
| Time-span ===> ____ (DAYS/WEEKS/MONTHS)                                 |
| Plot by hour if time-span is DAYS of WEEKS   ===> _ (Y/N)               |
| Generate a plot for each value of Zone       ===> _ (Y/N)               |
| Generate color graphic plots                 ===> _ (Y/N)               |
| Generate CSV Extract for Query and Reporting ===> _ (Y/N)               |
|                                                                         |
| Data Selection:                                                         |
|                     dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy       |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __       |
|   Hour         ===> __ - __    __ - __    __ - __                      |
|   Zone         ===> _ _ _ _ _ _ _ _ _                                   |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                       |
|   WEB Short Names(WEBSNAME) ===> ____ ____ ____ ____ ____ ____          |
|   WEB Name  ===> ____________________ ____________________              |
|                  ____________________ ____________________              |
|   WEB Group(WEBGROUP) ===> ________ ________ ________ ________          |
+-------------------------------------------------------------------------+
```

PARAMETER SPECIFICATIONS

REQUIRED PARAMETERS:

The following parameters require specification for the
inquiry to execute.

DATABASE. DEFAULT: None

Specify up to six database unit IDs (the IDs of the CA MICS database units from which files are read).

If CA MICS files exist in only one database unit, this parameter is set automatically.

To display a selection list of valid database unit IDs, specify ? (question mark).

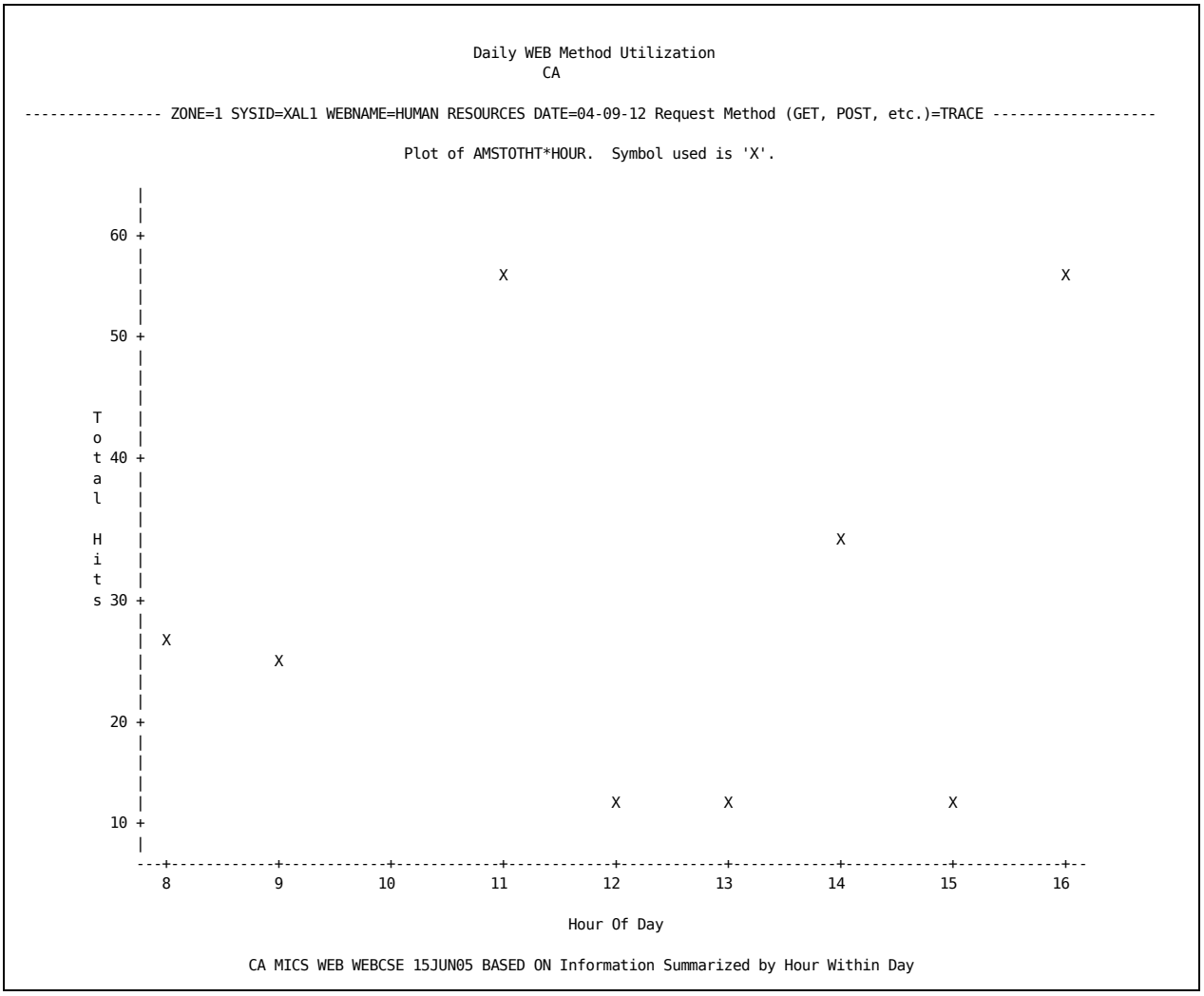
Duplicates IDs are not allowed.

```
CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).

TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)

GENERATE A PLOT FOR EACH VALUE OF ZONE.  DEFAULT: Y
         Y - (Yes) to generate reports for each ZONE.

         N - (No)  to NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.

GENERATE COLOR GRAPHIC PLOTS.  DEFAULT: N
        Y - (Yes) generates color graphic plots.  SAS/GRAPH
            software is a prerequisite for color graphics.
        N - (No) Standard SAS plots and graphs will be
            produced (non-color, line formats).
```

```
GENERATE CSV EXTRACT FOR QUERY AND REPORTING.  DEFAULT: N
        Y - (Yes) generates an extract CSV file for
            subsequent use in CA MICS Query and Reporting.
            This option is only valid when the CA MICS Query
            and Reporting is installed.  The elements
            available for CSV extract are  AMSREQME and
            AMSTOTHT.
        N - (No) CSV file not produced.


OPTIONAL PARAMETERS:

The following parameters do not require specification for the
inquiry to execute, they are optional.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.
```

```
          The hour ranges should not overlap, but you can
          specify the higher range before the lower range.

          For example, a valid hour range specification is:

            18-23 09-17 02-04

          An input observation is selected if its hour value
          is greater than or equal to the starting hour and
          less than or equal to the ending hour of any
          specified hour range.

ZONE.  DEFAULT: None
          Specify up to six ZONE values.

          Each ZONE a 1 byte character field.

          Duplicates are not allowed.

          Using ZONEs you can restrict the inquiry analysis
          to only those ZONE values specified.

SYSID.  DEFAULT: None
          Specify up to six SYSIDs.

          Each SYSID a 1-4 byte character field.

          Duplicates are not allowed.

          Using SYSIDs you can restrict the inquiry analysis
          to only those SYSID values specified.

WEB SHORT NAMES (WEBSNAME).  DEFAULT: None
          Specify up to six Web Short Names.

          Each Web Short Name is a 1-4 byte character field.

          Duplicates are not allowed.

          Using WEBSNAMEs you can restrict the inquiry
          analysis to only those WEBSNAME values specified.
```

```
WEB NAME (WEBNAME).  DEFAULT: None
         Specify up to four Web Names.

         Each Web Name is a 1-20 byte character field.

         Duplicates are not allowed.

         Using WEBNAMEs you can restrict the inquiry analysis
         to only those WEBNAME values specified.

WEB GROUP (WEBGROUP).  DEFAULT: None
         Specify up to four Web Groups.

         Each Web Group is a 1-8 byte character field.

         Duplicates are not allowed.

         Using WEBGROUPs you can restrict the inquiry
         analysis to only those WEBGROUP values specified.
```


Daily WEB Method Utilization
CA
---------- ZONE=1 SYSID=XAL1 WEBNAME=HUMAN RESOURCES DATE=04-09-12 Request Method (GET, POST, etc.)=TRACE ----------
Plot of AMSTOTHT*HOUR. Symbol used is 'X'.
Total Hits
60
50
40
30
20
10
8
9
10
11
12
13
14
15
16
Hour Of Day
CA MICS WEB WEBCSE 15JUN05 BASED ON Information Summarized by Hour Within Day

3.1.2.6 WAS Bytes Sent and Received (WEBCSF)

The WAS Bytes Sent and Received Report uses measurements from the WebSphere Server Activity (WEBSAF) file to contrast the total bytes sent compared with the total bytes received over the date and/or time intervals you request.

The following CA MICS Web Analyzer elements will be selected:

KEY ELEMENTS:

- HOUR - Hour Of Day
- ZONE - Time Zone
- SYSID - System Identifier
- WEBSRVNM - WAS Server Name
- WEBINAME - Web Server Instance Name

DATA ELEMENTS:

- SAFBYTES - Bytes Received by the Server
- SAFBYTER - Bytes Sent by the Server

Note: The label associated to the field DATE will vary based on timespan used.

To produce the report:

- Enter the CA MICS Workstation Facility (MWF)
- Choose option 2, CA MICS Information Center Facility (MICF)
- Choose option 2, Database Inquiries
- Select WEBCSF, Catalog Group: PERFORM

o Complete the MICF Execution panel

o Complete the specification panel illustrated below
(Descriptions of the fields follow the panel)

The result will be displayed on your terminal.

```
+------------------------------------------------------------------------+
|-----------------------  WAS Bytes Sent and Received ----------------|
| Command ===>                                                           |
|                                                                        |
| Composing Inquiry: WEBCSF - WAS Bytes Sent and Received                |
|                                                                        |
| Database ===> ________________________________________                 |
| Cycle(s) ===> __ - __                                                  |
| Time-span ===> ______ (DETAIL/DAYS/WEEKS/MONTHS)                       |
| Plot by hour if time-span is DAYS or WEEKS   ===> _ (Y/N)              |
| Generate a plot for each value of Zone       ===> _ (Y/N)              |
| Generate color graphic plots                 ===> _ (Y/N)              |
| Generate CSV Extract for Query and Reporting ===> _ (Y/N)              |
|                                                                        |
| Data Selection:                                                        |
|                      dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy     |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __     |
|   Hour         ===> __ - __    __ - __    __ - __                     |
|   Zone         ===> _ _ _ _ _ _ _ _ _ _                                |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                      |
|   WAS Server Names(WEBSRVNM) ===> ________ ________ ________ ________|
|   WAS Server Instance                                                  |
|              Name (WEBINAME) ===> ________ ________ ________ ________|
+------------------------------------------------------------------------+
```

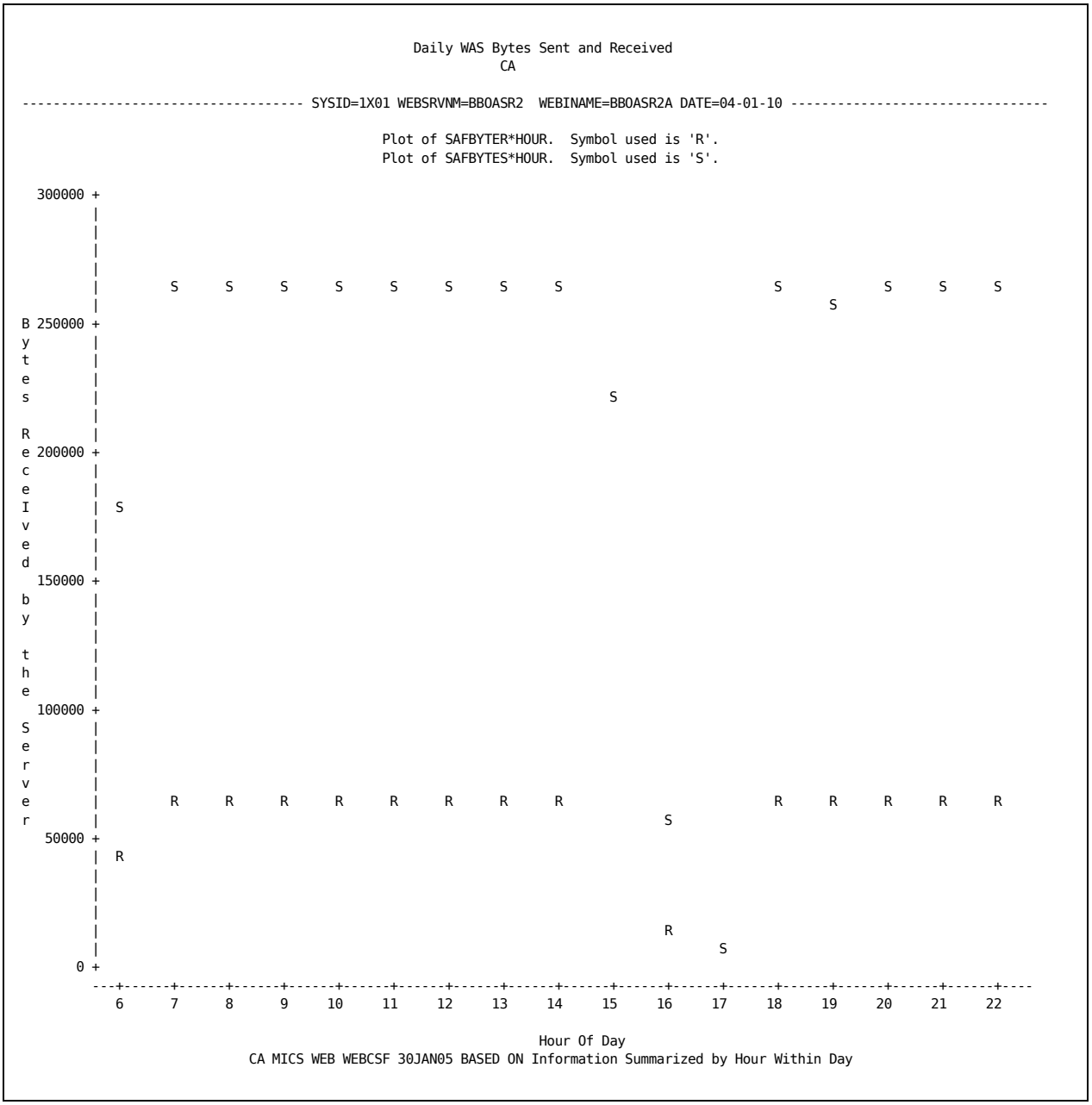
```
PARAMETER SPECIFICATIONS

_________________________

REQUIRED PARAMETERS:

The following parameters require specification for the
inquiry to execute.

DATABASE.  DEFAULT: None
         Specify up to six database unit IDs (the IDs of the
         CA MICS database units from which files are read).

         If CA MICS files exist in only one database unit,
         this parameter is set automatically.

         To display a selection list of valid database unit
         IDs, specify ?  (question mark).

         Duplicates IDs are not allowed.

CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).

TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)
```

```
PLOT BY HOUR.  DEFAULT: Y
         At the DAYS and WEEKS timespans, reports can be
         produced at an hourly, DAILY, or WEEKLY level.

         Y - (Yes) to Report by HOUR.
         The data is summarized and displayed by hour and
         separate reports are produced for each day or week.

         N - (No) to NOT report by HOUR.
         The data is summarized by day or week and a single
         report listing all days or weeks is produced.

GENERATE A PLOT FOR EACH VALUE OF ZONE.  DEFAULT: Y
         Y - (Yes) to generate reports for each ZONE.

         N - (No) to NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.

GENERATE COLOR GRAPHIC PLOTS.  DEFAULT: N
        Y - (Yes) generates color graphic plots.  SAS/GRAPH
            software is a prerequisite for color graphics.
        N - (No) Standard SAS plots and graphs will be
            produced (non-color, line formats).

GENERATE CSV EXTRACT FOR QUERY AND REPORTING.  DEFAULT: N
        Y - (Yes) generates an extract CSV file for
            subsequent use in CA MICS Query and Reporting.
            This option is only valid when CA MICS Query and
            Reporting is installed.  Elements available for
            CSV extract are SAFBYTES and SAFBYTER.
        N - (No) CSV file not produced.
```

```
OPTIONAL PARAMETERS:

The following parameters do not require specification for the
inquiry to execute, they are optional.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.
```

```
          The hour ranges should not overlap, but you can
          specify the higher range before the lower range.

          For example, a valid hour range specification is:

            18-23 09-17 02-04

          An input observation is selected if its hour value
          is greater than or equal to the starting hour and
          less than or equal to the ending hour of any
          specified hour range.

ZONE.  DEFAULT: None
          Specify up to six ZONE values.

          Each ZONE a 1 byte character field.

          Duplicates are not allowed.

          Using ZONEs you can restrict the inquiry analysis
          to only those ZONE values specified.

SYSID.  DEFAULT: None
          Specify up to six SYSIDs.

          Each SYSID a 1-4 byte character field.

          Duplicates are not allowed.

          Using SYSIDs you can restrict the inquiry analysis
          to only those SYSID values specified.

WAS SERVER NAME (WEBSRVNM).  DEFAULT: None
          Specify up to four WAS Server Names.

          Each Server Name is a 1-8 character field.

          Duplicates are not allowed.

          Using WEBSRVNM you can restrict the inquiry analysis
          to only those WEBSRVNM values specified.

WAS SERVER INSTANCE NAME (WEBINAME).  DEFAULT: None
          Specify up to four WAS Server Instance Names.

          Each Instance Name is a 1-8 character field.

          Duplicates are not allowed.
```

Using WEBINAMEs you can restrict the inquiry analysis to only those WEBINAME values specified.



3.1.2.7 WAS HTTP Sessions Statistics (WEBCSG)

The WAS HTTP Sessions Statistics uses data from the HTTP Session Manager Interval File (WEBHTP) to display the Active Sessions Count, Invalidated Sessions Count, Finalized Sessions Count, and Total Live Sessions Count number over the date and/or time ranges you select.

The following CA MICS Web Analyzer elements will be selected:

KEY ELEMENTS:

- HOUR - Hour Of Day
- ZONE - Time Zone
- SYSID - System Identifier
- WEBSRVNM - WAS Server Name
- WEBINAME - Web Server Instance Name

DATA ELEMENTS:

- HTPASESS - Active Sessions Count
- HTPCSESS - Created Sessions Count
- HTPFSESS - Finalized Sessions Count
- HTPISESS - Invalidated Sessions Count
- HTPLSESS - Total Live Sessions Count

Note: The label associated to the field DATE will vary based on timespan used.

To produce the report:

- Enter the CA MICS Workstation Facility (MWF)
- Choose option 2, CA MICS Information Center Facility (MICF)
- Choose option 2, Database Inquiries
- Select WEBCSG, Catalog Group: PERFORM
- Complete the MICF Execution panel
- Complete the specification panel illustrated below (Descriptions of the fields follow the panel)

```
The result will be displayed on your terminal.

+-------------------------------------------------------------------------+
|----------------------- WAS HTTP Sessions Statistics ----------------|
| Command ===>                                                            |
|                                                                         |
| Composing Inquiry: WEBCSG - WAS HTTP Sessions Statistics                |
|                                                                         |
| Database ===> ____________________________________________              |
| Cycle(s) ===> __ - __                                                   |
| Time-span ===> ______ (DETAIL/DAYS/WEEKS/MONTHS)                        |
| Plot by hour if time-span is DAYS or WEEKS   ===> _ (Y/N)               |
| Generate a plot for each value of Zone       ===> _ (Y/N)               |
| Generate color graphic plots                 ===> _ (Y/N)               |
| Generate CSV Extract for Query and Reporting ===> _ (Y/N)               |
|                                                                         |
| Data Selection:                                                         |
|                     dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy       |
|   Date Ranges  ===> __ ___ __ - __ ___ __   __ ___ __ - __ ___ __       |
|   Hour         ===> __ - __    __ - __    __ - __                      |
|   Zone         ===> _ _ _ _ _ _ _ _ _                                   |
|   Sysid        ===> ____ ____ ____ ____ ____ ____                       |
|   WAS Server Names(WEBSRVNM) ===> ________ ________ ________ ________|
|   WAS Server Instance                                                   |
|              Name (WEBINAME) ===> ________ ________ ________ ________|
+-------------------------------------------------------------------------+

PARAMETER SPECIFICATIONS
________________________

REQUIRED PARAMETERS:

The following parameters require specification for the
inquiry to execute.

DATABASE.  DEFAULT: None
         Specify up to six database unit IDs (the IDs of the
         CA MICS database units from which files are read).

         If CA MICS files exist in only one database unit,
         this parameter is set automatically.

         To display a selection list of valid database unit
         IDs, specify ?  (question mark).

         Duplicates IDs are not allowed.
```

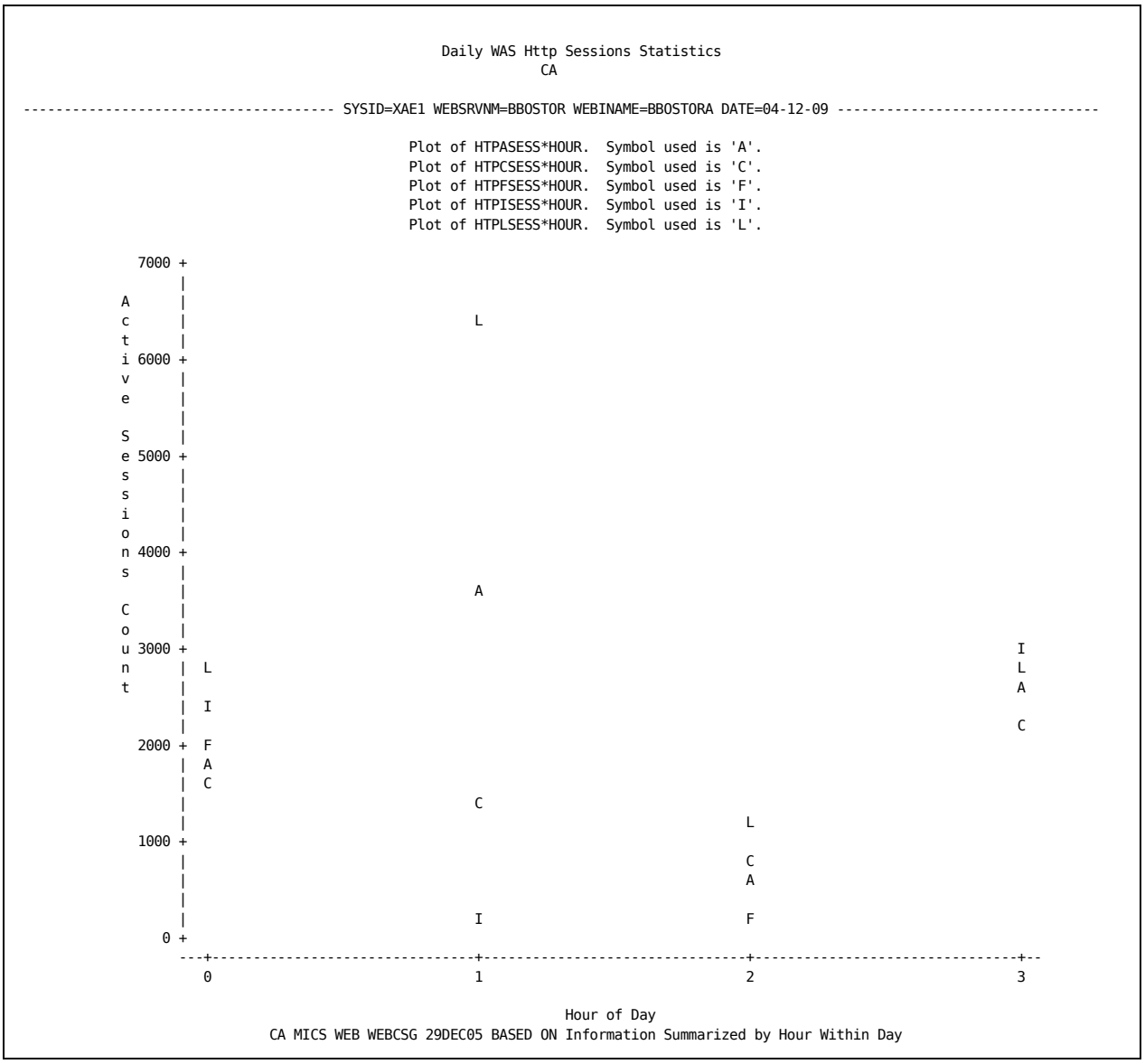
```
CYCLE(S).  DEFAULT: 01
        This is the cycle or cycles of the specified timespan
        of the CA MICS files that are used as input to the
        analysis.  Two data entry fields are provided so you
        can specify a range of cycles.  The fields can
        contain any numeric value between 0 and 99.  If you
        want only a single cycle (such as cycle 01), specify
        the selected cycle in the first entry field and leave
        the second field blank.  Leading zeros are not
        required and are inserted automatically if you supply
        a single-digit number (7 becomes 07).  If you specify
        a cycle range, you may specify it in either ascending
        or descending order (for example, 01-05 or 05-01).

TIMESPAN.  DEFAULT: varies by report
         The timespan of the CA MICS database from which the
         primary input files are selected.  Required input
         file(s) must be available at the specified timespan.
         (e.g., DETAIL, DAYS, WEEKS, MONTHS, YEARS)

PLOT BY HOUR.  DEFAULT: Y
         At the DAYS and WEEKS timespans, reports can be
         produced at an hourly, DAILY, or WEEKLY level.

         Y - (Yes) to Report by HOUR.
         The data is summarized and displayed by hour and
         separate reports are produced for each day or week.

         N - (No) to NOT report by HOUR.
         The data is summarized by day or week and a single
         report listing all days or weeks is produced.
```

```
GENERATE A PLOT FOR EACH VALUE OF ZONE.  DEFAULT: Y
         Y - (Yes) to generate reports for each ZONE.

         N - (No)  to NOT generate reports for each ZONE.

         The nature and resource requirements of processing
         often change from one zone to another (for example,
         prime shift online demands versus production batch
         requirements at non-prime shift periods).

         Zone reporting gives you the ability to display
         separate reports for each observed zone.

GENERATE COLOR GRAPHIC PLOTS.  DEFAULT: N
        Y - (Yes) generates color graphic plots.  SAS/GRAPH
            software is a prerequisite for color graphics.
        N - (No) Standard SAS plots and graphs will be
            produced (non-color, line formats).
```

```
GENERATE CSV EXTRACT FOR QUERY AND REPORTING.  DEFAULT: N
        Y - (Yes) generates an extract CSV file for
            subsequent use in CA MICS Query and Reporting.
            This option is only valid when CA MICS Query and
            Reporting is installed.  Elements available for
            CSV extract are SAFBYTES and SAFBYTER.
        N - (No) CSV file not produced.


OPTIONAL PARAMETERS:

The following parameters do not require specification for the
inquiry to execute, they are optional.

DATE RANGES.  DEFAULT: None
        Specify up to two date ranges in DD MMM YY format.
        The first value in each date range is the starting
        date and the last value is the ending date.  The
        ending date must be higher than the starting date.
        The date ranges should not overlap, but you can
        specify the higher range before the lower range.  For
        example, a valid date range specification is:

           date range 1:  01 JAN 01 - 31 DEC 01
           date range 2:  01 JAN 99 - 31 DEC 99

        An input observation is selected if its date value is
        greater than or equal to the starting date and is
        less than or equal to the ending date of any
        specified date range.

HOUR.  DEFAULT:  None
         Specify up to hour ranges of the form mm - nn, where
         mm and nn are two digit numbers between 00 and 23,
         inclusive.

           mm: range starting hour
           nn: range ending hour

         The ending hour must be higher than the starting
         hour.
```

```
              The hour ranges should not overlap, but you can
              specify the higher range before the lower range.

              For example, a valid hour range specification is:

                18-23 09-17 02-04

              An input observation is selected if its hour value
              is greater than or equal to the starting hour and
              less than or equal to the ending hour of any
              specified hour range.

ZONE.  DEFAULT: None
              Specify up to six ZONE values.

              Each ZONE a 1 byte character field.

              Duplicates are not allowed.

              Using ZONEs you can restrict the inquiry analysis
              to only those ZONE values specified.

SYSID.  DEFAULT: None
              Specify up to six SYSIDs.

              Each SYSID a 1-4 byte character field.

              Duplicates are not allowed.

              Using SYSIDs you can restrict the inquiry analysis
              to only those SYSID values specified.

WAS SERVER NAME (WEBSRVNM).  DEFAULT: None
              Specify up to four WAS Server Names.

              Each Server Name is a 1-8 character field.

              Duplicates are not allowed.

              Using WEBSRVNM you can restrict the inquiry analysis
              to only those WEBSRVNM values specified.

WAS SERVER INSTANCE NAME (WEBINAME).  DEFAULT: None
              Specify up to four WAS Server Instance Names.

              Each Instance Name is a 1-8 character field.

              Duplicates are not allowed.
```

Using WEBINAMEs you can restrict the inquiry
analysis to only those WEBINAME values specified.



Chapter 4: EXCEPTIONS

The CA MICS Web Analyzer does not distribute exceptions.

Chapter 5: FILES

The CA MICS Web Analyzer divides its data logically into the WebSphere Information Area (WEB) and the HTTP Log Files Information Area (WLG).

The WEB Information Area consists of the following files:

- HTTP Server Configuration File
- HTTP Server Performance File
- HTTP Server Performance Suspend File
- WAS Server Activity File
- WAS Server Interval File
- J2EE Method Interval File
- HTTP Session Manager Interval File
- WebContainer Servlet Interval File
- WebContainer Servlet Interval File
- JVM Heap Interval File
- Web Request Activity File
- Web CPU Usage Breakdown File
- Web User Data File

The WLG Information Area consists of the following files:

- HTTP Access Log File
- HTTP Agent Log File
- HTTP Referer Log File
- HTTP Error Messages Log File
- WLE Configuration File
- HTTP Access Method Summary File
- HTTP Access Path Summary File

The chart in Figure 5-2 describes the files supported by the CA MICS Web Analyzer and their data sources.

Figure 5-1 lists the product's files and the timespans that are activated for each file. The table identifies the individual database files and categorizes them by the information area to which they belong. For each file in an information area, the following data is provided:

XDWMYT - This entry defines the timespans in which the file is supported. A nonblank indicates that the file is supported. A period (.) indicates that the file is not supported. The timespans indicated by XDWMYT are as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA

File - The name used to access this file.

File Name - The descriptive label for the file.

Note: The timespans in which a file is supported are defined by each data center when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your data center.

```
+--------------------------------------+------------------+
| WEB                                  | Date Generated:  |
| Information Area                     |Thu, Feb 14, 2013 |
+--------------------------------------+------------------+
|                                                         |
| XDWMYT  File    File Name                               |
|                                                         |
| X.....  WEB_WP  HTTP SERVER PERFORMANCE SUSPEND FILE    |
| XDWM..  WEBHPI  JVM HEAP INTERVAL FILE                  |
| XDWMY.  WEBHTP  HTTP SESSION MANAGER INTERVAL FILE      |
| XDWMY.  WEBMIJ  J2EE METHOD INTERVAL FILE               |
| XDWMY.  WEBSAF  WAS SERVER ACTIVITY FILE                |
| XDWMY.  WEBSIF  WAS SERVER INTERVAL FILE                |
| XDWMY.  WEBSRV  WEBCONTAINER SERVLET INTERVAL FILE      |
| XDWMY.  WEBWAA  WEB ASYNCH ACTIVITY FILE                |
| X.....  WEBWCB  WEB CPU USAGE BREAKDOWN FILE            |
| X.....  WEBWCF  HTTP SERVER CONFIGURATION FILE          |
| XDWMY.  WEBWPR  HTTP SERVER PERFORMANCE FILE            |
| XDWMY.  WEBWRA  WEB REQUEST ACTIVITY FILE               |
| X.....  WEBWRS  WRA CPU DELTA SUSPEND FILE              |
| X.....  WEBWUD  WEB USER DATA FILE                      |
+--------------------------------------+------------------+
| WEB Log Files                        | Date Generated:  |
| Information Area                     |Thu, Feb 14, 2013 |
+--------------------------------------+------------------+
|                                                         |
| XDWMYT  File    File Name                               |
|                                                         |
| .DWM..  WLGAGL  HTTP AGENT LOG FILE                     |
| .DWMY.  WLGAMS  HTTP ACCESS METHOD SUMMARY FILE         |
| .DWMY.  WLGAPS  HTTP ACCESS PATH SUMMARY FILE           |
| X.....  WLGCFG  WLE CONFIGURATION FILE                  |
| XDWMY.  WLGERL  HTTP ERROR MESSAGES LOG FILE            |
| .DWM..  WLGRFL  HTTP REFERER LOG FILE                   |
| XDWMY.  WLGSAL  HTTP ACCESS LOG FILE                    |
|                                                         |
+---------------------------------------------------------+
```

Figure 5-1. Information Areas and Files

File	File Name	Description	Derivation
WEBWCF	HTTP Configuration	HTTP Configuration File	Type 103 (Subtype 1)
WEBWPR	WebSphere Performance	HTTP Performance File	Type 103 (Subtype 2)
WEB_WP	WebSphere Suspend	HTTP Performance Suspend File	Type 103 (Subtype 2)
WEBSAF	WebSphere Server Activity	WAS Server Activity File	Type 120 (Subtype 1)
WEBSIF	WebSphere Server Interval	WAS Server Interval File	Type 120 (Subtype 3)
WEBMIJ	J2EE Method Interval	J2EE Method Interval File	Type 120 (Subtype 1)
WEBHTP	HTTP Session Manager Interval	HTTP Session Manager Interval File	Type 120 (Subtype 1)
WEBSRV	WebContainer Servlet Interval	WebContainer Servlet Interval File	Type 120 (Subtype 1)
WEBHPI	Heap Interval	JVM Heap Interval File	Type 120 (Subtype 3)
WEBWRA	Web Request Activity	Web Request Activity File	Type 120 (Subtype 3)
WEBWCB	Web CPU Usage Breakdown	Web CPU Usage Breakdown File	Type 120 (Subtype 3)
WEBWUD	Web User Data	Web Usert Data File	Type 120 (Subtype 3)
WLGSAL	Server Access Log	HTTP Server Access Log File	WLE Utility Record (Subtype 1-4)
WLGAGL	Agent Log	HTTP Agent Log File	WLE Utility Record (Subtype 5)
WLGRFL	Referer Log	HTTP Referer Log File	WLE Utility Record (Subtype 6)
WLGERL	Error Log	HTTP Error Messages Log File	WLE Utility Record (Subtype 7)
WLGCFG	WLE Configuration	WLE Configuration File	WLE Utility Record (Subtype 9)
WLGAMS	Access Method Summary	HTTP Access Method Summary File	WLE Utility Record (Subtype 1)
WLGAPS	Access Path Summary	HTTP Access Path Summary File	WLE Utility Record (Subtype 1)

Figure 5-2. Supported Files and Their Data Sources

This chapter presents the CA MICS Web Analyzer data files as they are seen in the online database structure. The files may also be located offline in either audit or history archive files. The PIOM and the Database Structure and Content Guide discuss these other structures.

This chapter is divided into the following sections:

1 - Data Element Naming Conventions
2 - Web Information Area (WEB) Files
3 - Web Log Information Area (WLG) Files

This section contains the following topics:

5.1 Data Element Naming Conventions

CA MICS data elements follow naming conventions that depend on whether they are standard or common data elements. Standard data elements use the first three characters of their names to identify the file in which they are defined. The following charts list the three-character prefixes with which the standard data element names begin and the files in which they are contained.

WEB Information Area (WEB)

File Name	File	Names Begin With
HTTP Server Configuration File	WEBWCF	WCF
HTTP Server Performance File	WEBWPR	WPR
HTTP Server Performance Suspend File	WEB_WP	_WP *
WAS Server Activity File	WEBSAF	SAF
WAS Server Interval File	WEBSIF	SIF
J2EE Method Interval File	WEBMIJ	MIJ
HTTP Session Manager Interval File	WEBHTP	HTP
WebContainer Servlet Interval File	WEBSRV	SRV
JVM Heap Interval File	WEBHPI	HPI

WLG Files Information Area (WLG)

File Name	File	Names Begin With
HTTP Access Log File	WLGSAL	SAL
HTTP Agent Log File	WLGAGL	AGL
HTTP Referer Log File	WLGRFL	RFL
HTTP Error Messages Log File	WLGERL	ERL
WLE Configuration File	WLGCFG	CFG
HTTP Access Method Summary File	WLGAMS	AMS
HTTP Access Path Summary File	WLGAPS	APS

*Note: The "suspend" file (denoted with the asterisk) is used to provide continuity of data from one DAILY execution to the next. These files generally contain the same three letter prefix as the xxx file they are associated with.

Common data elements do not use a data element prefix. They have a common definition across database information areas or across files within an information area. You will find common data elements listed under Sequence/Summary Data Elements and Common Data Elements in the Data Elements List sections that follow.

5.2 WEB Information Area (WEB) Files

This section identifies each file in the WEB Information Area and defines its levels of summarization and data sequencing. Also, it presents the list of available data elements for each file.

The files maintained in the WEB Information Area include:

1 - HTTP Server Configuration File (WEBWCF)
2 - HTTP Server Performance File (WEBWPR)
3 - HTTP Server Performance Suspend File (WEB_WP)
4 - WAS Server Activity File (WEBSAF)
5 - WAS Server Interval File (WEBSIF)
6 - J2EE Method Interval File (WEBMIJ)
7 - HTTP Session Manager Interval File (WEBHTP)
8 - Servlet Interval File (WEBSRV)
9 - JVM Heap Interval File (WEBHPI)
10 - Web Request Activity File (WEBWRA)
11 - Web Request CPU Breakdown File (WEBWCB)
12 - Web Request User Data File (WEBWUD)
13 - Web Asynch Activity File (WEBWAA)

5.2.1 HTTP Server Configuration File (WEBWCF)

The HTTP Server Configuration (WEBWCF) File is derived from the SMF type 103 subtype 1 record. The information in the SMF configuration record data segment is taken from the HTTPD.CONF server configuration file. The configuration record is written each time the IBM WebSphere server is restarted and fully initialized.

The WEBWCF file exists as a single cycle only at the DETAIL timespan, where it contains one record for each SMF configuration record. The WEBWCF is known as a 'spin' file, unlike most other files there is only one DETAIL cycle (01).

```
This single file contains observations representing the last
'nn' SMF type 103 subtype 1 configuration records, where 'nn'
defaults to seven (7).  You can specify a different value for
'nn' using the "WCFLIMIT nn" parameter in
prefix.MICS.PARMS(WEBOPS).  For more information on the
WCFLIMIT parameter, refer to section 7.3.2 of this guide.

The following sections describe the file's organization, list
data elements maintained, and provide usage hints.

    1 - WEBWCF File Organization
    2 - WEBWCF Data Elements List
    3 - WEBWCF Usage Considerations
    4 - WEBWCF Retrieval Examples
```

5.2.1.1 WEBWCF File Organization

The table below (Figure 5-3) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the

CA MICS system at your installation.

Timespan	Level of Data Granularity
DETAIL	SYSID WEBIP WEBPORT YEAR MONTH DAY HOUR ENDTS
DAYS	N/A
WEEKS	N/A
MONTHS	N/A
YEARS	N/A
TABLES	N/A

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-3. WEBWCF Sort Sequence and Data Granularity

5.2.1.2 WEBWCF Elements List

```
_ins000e.html--share


GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *  Element    Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 X.....E  DAY      - Day of Month
 X.....E  ENDTS    - End Time Stamp
 X.....E  HOUR     - Hour of Day
 X.....E  MONTH    - Month of Year
 X.....E  SYSID    - System Identifier
 X.....E  WEBIP    - Web IP Address
 X.....E  WEBPORT  - Web Server Port Number
 X.....E  YEAR     - Year of Century

Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 X.....E  MICSVER  - CA MICS Version Number
 X.....E  STARTTS  - Start Time Stamp
 X.....E  WEBGROUP - Web Group Name
 X.....E  WEBNAME  - Web Server Name
 X.....E  WEBSNAME - Web Server Short Name
 X.....E  WEBSRVRT - Directory for Server Root
 X.....E  WEBSRVVR - Server Version
 X.....E  WEBSTYPE - Web Server Type
 X.....E  WEBSYSNM - System Name
 X.....E  WEEK     - Week of Year
 X.....E  ZONE     - Time Zone
```

```
Retained Data Elements

 X.....E  WCFACL   - ACL Settings
 X.....E  WCFCACFL - Cache Flag
 X.....E  WCFCACL1 - Cache Limit 1 (Kbytes)
 X.....E  WCFCACL2 - Cache Limit 2 (Kbytes)
 X.....E  WCFCALTO - Cache Lock Timeout
 X.....E  WCFCAMTM - Cache Time Margin
 X.....E  WCFCONFL - Cache Connect Flag
 X.....E  WCFDIRAC - Directory Access Flag
 X.....E  WCFDNSFL - DNS Lookup Flag
 X.....E  WCFGCFL  - Garbage Collection Flag
 X.....E  WCFGCINT - Garbage Collection Interval (long)
 X.....E  WCFGCMEM - Garbage Collection Memory Use
 X.....E  WCFGMTFL - GMT Flag
 X.....E  WCFIDLTO - Timeout for Idle Threads
 X.....E  WCFINTO  - Input Timeout
 X.....E  WCFKEXFL - Keep Expired Flag
 X.....E  WCFMETFL - Meta File Flag
 X.....E  WCFMNTRD - Min Number of Threads
 X.....E  WCFMODFL - Normal Mode Flag
 X.....E  WCFMXCBF - Max Size of Content Buffer
 X.....E  WCFMXFCA - Max File to Cache
 X.....E  WCFMXKCA - Max k Cache

 X.....E  WCFMXTRD - Max Number of Threads
 X.....E  WCFOUTTO - Output Timeout
 X.....E  WCFPXYFL - Proxy Flag
 X.....E  WCFSCRTO - Script Timeout
 X.....E  WCFSECPT - Security Port
 X.....E  WCFSECTP - Security Type
 X.....E  WCFSIHFL - Server Imbeds HTML Flag
```

5.2.1.3 WEBWCF Usage Considerations

As documented in the WebSphere Application Server for OS/390 HTTP Server Guide, the subtype 01 record is generated only at the server start-up time.

The WEBWCF file is only available at the DETAIL time-span. An SMF type 103 subtype 01 record is generated only at WebSphere server initialization, so there may be multiple records for a specific server for a particular day, if that server is restarted multiple times on that date. Conversely, if a server is up for a long duration without restarting, there will be long time-spans between WEBWCF observations for that server. This file contains the configuration information, defined in the HTTPD.CONF file, which does not lend itself to summarization at higher time-spans.

By default, the file contains the most recent seven observations for each web server. The observation count for web server is controlled with the "WCFLIMIT nn" parameter value specified in prefix.MICS.PARMS(WEBOPS). See section 7.2.3 in this guide for more information about the WCFLIMIT parameter.

The WEBWCF file contains the HTTP Server Configuration Directive values contained in the SMF type 103 subtype 1 record. Configuration directive values not contained in the WEBWCF file may be found in the WebSphere server's HTTPD.CONF file.

5.2.1.4 WEBWCF Retrieval Examples

This section presents a typical WEBWCF retrieval example.

Produce a report showing the web server working directory name and the end time stamp to see when and how many times the working directory name changed.

```
PROC PRINT DATA=&PWEBX..WEBWCF01;
  ID WEBIP WEBPORT;
  VAR ENDTS WEBSRVRT;
```

5.2.2 HTTP Server Performance File (WEBWPR)

The HTTP Server Performance (WEBWPR) File contains WebSphere HTTP server performance data. This file is derived from the SMF type 103 record (subtype 2). The performance data in the SMF performance record data segment is provided by the SNMP (Simple Network Management Protocol) MIB (Management Information Block). The data is accumulated continuously and written at intervals defined in the HTTP.CONF file by the SMFRecordingInterval directive. For more information on this statement, refer to section 6.2.1 of this guide.

The HTTP performance data, along with the HTTP Server Configuration data, help you to monitor the health, throughput, and activity of your Web Server.

The following sections describe the file organization, list data elements maintained, and provide usage hints.

1 - WEBWPR File Organization
2 - WEBWPR Data Elements List
3 - WEBWPR Usage Considerations
4 - WEBWPR Retrieval Examples

5.2.2.1 WEBWPR File Organization

The table below (Figure 5-4) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	DAY	HOUR	ENDTS		
DAYS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	DAY	HOUR			
WEEKS	SYSID	WEBIP	WEBPORT	YEAR	WEEK
	ZONE	HOUR			
MONTHS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	ZONE				
YEARS	SYSID	WEBIP	WEBPORT	YEAR	ZONE
TABLES	N/A				

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-4. WEBWPR Sort Sequence and Data Granularity

5.2.2.2 WEBWPR Elements List

```
_ins000e.html

GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *   Element   Description (LABEL)
 -------  --------   ---------------------------------------


Sequence/Summary Data Elements

 XD....E  DAY      - Day of Month
 XDW...E  HOUR     - Hour of Day
 XDWM..E  MONTH    - Month of Year
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBIP    - Web IP Address
 XDWMY.E  WEBPORT  - Web Server Port Number
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.E  DURATION - Recording Interval Time
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  INTERVLS - Number of Recording Intervals
 XDWMY.E  MICSVER  - CA MICS Version Number
 XDWMY.E  STARTTS  - Start Time Stamp
 XDWMY.E  WEBGROUP - Web Group Name
 XDWMY.E  WEBNAME  - Web Server Name
 XDWMY.E  WEBSNAME - Web Server Short Name
 XDWMY.E  WEBSRVVR - Server Version
 XDWMY.E  WEBSTYPE - Web Server Type
 XDWMY.E  WEBSYSNM - System Name

Retained Data Elements

 X.....E  WPRAVCGI - Avg CGI Response Time
 X.....E  WPRAVDNS - Avg DNS Lookup Response Time
 X.....E  WPRAVPRX - Avg Proxy Response Time
 X.....E  WPRAVSPL - Avg Service Plugins Response Time
 X.....E  WPRAVSSH - Avg SSL Handshake Response Time
 XDWMY.E  WPRVRRL  - Server Version/Release Number
```

```
Accumulated Data Elements

 XDWMY.E  WPRAPICT - GWAPI Requests
 XDWMY.E  WPRBCAC  - Bytes Read from Cache
 XDWMY.E  WPRBYTR  - Bytes Received
 XDWMY.E  WPRBYTS  - Bytes Sent
 XDWMY.E  WPRCACFC - Cached Files
 XDWMY.E  WPRCACHT - Cache Hits
 XDWMY.E  WPRCACRM - Bytes of Cache Ram in Use
 XDWMY.E  WPRCGICT - CGI Requests
 XDWMY.E  WPRCONN  - Connection Count
 XDWMY.E  WPRERRCT - Total Error Responses
 XDWMY.E  WPRGETCT - GET Requests
 XDWMY.E  WPRHEDCT - HEAD Requests
 XDWMY.E  WPRKBCAC - KBytes Read from Cache
 XDWMY.E  WPRKBR   - Kilobytes Received
 XDWMY.E  WPRKBS   - Kilobytes Sent
 XDWMY.E  WPRL200R - Error Level 200 Response Count
 XDWMY.E  WPRL300R - Error Level 300 Response Count
 XDWMY.E  WPRL400R - Error Level 400 Response Count
 XDWMY.E  WPRL500R - Error Level 500 Response Count
 XDWMY.E  WPRMBR   - Megabytes Received
 XDWMY.E  WPRMBS   - Megabytes Sent
 XDWMY.E  WPRREQDC - Requests Discarded
 XDWMY.E  WPRREQER - Request Errors Received
 XDWMY.E  WPRREQR  - Requests Received
 XDWMY.E  WPRRSPD  - Responses Discarded
 XDWMY.E  WPRRSPS  - Responses Sent
 XDWMY.E  WPRTIMOT - Timeouts
 XDWMY.E  WPRTRDCT - Threads Used
 XDWMY.E  WPRUBYTR - Unknown Type Bytes Received
 XDWMY.E  WPR200CT - Error200 Response Count
 XDWMY.E  WPR302CT - Error302 Response Count
 XDWMY.E  WPR401CT - Error401 Response Count
 XDWMY.E  WPR403CT - Error403 Response Count
 XDWMY.E  WPR404CT - Error404 Response Count
 XDWMY.E  WPR407CT - Error407 Response Count
 XDWMY.E  WPR500CT - Error500 Response Count
```

```
Minimum Data Elements

 XDWMY.E  WPRMNBRS - Min Bytes Received Rate
 XDWMY.E  WPRMNBSS - Min Bytes Sent Rate
 XDWMY.E  WPRMNBYR - Min Bytes Received
 XDWMY.E  WPRMNBYS - Min Bytes Sent
 XDWMY.E  WPRMNCGI - Min CGI Response Time
 XDWMY.E  WPRMNCON - Min Connection Count
 XDWMY.E  WPRMNDNS - Min DNS Lookup Response Time
 XDWMY.E  WPRMNPRX - Min Proxy Response Time
 XDWMY.E  WPRMNRR  - Min Request Received
 XDWMY.E  WPRMNRRS - Min Request Received Rate
 XDWMY.E  WPRMNRS  - Min Response Sent
 XDWMY.E  WPRMNRSS - Min Response Sent Rate
 XDWMY.E  WPRMNSPL - Min Service Plugins Response Time
 XDWMY.E  WPRMNSSH - Min SSL Handshake Response Time
 XDWMY.E  WPRMNTRU - Min Threads Used

Maximum Data Elements

 XDWMY.E  WPRMXBRS - Max Bytes Received Rate
 XDWMY.E  WPRMXBSS - Max Bytes Sent Rate
 XDWMY.E  WPRMXBYR - Max Bytes Received
 XDWMY.E  WPRMXBYS - Max Bytes Sent
 XDWMY.E  WPRMXCGI - Max CGI Response Time
 XDWMY.E  WPRMXCON - Max Connection Count
 XDWMY.E  WPRMXDNS - Max DNS Lookup Response Time
 XDWMY.E  WPRMXPRX - Max Proxy Response Time
 XDWMY.E  WPRMXRR  - Max Requests Received
 XDWMY.E  WPRMXRRS - Max Request Received Rate
 XDWMY.E  WPRMXRS  - Max Responses Sent
 XDWMY.E  WPRMXRSS - Max Response Sent Rate
 XDWMY.E  WPRMXSPL - Max Service Plugins Response Time
 XDWMY.E  WPRMXSSH - Max SSL Handshake Response Time
 XDWMY.E  WPRMXTRD - Max Threads Defined
 XDWMY.E  WPRMXTRU - Max Threads Used

Derived Data Elements

 XDWMY.E  WPRAVBCH - Average Bytes Per Cache Hit
 .DWMY.E  WPRAVCON - Avg No. of Connections
 XDWMY.E  WPRBYRPS - Average Bytes Received Rate
 XDWMY.E  WPRBYSPS - Average Bytes Sent Rate
 XDWMY.E  WPRCACPS - Average Cache Hit Rate
 XDWMY.E  WPRPCCAH - Percent CACHE Hits
 XDWMY.E  WPRRPSPS - Average Responses Sent Rate
 XDWMY.E  WPRRQRPS - Average Requests Received Rate
```

5.2.2.3 WEBWPR Usage Considerations

The WEBWPR file is derived from the SMF type 103 subtype 02 record. Provided that the logging queue is full, the records for subtype 2 are written at the defined SMF interval. A low activity level on the server results in:

- The logging queue filling more slowly
- The SMF records are written less frequently than the interval specified.

5.2.2.4 WEBWPR Retrieval Examples

This section presents a typical WEBWPR retrieval example.

1. Print yesterdays error counts for errors 302, 401, 403, 404, 407, and 500 by hour for each server.

```
PROC SORT DATA=&PWEBD..WEBWPR01 OUT=DAYSWPR;
 BY WEBNAME HOUR;

PROC PRINT DATA=DAYSWPR LABEL;
 BY WEBNAME HOUR;
 ID WEBNAME HOUR;
 VAR WPR302CT WPR401CT WPR403CT WPR404CT WPR407CT WPR500CT;
RUN;
```

2. Produce a report showing hourly hits and bytes sent for each server.

```
PROC SORT DATA=&PWEBD..WEBWPR01 OUT=DAYSWPR;
 BY WEBNAME HOUR;

PROC PRINT DATA=DAYSWPR LABEL;
 BY WEBNAME HOUR;
 ID WEBNAME HOUR;
 VAR WPRREQR WPRBYTS;
RUN;
```

5.2.3 HTTP Server Performance Suspend File (WEB_WP)

FILENAME = WEB_WP01

The WEB_WP Suspend File maintains information from the last SMF type 103 subtype 2 interval record for each WebSphere server. Because SMF type 103 subtype 2 interval records contain cummulative, rather than true interval, field values, CA MICS must compare the previous interval record values with current interval record values in order to compute true interval metrics.

The WEB_WP suspend file records are used to maintain continuity between successive executions of the CA MICS Web Analyzer daily update runs. This allows CA MICS to always provide true interval metrics in each WebSphere Performance (WEBWPR) file record.

The WEB_WP File is a permanent file of the CA MICS Web Analyzer. It is strictly for internal use and exists only at the DETAIL timespan.

5.2.4 WAS Server Activity File (WEBSAF)

The WAS Server Activity (WEBSAF) file is derived from the SMF type 120, subtype 1 records. The server activity record is written each time an activity occurs in J2EE (Java II Enterprise Edition) servant (server region). It contains information about the communication between the client and the servant. There can be one or more communication sections in each server activity record. The values for bytes received and bytes sent represent total values summed for all communication sections before an observation is written to the file.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WEBSAF File Organization
2 - WEBSAF Data Elements List
3 - WEBSAF Usage Considerations
4 - WEBSAF Retrieval Examples

5.2.4.1 WEBSAF File Organization

The table below (Figure 5-5) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBSRVNM
	WEBINAME	YEAR	MONTH	DAY	HOUR
	ENDTS				
DAYS	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBSRVNM
	WEBINAME	YEAR	MONTH	DAY	HOUR
WEEKS	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBSRVNM
	WEBINAME	YEAR	WEEK	ZONE	HOUR
MONTHS	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBSRVNM
	WEBINAME	YEAR	MONTH	ZONE	
YEARS	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBSRVNM
	WEBINAME	YEAR	ZONE		
TABLES	N/A				

Generation Date: Tue, May 12, 2009

```
NOTE:  This file was generated with ESSENTIAL=ALL
       option in effect.  All data elements defined in
       the file are generated.
NOTE:  This file was generated with DERIVED=DEFault
       option in effect.  Whether data elements are
       kept on the file on auxiliary storage or not
       is controlled by the complex definition of the
       DERIVED option.
```

Figure 5-5. WEBSAF Sort Sequence and Data Granularity

5.2.4.2 WEBSAF Elements List

```
_ins000e.html
GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *   Element   Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 XD....E  DAY      - Day of Month
 XDW...E  HOUR     - Hour of Day
 XDWM..E  MONTH    - Month of Year
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBACT1  - DIVISION
 XDWMY.E  WEBACT2  - DEPARTMENT
 XDWMY.E  WEBACT3  - SERVER APPLICATION
 XDWMY.E  WEBINAME - WebSphere Server Instance Name
 XDWMY.E  WEBSRVNM - WebSphere Server Name
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone
```

```
Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 XDWMY.E  ENDTS    - End Time Stamp
 X.....E  MICSVER  - CA MICS Version Number
 XDWMY.E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp
 XDWMY.E  WEBCELL  - WebSphere Cell Name
 X.....E  WEBIDENT - Identity of the Activity
 XDWMY.E  WEBNODE  - WebSphere Node Name
 XDWMY.E  WEBSHOST - WebSphere Host Name

Retained Data Elements

 X.....E  SAFATYPE - Type of Activity
 XDWMY.E  SAFCRED  - User Credentials Activity Ran Under
 XDWMY.E  SAFREGN1 - Server Instance Region 1
 XDWMY.E  SAFREGN2 - Server Instance Region 2
 XDWMY.E  SAFREGN3 - Server Instance Region 3
 XDWMY.E  SAFREGN4 - Server Instance Region 4
 XDWMY.E  SAFREGN5 - Server Instance Region 5
 X.....E  SAFSADDR - Communication Session Address
 X.....E  SAFSHNDL - Communication Session Handle
 X.....E  SAFSOPTM - Communication Session Optimization
 X.....E  SAFWASTY - WebSphere Server Type
 X.....E  SAFWLMET - WLM Enclave Token

Accumulated Data Elements

 XDWMY.E  SAFBYTER - Bytes Received by the Server
 XDWMY.E  SAFBYTES - Bytes Sent by the Server
 XDWMY.E  SAFCMCNT - Communication Session Count
 XDWMY.E  SAFCOST  - Processing Charges
 XDWMY.E  SAFENCTM - WLM Enclave: Total CPU Time
 XDWMY.E  SAFEXCTM - Server Activity Execution Time
 XDWMY.E  SAFGTCNT - Global Transactions in Region Count
 XDWMY.E  SAFIMCNT - Input Methods Count
 XDWMY.E  SAFLTCNT - Local Transactions in Region Count
 XDWMY.E  SAFRGCNT - Server Regions for this Activity Count
```

```
Derived Data Elements

 XDWMY.E  SAFAVBYR - Avg Bytes Received
 XDWMY.E  SAFAVBYS - Avg Bytes Sent
 XDWMY.E  SAFAVKBR - Avg Bytes Received (Kbytes)
 XDWMY.E  SAFAVKBS - Avg Bytes Sent (Kbytes)
 XDWMY.E  SAFAVMBR - Avg Bytes Received (Mbytes)
 XDWMY.E  SAFAVMBS - Avg Bytes Sent (Mbytes)
 XDWMY.E  SAFBYTET - Total Bytes Transferred
 XDWMY.E  SAFBYTTR - Bytes Transferred Rate
 XDWMY.E  SAFKBYTR - Bytes Received (Kbytes)
 XDWMY.E  SAFKBYTS - Bytes Sent (Kbytes)
 XDWMY.E  SAFKBYTT - Total Bytes Transferred (Kilobytes)
 XDWMY.E  SAFMBYTR - Bytes Received (Mbytes)
 XDWMY.E  SAFMBYTS - Bytes Sent (Mbytes)
 XDWMY.E  SAFMBYTT - Total Bytes Transferred (Megabytes)
 XDWMY.E  SAFPCBYR - Percent Bytes Received
 XDWMY.E  SAFPCBYS - Percent Bytes Sent
 XDWMY.E  SAFPCGT  - Percent Global Transaction
 XDWMY.E  SAFPCLT  - Percent Local Transaction
 XDWMY.E  SAFTTCNT - Total Transaction Count
```

5.2.4.3 WEBSAF Usage Considerations

The WEBSAF file is derived from SMF type 120 subtype 01 records. Each record contains a server record and one or more communication records. Each observation in this file contains complete detail for every communication record. Multiple communication records in one subtype 01 record will result in multiple observations in this file. When interpreting elements, SAFIMCNT, SAFGTCNT, SAFLTCNT, and SAFEXCTM, keep in mind that these values are repeating for different communication handles. These elements contain values that are aggregate of the work that ran inside the server instance.

5.2.4.4 WEBSAF Retrieval Examples

This section presents a typical WEBSAF retrieval example.

Produce a report showing the web server working directory name and the end time stamp to see when and how many times the working directory name changed.

```
PROC PRINT DATA=&PWEBX..WEBSAF01;
  ID WEBSRVNM WEBINAME;
  VAR SAFIMCNT SAFGTCNT SAFLTCNT SAFBYTER SAFBYTES;
```

5.2.5 WAS Server Interval File (WEBSIF)

The WAS Server Interval File (WEBSIF) is derived from the SMF type 120 subtype 3 records. The server interval record is written at the installation-specified interval. The purpose of this record is to record the activity that is running inside a WebSphere Application Server. It contains aggregate information about each activity that ran inside the server instance during the interval.

If a server has more than one server instance, then a record for each server instance is written and the records must be merged to get a complete picture of the work that ran inside the server.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WEBSIF File Organization
2 - WEBSIF Data Elements List
3 - WEBSIF Usage Considerations
4 - WEBSIF Retrieval Examples

5.2.5.1 WEBSIF File Organization

The table below (Figure 5-6) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	SYSID	WEBSRVNM	WEBINAME	YEAR	MONTH
	DAY	HOUR	ENDTS		
DAYS	SYSID	WEBSRVNM	WEBINAME	YEAR	MONTH
	DAY	HOUR			
WEEKS	SYSID	WEBSRVNM	WEBINAME	YEAR	WEEK
	ZONE	HOUR			
MONTHS	SYSID	WEBSRVNM	WEBINAME	YEAR	MONTH
	ZONE				
YEARS	SYSID	WEBSRVNM	WEBINAME	YEAR	ZONE
TABLES	N/A				

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-6. WEBSIF Sort Sequence and Data Granularity

5.2.5.2 WEBSIF Elements List

```
_ins000e.html
GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *   Element    Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 XDW...E  HOUR     - Hour of Day
 XDWM..E  MONTH    - Month of Year
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBINAME - WebSphere Server Instance Name
 XDWMY.E  WEBSRVNM - WebSphere Server Name
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone

Common Data Elements

 XDW...E  DAY      - Day of Month
 X.....E  DAYNAME  - Name of Day of Week
 XDWMY.E  DURATION - Recording Interval Time
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  INTERVLS - Number of Recording Intervals
 X.....E  MICSVER  - CA MICS Version Number
 XDWMY.E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp
 XDWMY.E  WEBCELL  - WebSphere Cell Name
 XDWMY.E  WEBNODE  - WebSphere Node Name
 XDWMY.E  WEBSHOST - WebSphere Host Name
```

```
Retained Data Elements

 X.....E  SIFWASTY - WebSphere Server Type

Accumulated Data Elements

 XDWMY.E  SIFABYTR - Bytes Rcvd From All Attached Clients
 XDWMY.E  SIFABYTS - Bytes Sent to All Attached Clients
 XDWMY.E  SIFCSACT - Communications Session Active Count
 XDWMY.E  SIFCSEXS - Communications Session Existed Count
 XDWMY.E  SIFENCTM - WLM Enclave: Total CPU Time
 XDWMY.E  SIFGTCNT - Global Transaction Count
 XDWMY.E  SIFHBYTR - Bytes Received From HTTP Clients
 XDWMY.E  SIFHBYTS - Bytes Sent to HTTP Clients
 XDWMY.E  SIFHSAA  - HTTP Comm. Sessions Active Count
 XDWMY.E  SIFHSEXS - HTTP Session Existed Count
 XDWMY.E  SIFLBYTR - Bytes Rcvd From Local Attached Clients
 XDWMY.E  SIFLBYTS - Bytes Sent to Local Attached Clients
 XDWMY.E  SIFLCSAA - Local Comm. Session Active Count
 XDWMY.E  SIFLCSEX - Local Comm. Session Existed Count
 XDWMY.E  SIFLTCNT - Local Transaction Count
 XDWMY.E  SIFRBYTR - Bytes Rcvd From Remote Attached Clients
 XDWMY.E  SIFRBYTS - Bytes Sent to Remote Attached Clients
 XDWMY.E  SIFRCSAA - Remote Comm. Session Active Count
 XDWMY.E  SIFRCSEX - Remote Comm. Session Existed Count

Derived Data Elements

 XDWMY.E  SIFBYTET - Tot Bytes Transferred - All Clients
 XDWMY.E  SIFBYTTR - Bytes Transferred Rate
 XDWMY.E  SIFKBYTT - Tot Bytes Transferred - All (Kbytes)
 XDWMY.E  SIFMBYTT - Tot Bytes Transferred - All (Mbytes)
 XDWMY.E  SIFPCGTC - Percent Global Transaction
 XDWMY.E  SIFPCLBR - Percent Bytes Rcvd from Local Clients
 XDWMY.E  SIFPCLBS - Percent Bytes Sent to Local Clients
 XDWMY.E  SIFPCLSA - Percent Local Comm. Session Active
 XDWMY.E  SIFPCLSE - Percent Local Comm. Session Existed
 XDWMY.E  SIFPCLTC - Percent Local Transaction
 XDWMY.E  SIFPCRBR - Percent Bytes Rcvd from Remote Clients
 XDWMY.E  SIFPCRBS - Percent Bytes Sent to Remote Clients
 XDWMY.E  SIFPCRSA - Percent Remote Comm. Session Active
 XDWMY.E  SIFPCRSE - Percent Remote Comm. Session Existed
 XDWMY.E  SIFTTCNT - Total Transaction Count
```

5.2.5.3 WEBSIF Usage Considerations

This WEBSIF file is derived from SMF type 120 subtype 03 records. Each record contains activity that is running inside the WebSphere for z/OS application server. Each observation in this file is produced from one subtype 03 record. The observation represents data created at regular intervals and contains information that is an aggregate of the work that ran inside the server instance during the interval.

A single observation is present for each server instance that has interval recording active during the interval. If a server has multiple server instances, then an observation for each server instance will be present in the file. The observations from these multiple instances must be merged to get a complete view of the work that ran inside the server.

5.2.5.4 WEBSIF Retrieval Examples

This section presents a typical WEBSIF retrieval example.

Produce a report showing the web server and its instances and the count of local and global transactions along with bytes sent and bytes received by all attached clients.

```
PROC PRINT DATA=&PWEBX..WEBSIF01;
  ID WEBSRVNM WEBINAME;
  VAR SIFGTCNT SIFLTCNT SIFABYTR SIFABYTS;
```

5.2.6 J2EE Method Interval File (WEBMIJ)

The J2EE Method Interval File (WEBMIJ) is derived from the SMF type 120 subtype 6 records. The J2EE Container Interval record is written at the installation-specified interval. It contains aggregate information about each activity that ran inside a Java (J2EE) container during the interval.

If a server has more than one server instance, then a record for the container will exist for each server instance. To get a complete view of the work running in the Java container during the interval, you must merge records from all server instances.

This file is populated with the SAS releases that support $UCS2Xw informat. No observation will be saved when running with an unsupported SAS release.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WEBMIJ File Organization
2 - WEBMIJ Data Elements List
3 - WEBMIJ Usage Considerations
4 - WEBMIJ Retrieval Examples

5.2.6.1 WEBMIJ File Organization

The table below (Figure 5-7) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	SYSID	WEBSRVNM	WEBINAME	MIJCONT	MIJBEAN
	MIJBTYPE	MIJMETH	YEAR	MONTH	DAY
	HOUR	ENDTS			
DAYS	SYSID	WEBSRVNM	WEBINAME	MIJCONT	MIJBEAN
	MIJBTYPE	MIJMETH	YEAR	MONTH	DAY
	HOUR				
WEEKS	SYSID	WEBSRVNM	WEBINAME	MIJCONT	MIJBEAN
	MIJBTYPE	MIJMETH	YEAR	WEEK	ZONE
	HOUR				
MONTHS	SYSID	WEBSRVNM	WEBINAME	MIJCONT	MIJBEAN
	MIJBTYPE	MIJMETH	YEAR	MONTH	ZONE
YEARS	SYSID	WEBSRVNM	WEBINAME	MIJCONT	MIJBEAN
	MIJBTYPE	MIJMETH	YEAR	ZONE	
TABLES	N/A				

Generation Date: Tue, May 12, 2009

```
NOTE:  This file was generated with ESSENTIAL=ALL
       option in effect.  All data elements defined in
       the file are generated.
NOTE:  This file was generated with DERIVED=DEFault
       option in effect.  Whether data elements are
       kept on the file on auxiliary storage or not
       is controlled by the complex definition of the
       DERIVED option.
```

Figure 5-7. WEBMIJ Sort Sequence and Data Granularity

5.2.6.2 WEBMIJ Elements List

```
_ins000e.html
GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *   Element    Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 XDW...E  HOUR     - Hour of Day
 XDWMY.E  MIJBEAN  - Bean Name
 XDWMY.E  MIJBTYPE - Bean Type
 XDWMY.E  MIJCONT  - Container Name
 XDWMY.E  MIJMETH  - Method Name
 XDWM..E  MONTH    - Month of Year
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBINAME - WebSphere Server Instance Name
 XDWMY.E  WEBSRVNM - WebSphere Server Name
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone
```

```
Common Data Elements

 XDW...E  DAY      - Day of Month
 X.....E  DAYNAME  - Name of Day of Week
 XDWMY.E  DURATION - Recording Interval Time
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  INTERVLS - Number of Recording Intervals
 X.....E  MICSVER  - CA MICS Version Number
 X.....E  STARTTS  - Start Time Stamp
 XDWMY.E  WEBCELL  - WebSphere Cell Name
 XDWMY.E  WEBNODE  - WebSphere Node Name
 XDWMY.E  WEBSHOST - WebSphere Host Name

Retained Data Elements

 X.....E  MIJBMTP  - Bean Method Transaction Policy
 X.....E  MIJBRENT - Bean Reentrance Policy
 X.....E  MIJEJBRA - ejbRoles Associated With Method

Accumulated Data Elements

 XDWMY.E  MIJEJBAI - ejbActivate: Invocation Count
 XDWMY.E  MIJEJBLI - ejbLoad: Invocation Count
 XDWMY.E  MIJEJBPI - ejbPassive: Invocation Count
 XDWMY.E  MIJEJBSI - ejbStore: Invocation Count
 XDWMY.E  MIJMICNT - Times Method Invoked in Activity
 XDWMY.E  MIJTOCPU - Total CPU Time
 XDWMY.E  MIJTOJBA - ejbActivate: Total Execution Time
 XDWMY.E  MIJTOJBL - ejbLoad: Total Execution Time
 XDWMY.E  MIJTOJBP - ejbPassive: Total Execution Time
 XDWMY.E  MIJTOJBS - ejbStore: Total Execution Time
 XDWMY.E  MIJTORTM - Total Response Time

Minimum Data Elements

 XDWMY.E  MIJMNCPU - Min CPU Time

Maximum Data Elements

 XDWMY.E  MIJMXCPU - Max CPU Time
 XDWMY.E  MIJMXJBA - ejbActivate: Max Execution Time
 XDWMY.E  MIJMXJBL - ejbLoad: Max Execution Time
 XDWMY.E  MIJMXJBP - ejbPassive: Max Execution Time
 XDWMY.E  MIJMXJBS - ejbStore: Max Execution Time
 XDWMY.E  MIJMXRTM - Max Response Time
```

```
Derived Data Elements

 XDWMY.E  MIJAVCPU - Avg CPU Time
 XDWMY.E  MIJAVJBA - ejbActivate: Avg Execution Time
 XDWMY.E  MIJAVJBL - ejbLoad: Avg Execution Time
 XDWMY.E  MIJAVJBP - ejbPassive: Avg Execution Time
 XDWMY.E  MIJAVJBS - ejbStore: Avg Execution Time
 XDWMY.E  MIJAVRTM - Avg Response Time
```

5.2.6.3 WEBMIJ Usage Considerations

The WEBMIJ file is derived from SMF type 120 subtype 6 records. Each record contains activity that is running inside the WebSphere for z/OS application server's J2EE container. Multiple observations in this file are produced from one subtype 6 record. There can be one or more observations within a bean, within a container, which contain information that is an aggregate of the work that ran inside the server instance during the interval.

Many observations may be present for each server instance that has interval recording active during the interval. If a server has multiple server instances, then many observations for each server instance may be present in the file. The observations from these multiple instances must be merged to obtain a complete view of the work that ran inside the server.

5.2.6.4 WEBMIJ Retrieval Examples

This section presents a typical WEBMIJ retrieval example.

Produce a report showing the web server, instances, containers, beans, and method with maximum response time exceeding 100 milliseconds.

```
PROC PRINT DATA=&PWEBX..WEBMIJ01;
  ID WEBSRVNM WEBINAME MIJCONT MIJBEAN;
  VAR MIJMXRTM;
  WHERE MIJMXRTM GT 100;
```

5.2.7 HTTP Session Manager Interval File (WEBHTP)

The HTTP Session Manager Interval File (WEBHTP) is derived from the SMF type 120 subtype 8 records. The Web container Interval record that also contains HTTP Session Manager information is written at the installation-specified interval. It contains information about HTTP sessions within one Web container.

This file is populated with the SAS releases that support $UCS2Xw informat. No observation will be saved when running with an unsupported SAS release.

The following sections describe the file's organization, list data elements maintained, and provide usage hints and examples.

1 - WEBHTP File Organization
2 - WEBHTP Data Elements List
3 - WEBHTP Usage Considerations
4 - WEBHTP Retrieval Examples

5.2.7.1 WEBHTP File Organization

The table below (Figure 5-8) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity
DETAIL	SYSID WEBSRVNM WEBINAME YEAR MONTH DAY HOUR ENDTS
DAYS	SYSID WEBSRVNM WEBINAME YEAR MONTH DAY HOUR
WEEKS	SYSID WEBSRVNM WEBINAME YEAR WEEK ZONE HOUR
MONTHS	SYSID WEBSRVNM WEBINAME YEAR MONTH ZONE
YEARS	SYSID WEBSRVNM WEBINAME YEAR ZONE
TABLES	N/A

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-8. WEBHTP Sort Sequence and Data Granularity

5.2.7.2 WEBHTP Elements List

```
_ins000e.html

GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *  Element    Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 XD....E  DAY      - Day of Month
 XDW...E  HOUR     - Hour of Day
 XDWM..E  MONTH    - Month of Year
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBINAME - WebSphere Server Instance Name
 XDWMY.E  WEBSRVNM - WebSphere Server Name
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.E  DURATION - Recording Interval Time
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  INTERVLS - Number of Recording Intervals
 X.....E  MICSVER  - CA MICS Version Number
 XDWMY.E  STARTTS  - Start Time Stamp
 XDWMY.E  WEBCELL  - WebSphere Cell Name
 XDWMY.E  WEBNODE  - WebSphere Node Name
 XDWMY.E  WEBSHOST - WebSphere Host Name

Accumulated Data Elements

 XDWMY.E  HTPASESS - Active Sessions Count
 XDWMY.E  HTPCSESS - Created Sessions Count
 XDWMY.E  HTPFSESS - Finalized Sessions Count
 XDWMY.E  HTPISESS - Invalidated Sessions Count
 XDWMY.E  HTPLSESS - Total Live Sessions Count
 XDWMY.E  HTPTOILT - Total Lifetime of Invalidated Sessions
 XDWMY.E  HTPTOIRT - Total Invalidation Required Time
```

```
Minimum Data Elements

 XDWMY.E  HTPMNAS  - Min Active Sessions Count
 XDWMY.E  HTPMNLS  - Min Live Sessions Count

Maximum Data Elements

 XDWMY.E  HTPMXAS  - Max Active Sessions Count
 XDWMY.E  HTPMXLS  - Max Live Sessions Count

Derived Data Elements

 XDWMY.E  HTPAVILT - Avg Lifetime of Invalidated Sessions
 XDWMY.E  HTPAVIRT - Avg Invalidation Required Time
```

5.2.7.3 WEBHTP Usage Considerations

```
The WEBHTP file is derived from SMF type 120 subtype 8
records.  Each record contains activity that is running
inside the WebSphere for z/OS application server's
WebContainer.  One observation is produced from one subtype 8
record.  There is only one (or no) HttpSessionManager
interval section in one physical record, which contains
information that is an aggregate of the work that ran inside
the server instance during the interval.  It provides various
statistics about HTTP sessions within a WebContainer.
```

5.2.7.4 WEBHTP Retrieval Examples

```
This section presents a typical WEBHTP retrieval example.

   Produce a report showing the observations where more than
   10 Http sessions were invalidated.

   PROC PRINT DATA=&PWEBX..WEBHTP01;
     ID WEBCELL WEBNODE WEBSRVNM WEBINAME;
     VAR HTPISESS;
     WHERE HTPISESS GT 10;
```

5.2.8 WebContainer Servlet Interval File (WEBSRV)

The WebContainer Servlet Interval File (WEBSRV) is derived from the SMF type 120 subtype 8 records. The Web container execution environment is inside an EJB container. The Web container acts as a Web Server handling HttpSessions and servlet. The EJB container is not aware of the purpose of the activity running inside the WebContainer and only notes that the EJB has been dispatched. A single WebContainer record is created for each Web container.

Some additional work is performed that is outside the scope of individual requests that ran inside a Web container. For example, finalization of HTTP instances, invalidation of HTTP sessions, and so on.

This file is populated with the SAS releases that support $UCS2Xw informat. No observation will be saved when running with an unsupported SAS release.

The following sections describe the file's organization, list data elements maintained, and provide usage hints and examples.

1 - WEBSRV File Organization
2 - WEBSRV Data Elements List
3 - WEBSRV Usage Considerations
4 - WEBSRV Retrieval Examples

5.2.8.1 WEBSRV File Organization

The table below (Figure 5-9) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	SYSID	WEBSRVNM	WEBINAME	WEBAPPL	SRVSNAME
	YEAR	MONTH	DAY	HOUR	ENDTS
DAYS	SYSID	WEBSRVNM	WEBINAME	WEBAPPL	SRVSNAME
	YEAR	MONTH	DAY	HOUR	
WEEKS	SYSID	WEBSRVNM	WEBINAME	WEBAPPL	SRVSNAME
	YEAR	WEEK	ZONE	HOUR	
MONTHS	SYSID	WEBSRVNM	WEBINAME	WEBAPPL	SRVSNAME
	YEAR	MONTH	ZONE		
YEARS	SYSID	WEBSRVNM	WEBINAME	WEBAPPL	SRVSNAME
	YEAR	ZONE			
TABLES	N/A				

Generation Date: Tue, May 12, 2009

```
NOTE:  This file was generated with ESSENTIAL=ALL
       option in effect.  All data elements defined in
       the file are generated.
NOTE:  This file was generated with DERIVED=DEFault
       option in effect.  Whether data elements are
       kept on the file on auxiliary storage or not
       is controlled by the complex definition of the
       DERIVED option.
```

Figure 5-9. WEBSRV Sort Sequence and Data Granularity

5.2.8.2 WEBSRV Elements List

```
_ins000e.html

GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *  Element    Description (LABEL)
 -------  --------   ----------------------------------------
```

```
Sequence/Summary Data Elements

 XD....E  DAY      - Day of Month
 XDW...E  HOUR     - Hour of Day
 XDWM..E  MONTH    - Month of Year
 XDWMY.E  SRVSNAME - Servlet Name
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBAPPL  - WebApplication Name
 XDWMY.E  WEBINAME - WebSphere Server Instance Name
 XDWMY.E  WEBSRVNM - WebSphere Server Name
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.E  DURATION - Recording Interval Time
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  INTERVLS - Number of Recording Intervals
 X.....E  MICSVER  - CA MICS Version Number
 XDWMY.E  STARTTS  - Start Time Stamp
 XDWMY.E  WEBCELL  - WebSphere Cell Name
 XDWMY.E  WEBNODE  - WebSphere Node Name
 XDWMY.E  WEBSHOST - WebSphere Host Name

Retained Data Elements

 X.....   SRVLSTS  - LoadedSince Timestamp

Accumulated Data Elements

 XDWMY.E  SRVERRTO - Errors Encountered During Execution
 XDWMY.E  SRVLSERV - Servlet Loaded Count
 XDWMY.E  SRVREQTO - Servlet Service Requested Count
 XDWMY.E  SRVTOCPU - Total CPU Time
 XDWMY.E  SRVTORTM - Total Response Time

Minimum Data Elements

 XDWMY.E  SRVMNCPU - Min CPU Time
 XDWMY.E  SRVMNRTM - Min Response Time

Maximum Data Elements

 XDWMY.E  SRVMXCPU - Max CPU Time
 XDWMY.E  SRVMXRTM - Max Response Time
```

```
Derived Data Elements

 XDWMY.E  SRVAVCPU - Avg CPU Time
 XDWMY.E  SRVAVRTM - Avg Response Time
```

5.2.8.3 WEBSRV Usage Considerations

The WEBSRV file is derived from SMF type 120 subtype 8 records. Each record contains activity that is running inside the WebSphere for z/OS application server's WebContainer. There can be multiple observations per WebApplication within a WebContainer, and each WebApplication can contain multiple servlet information. The file provides statistics about every servlet that is referenced by the WebApplication.

5.2.8.4 WEBSRV Retrieval Examples

This section presents a typical WEBSRV retrieval example.

```
Produce a report showing the observations where the
servlet response time exceeds 7 seconds.

PROC PRINT DATA=&PWEBX..WEBSRV01;
  ID WEBCELL WEBNODE WEBSRVNM WEBINAME WEBAPPL SRVSNAME;
  VAR SRVMNRTM SRVMXRTM;
  WHERE SRVMXRTM GT 7;
```

5.2.9 JVM Heap Interval File (WEBHPI)

The JVM Heap Interval (WEBHPI) file is derived from the SMF type 120, subtype 3 records. The heap interval data is logged with subtype 3, which is written each time an activity occurs in the J2EE (Java II Enterprise Edition) servant (server region) at the site-specified interval. It contains information about the specific heap and its size.

The following sections describe the file's organization, list data elements maintained, and provide usage hints and examples.

```
1 - WEBHPI File Organization
2 - WEBHPI Data Elements List
3 - WEBHPI Usage Considerations
4 - WEBHPI Retrieval Examples
```

5.2.9.1 WEBHPI File Organization

The table below (Figure 5-10) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

Note: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	SYSID	WEBSRVNM	WEBINAME	HPIREGN	HPIHEAPI
	YEAR	MONTH	DAY	HOUR	ENDTS
DAYS	SYSID	WEBSRVNM	WEBINAME	HPIREGN	HPIHEAPI
	YEAR	MONTH	DAY	HOUR	
WEEKS	SYSID	WEBSRVNM	WEBINAME	HPIREGN	HPIHEAPI
	YEAR	WEEK	ZONE	HOUR	
MONTHS	SYSID	WEBSRVNM	WEBINAME	HPIREGN	HPIHEAPI
	YEAR	MONTH	ZONE		
YEARS	N/A				
TABLES	N/A				

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-10. WEBHPI Sort Sequence and Data Granularity

5.2.9.2 WEBHPI Elements List

```
_ins000e.html

GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *  Element    Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 XDW...E  HOUR     - Hour of Day
 XDWM..E  HPIHEAPI - Heap ID
 XDWM..E  HPIREGN  - Server Instance Region
 XDWM..E  MONTH    - Month of Year
 XDWM..E  SYSID    - System Identifier
 XDWM..E  WEBINAME - WebSphere Server Instance Name
 XDWM..E  WEBSRVNM - WebSphere Server Name
 XDW...E  WEEK     - Week of Year
 XDWM..E  YEAR     - Year of Century
 XDWM..E  ZONE     - Time Zone

Common Data Elements

 XDW...E  DAY      - Day of Month
 X.....E  DAYNAME  - Name of Day of Week
 XDWM..E  DURATION - Recording Interval Time
 XDWM..E  ENDTS    - End Time Stamp
 XDWM..E  INTERVLS - Number of Recording Intervals
 X.....E  MICSVER  - CA MICS Version Number
 XDWM..E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp
 XDWM..E  WEBCELL  - WebSphere Cell Name
 XDWM..E  WEBNODE  - WebSphere Node Name
 XDWM..E  WEBSHOST - WebSphere Host Name
```

```
Accumulated Data Elements

 XDWM..E  HPIALOCF - Heap Allocation Failures Count

Minimum Data Elements

 XDWM..E  HPIMNHIF - Min Free Bytes
 XDWM..E  HPIMNHIT - Min Heap Size Bytes
 XDWM..E  HPIMNKBF - Min Free Bytes (Kbytes)
 XDWM..E  HPIMNKBT - Min Heap Size Bytes (Kbytes)
 XDWM..E  HPIMNMBF - Min Free Bytes (Mbytes)
 XDWM..E  HPIMNMBT - Min Heap Size Bytes (Mbytes)

Maximum Data Elements

 XDWM..E  HPIMXHIF - Max Free Bytes
 XDWM..E  HPIMXHIT - Max Heap Size Bytes
 XDWM..E  HPIMXKBF - Max Free Bytes (Kbytes)
 XDWM..E  HPIMXKBT - Max Heap Size Bytes (Kbytes)
 XDWM..E  HPIMXMBF - Max Free Bytes (Mbytes)
 XDWM..E  HPIMXMBT - Max Heap Size Bytes (Mbytes)

Derived Data Elements

 XDWM..E  HPIAVHIF - Avg Free Bytes
 XDWM..E  HPIAVHIT - Avg Heap Size Bytes
 XDWM..E  HPIAVKBF - Avg Free Bytes (Kbytes)
 XDWM..E  HPIAVKBT - Avg Heap Size Bytes (Kbytes)
 XDWM..E  HPIAVMBF - Avg Free Bytes (Mbytes)
 XDWM..E  HPIAVMBT - Avg Heap Size Bytes (Mbytes)
```

5.2.9.3 WEBHPI Usage Considerations

The WEBHPI file is derived from SMF type 120 subtype 3 records. Each record contains information from the server interval record and one or more heap records. Each observation in this file contains detail information that is aggregated at the interval level. Multiple heap records in one subtype 3 heap section result in multiple observations in this file. The average, minimum, and maximum value elements are taken from the raw data.

5.2.9.4 WEBHPI Retrieval Examples

This section presents a typical WEBHPI retrieval example.

Produce a report showing the observations where allocation failures to the heap exceed 0.

```
PROC PRINT DATA=&PWEBX..WEBHPI01;
  ID WEBCELL WEBNODE WEBSRVNM WEBINAME HEAPID;
  VAR HPIAVHIT;
  WHERE HPIALOCF GT 0;
```

5.2.10 Web Request Activity File (WEBWRA)

The Web Request Activity (WEBWRA) file is derived from the SMF type 120, subtype 9 records. The request activity data is written each time a request is received by the WebSphere Application Server (WAS). The file provides detail information about the type of request, servant where the request ran, the CPU time consumed by the request, etc.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WEBWRA File Organization
2 - WEBWRA Data Elements List
3 - WEBWRA Usage Considerations
4 - WEBWRA Retrieval Examples

5.2.10.1 WEBWRA File Organization

The table (Figure 5-11) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your site.

Timespan	Level of Data Granularity					
DETAIL	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	MONTH	DAY	
	HOUR	ENDTS				
DAYS	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	MONTH	DAY	
	HOUR					
WEEKS	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	WEEK	ZONE	
	HOUR					
MONTHS	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	MONTH	ZONE	
YEARS	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	ZONE		
TABLES	N/A					

Generation Date: Thu, Feb 14, 2013

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are

```
kept on the file on auxiliary storage or not
is controlled by the complex definition of the
DERIVED option.
```

Figure 5-11. WEBWRA Sort Sequence and Data Granularity

5.2.10.2 WEBWRA Elements List

The table below identifies data elements contained in this
file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element
is supported. The timespans are indicated by
the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with
the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are
defined by each installation when CA MICS is installed.
Therefore, this table has been generated as part of the
installation process to accurately reflect the CA MICS
system at your installation.

GENERATION DATE: Thu, Feb 14, 2013

Note: Essential data elements are identified by an "E"
under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XDWM..E	MONTH	- Month of Year

```
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBACT1  - DIVISION
 XDWMY.E  WEBACT2  - DEPARTMENT
 XDWMY.E  WEBACT3  - SERVER APPLICATION
 XDWMY.E  WEBINAME - WebSphere Server Instance Name
 XDWMY.E  WEBRTYPE - Web Request Type
 XDWMY.E  WEBSRVNM - WebSphere Server Name
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp
 XDWMY.E  SYSNAME  - System Name
 XDWMY.E  SYSPLEX  - Sysplex Name
 XDWMY.E  WEBCELL  - WebSphere Cell Name
 X.....E  WEBCY    - Enclave Token
 XDWMY.E  WEBNODE  - WebSphere Node Name

Retained Data Elements

 X.....E  WRABL    - Server PID
 X.....E  WRABT    - Controller Jobname
 X.....E  WRABU    - Controller Job ID
 X.....E  WRABV    - Controller STOKEN
 X.....E  WRABW    - Controller ASID
 X.....   WRABX    - Controller UUID
 X.....   WRABY    - Server UUID
 X.....E  WRABZ    - Daemon Group Name
 X.....E  WRACG    - Dispatched Servant PID
 X.....E  WRACH    - Dispatched Servant ID
 X.....E  WRACJ    - Completion Code
 X.....E  WRACM    - Time Request Received
 X.....E  WRACN    - Time Request Queued
 X.....E  WRACO    - Time Request Dispatched
 X.....E  WRACP    - Time Dispatch Completed
 X.....E  WRACQ    - Time Controller Finished Request
 XD....   WRACR    - Job Name for Dispatch Servant
 XD....   WRACS    - Job ID for Dispatch Servant
 X.....   WRACT    - Dispatch Servant STOKEN
 X.....   WRACU    - ASID for Dispatch Servant
 X.....   WRACV    - Address of the Dispatch TCB
 X.....   WRACW    - TTOKEN for Dispatch TCB
 X.....E  WRADG    - zAAP CPU Normalization Factor IWMEQTME
 X.....E  WRADJ    - zAAP CPU Normalization Factor IWM4EDEL
```

```
 X.....E  WRADO    - Enclave Delete Response Time Ratio
 X.....   WRADQ    - Global Transaction ID
 X.....   WRADR    - Dispatch Timeout Value
 XD....E  WRADS    - Transaction Class
 X.....E  WRAEI    - Target Port of Request
 X.....E  WRAEK    - Request Origin
 X.....E  WRAEOAPP - IIOP Application Name
 X.....E  WRAEOCIC - CICS Imported Transaction Name
 X.....E  WRAEOCLS - IIOP Class Name
 X.....E  WRAEOCOM - IIOP Component Name
 X.....E  WRAEOMLP - MDB Message Listener Port
 X.....E  WRAEOMOD - IIOP Module Name
 X.....E  WRAEOMTH - IIOP Method Name
 X.....E  WRAEOOLA - OLA Name
 X.....E  WRAEOSEL - MDB Selector Name
 X.....E  WRAEOTGH - HTTP Target Host
 X.....E  WRAEOTGP - HTTP Target Port
 X.....E  WRAEOURI - HTTP URI
 X.....   WRAESII  - Invocation Identity
 X.....   WRAESRI  - Received Identity
 X.....   WRAESSI  - Server Identity
 X.....   WRAFR    - Stalled Thread Dump Action
 X.....   WRAFS    - CPU Time Used Dump Action
 X.....   WRAFT    - DPM Dump Action
 X.....   WRAFU    - Timeout Recovery
 X.....   WRAFV    - Dispatch Timeout
 X.....   WRAFW    - Queue Timeout
 X.....   WRAFX    - Request Timeout
 X.....   WRAFY    - CPU Time Used Limit
 X.....   WRAFZ    - DPM Interval
 X.....   WRAGA    - Message Tag
 X.....   WRAGI    - Obtained Affinity RName
 X.....   WRAGK    - Routing Affinity RName
 XDWMY.E  WRAGMTOF - GMT Offset
 X.....E  WRAHV    - Service Level
 X.....E  WRAPVER  - WAS Product Version
 X.....E  WRASMFTS - SMF Record Timestamp
 XDWM..E  WRASPCNF - Specialty Engine Normalization Factor

Accumulated Data Elements

 XDWMY.E  WRACI    - CPU Time TIMEUSED Macro
 .DWMY.E  WRACOST  - Processing Cost
 XDWMY.E  WRACPUTM - CP CPU Time
 XDWMY.E  WRACX    - Specialty CPU Time TIMEUSED Macro
 X.....E  WRADA    - CPU Time IWMEQTME Macro
 X.....E  WRADA2   - CPU Time IWMEQTME Macro Delta
 X.....E  WRADB    - zAAP CPU Time IWMEQTME Macro
 X.....E  WRADB2   - zAAP CPU Time IWMEQTME Macro Delta
```

```
 X.....E  WRADC    - zAAP CPU Time on CP IWMEQTME Macro
 X.....E  WRADC2   - zAAP CPU Time on CP IWMEQTME Macro Dlt
 X.....E  WRADD    - zIIP CPU Time on CP IWMEQTME Macro
 X.....E  WRADD2   - zIIP CPU Time on CP IWMEQTME Macro Dlt
 X.....E  WRADE    - zIIP CPU Qualified Time IWMEQTME Macro
 X.....E  WRADE2   - zIIP CPU Qual Time IWMEQTME Macro Dlt
 X.....E  WRADF    - zIIP CPU Time IWMEQTME Macro
 X.....E  WRADF2   - zIIP CPU Time IWMEQTME Macro Delta
 XDWMY.E  WRADH    - CPU Time IWM4EDEL Macro
 XDWMY.E  WRADI    - zAAP CPU Time IWM4EDEL Macro
 XDWMY.E  WRADK    - zIIP CPU Time IWM4EDEL Macro
 XDWMY.E  WRADL    - Enclave Delete zIIP Service Units
 XDWMY.E  WRADM    - Enclave Delete zAAP Service Units
 XDWMY.E  WRADN    - Enclave Delete CPU Service Units
 XDWMY.E  WRADSPTM - Web Request Dispatch Time
 XDWMY.E  WRADU    - Times Enclave created by Server
 XDWMY.E  WRADX    - Times One Way IIOP Requests
 XDWMY.E  WRADZ    - Times Queued to Specific Server
 XDWMY.E  WRAEG    - Bytes Received
 XDWMY.E  WRAEH    - Bytes Sent
 XDWMY.E  WRAFQ    - Times Transaction Name Propagated
 XDWMY.E  WRANZRC  - Times Request End With Non-Zero RC
 XDWMY.E  WRAPRQTM - Web Request Pre-Queue Time
 XDWMY.E  WRAQTM   - Web Request Queue Time
 XDWMY.E  WRAREQST - Total Web Requests Processed
 XDWMY.E  WRARGMS  - Times WLM Response Goal Missed
 XDWMY.E  WRARGMT  - Times WLM Response Goal Met
 XDWMY.E  WRASPNTM - Normalized zIIP CPU Time
 XDWMY.E  WRASUCTM - zIIP Eligible CPU Time on CP
 XDWMY.E  WRASUPTM - zIIP CPU Time
 XDWMY.E  WRATRSTM - Total Web Request Response Time
 XDWMY.E  WRAXCSTM - CP CPU Time Minus zIIP/zAAP Eligible
 XDWMY.E  WRAZACTM - zAAP Eligible CPU Time on CP
 XDWMY.E  WRAZAPTM - zAAP CPU Time
 XDWMY.E  WRAZPNTM - Normalized zAAP CPU Time

Derived Data Elements

 XDWMY.E  WRAAVDTM - Avg Request Dispatch Time
 XDWMY.E  WRAAVQTM - Avg Request Queue Time
 XDWMY.E  WRAAVTTM - Avg Request Response Time
 XDWMY.E  WRAPCGMS - Pct WLM Response Goal Missed
 XDWMY.E  WRAPCGMT - Pct WLM Response Goal Met
```

5.2.10.3 WEBWRA Usage Considerations

The WEBWRA file is derived from SMF type 120 subtype 09 records. Each record contains ten triplet sections (v7) or 11 triplet sections (v8) with platform and z/OS specific details for every web request. Each observation in the MICS file contains complete details for the web request activity.

The WEBWRA file interfaces with Accounting and Chargeback to support charging for CPU consumption and network I/O if desired.

Understanding the WEBWRA File CPU Time Data Elements

The following discussion explains how the WEBWRA file CPU time measurements are calculated from the various raw metrics found in the SMF type 120 subtype 9 record. The raw metrics include values that are redundant, contain a combination of normalized and unnormalized values, and in the case of requests where enclaves are re-used, contain values that include CPU consumption from previous uses of the enclave.

Most data elements in the WEBWRA file are named WRAxx, where xx correlates to the raw data metric naming convention adopted by IBM when designing the SMF type 120 subtype 9 record. For example:

IBM metric Name	WEBWRA file data element name
SMF1209BG	WRABG
SMF1209BH	WRABH

-and so on.

This convention allows users familiar with the SMF type 120 subtype 9 record metrics to easily identify the corresponding CA MICS WEBWRA file data elements.

The raw data CPU time related metrics found in the SMF type 120 record are kept in the WEBWRA file, using the same convention shown above, but additional, more traditionally named CPU time related data elements are added that simplify CPU resource use analysis and chargeback. These additional data elements are listed below:

WRACPUTM - CP CPU Time
WRAXCSTM - CP CPU Time Without zAAP/zIIP Eligible

WRASUPTM - zIIP CPU Time

```
WRASPNTM - Normalized zIIP CPU Time
WRASUCTM - zIIP Eligible CPU Time on CP

WRAZAPTM - zAAP CPU Time
WRAZPNTM - Normalized zAAP CPU Time
WRAZACTM - zAAP Eligible CPU Time on CP
```

```
The CA MICS WEBWRA file data elements previously listed are derived from the following raw CPU time-related metrics found in
the SMF type 120 subtype 9 record. These various CPU time-related metrics are obtained by WebSphere issuing 3 different IBM
Macro calls, TIMEUSED, IWMEQTME, and IWM4EDEL. The values returned from the macro calls might or might not be populated.  For
example, if the enclave servicing the just-completed request is immediately re-used by a different request, the values returned
by the IWM4EDEL macro contain zeros. And in this same multi-use enclave scenario, the values returned by the IWMEQTME macro are
cumulative as each request re-uses the same enclave:

--Obtained by WebSphere using the--     --Obtained by WebSphere using the--     --Obtained by WebSphere using the--
          TIMEUSED Macro                          IWMEQTME Macro                          IWM4EDEL Macro

IBM Metric  CA MICS                     IBM Metric  CA MICS                     IBM Metric  CA MICS
Name        Name     Description        Name        Name     Description        Name        Name     Description
---------  -------   --------------     ---------  -------   --------------     ---------  -------   --------------
SMF1209CI  (WRACI) - Total CPU Time     SMF1209DA (WRADA) - Total CPU Time      SMF1209DH (WRADH) - Total CPU Time
                                        SMF1209DB (WRADB) - zAAP CPU Time       SMF1209DI (WRADI) - zAAP CPU Time
SMF1209CX  (WRACX) - Specialty Eng      SMF1209DC (WRADC) - zAAP Time on CP     SMF1209DK (WRADK) - zIIP CPU Time
                     CPU Time           SMF1209DD (WRADD) - zIIP Time on CP
                                        SMF1209DE (WRADE) - zIIP Eligible (Note 3)
                                        SMF1209DF (WRADF) - zIIP CPU Time
                                        SMF1209DG (WRADG) - zAAP Factor
                                                            Normalization

For requests where the enclave was used once, and then deleted, the CPU time metrics are populated from the values
returned by the IWM4EDEL macro as follows:

WRACPUTM - CP CPU Time                          = WRADH - (WRADI + WRADK) ;
WRAXCSTM - CP CPU Time Without zAAP/zIIP Eligible = WRACPUTM - (WRADD + WRADC) ;

WRASUPTM - zIIP CPU Time                        = WRADK / WRADG ;   (Note 1)
WRASPNTM - Normalized zIIP CPU Time             = WRADK ;
WRASUCTM - zIIP Eligible CPU Time on CP         = WRADD / WRADG ;   (Note 1)

WRAZAPTM - zAAP CPU Time                        = WRADI ;
WRAZPNTM - Normalized zAAP CPU Time             = WRADI * WRADG ;   (Note 2)
WRAZACTM - zAAP Eligible CPU Time on CP         = WRADC / WRADG ;   (Note 2)

For requests where the enclave was used multiple times, the process involved in determining CPU time values is
complicated.  In these cases, the values associated with the IWME4DEL macro contain zeros, except for the final use of
the enclave, so for requests 1 through n-1 in a multi-use enclave scenario, the values returned by the IWMEQTME macro
are used.  For the final use of the enclave, the values returned by the IWME4DEL macro are populated, and are used
for the CP, zIIP, and zAAP CPU time derivations.

An additional complication arises because the values returned by the IWMEQTME macro are cumulative over each use of
the enclave. For the first use of an enclave, the raw metrics represent the actual CPU usage.  For subsequent requests
using the same enclave, however, the metric values from the previous request are subtracted from the values in the
current request record to determine the CPU time used by the current request. This delta processing is performed during
raw data read-up, and the delta results are stored in the following data elements:

WRADA2 - Total CPU Time Delta
WRADB2 - zAAP CPU Time Delta
WRADC2 - zAAP Time on CP Delta
WRADD2 - zIIP Time on CP Delta
WRADE2 - zIIP Eligible Delta     (Note 3)
WRADF2 - zIIP CPU Time Delta
$SPAG
After the correct CPU time values are determined for SMF type 120 records involving multiple uses of an enclave, the
following formulas are used to populate the CPU time related data elements as shown below:

WRACPUTM - CP CPU Time                          = WRADA2 - (WRADB2 + WRADF2) ;
WRAXCSTM - CP CPU Time Without zAAP/zIIP Eligible = WRACPUTM - (WRADD2 + WRADC2) ;

WRASUPTM - zIIP CPU Time                        = WRADF2 / WRADG;  (Note 1)
WRASPNTM - Normalized zIIP CPU Time             = WRADF2;
WRASUCTM - zIIP Eligible CPU Time on CP         = WRADD2 / WRADG;  (Note 1)

WRAZAPTM - zAAP CPU Time                        = WRADB2;
WRAZPNTM - Normalized zAAP CPU Time             = WRADB2 * WRADG;  (Note 2)
WRAZACTM - zAAP Eligible CPU Time on CP         = WRADC2 / WRADG;  (Note 2)

Note 1: zIIP CPU times, as recorded in the SMF type 120 subtype 9 record are normalized. To determine actual zIIP
        processor CPU time, the normalized value must be divided by the specialty engine normalization factor. The
        factor provided in the raw data is for zAAP processors, but in current IBM mainframe design, zAAP and zIIP
```

```
        processors always run at the same cycle speed, so the zAAP normalization factor is also the zIIP normalization
        factor.

Note 2: zAAP CPU times, as recorded in the SMF type 120 subtype 9 record are not normalized. To determine normalized zAAP
        processor CPU time, the zAAP CPU time value must be multiplied by the zAAP engine normalization factor.

Note 3: SMF type 120 subtype 9 records consistently record binary zeros in raw data field SMF1209DE, corresponding to
        CA MICS data elements WRADE and WRADE2, which should show the zIIP processor eligible CPU time.
```

5.2.10.4 WEBWRA Retrieval Examples

This section presents a typical WEBWRA retrieval example.

```
Produce summary reports showing the web request activity
for applications running in a server of a cluster by
request type.

PROC SORT DATA=&WEBD..WEBWRA01 OUT=TEMP;
  BY WEBSRVNM WEBINAME WEBRTYPE;
RUN;

DATA TEMP;
  SET TEMP;
    BY WEBSRVNM WEBINAME WEBRTYPE;
    %WRASUM;
END;
RUN;

PROC PRINT DATA=TEMP LABEL;
  ID WEBSRVNM WEBINAME WEBRTYPE;
  VAR WRAREQST WRAPRQTM WRAQTM WRADSPTM WRATRSTM WRAVTTM
      WRAPCGMT;
  TITLE 'Web Request Response Time Report';
RUN;

PROC PRINT DATA=TEMP LABEL;
  ID WEBSRVNM WEBINAME WEBRTYPE;
  VAR WRAREQST WRACPUTM WRAXCSTM WRASUPTM WRASPNTM
      WRASUCTM WRAZAPTM WRAZPNTM WRAZACTM;
  TITLE 'Web Request CPU Usage Report';
RUN;
```

5.2.11 Web Request CPU Breakdown File (WEBWCB)

The Web CPU Usage Breakdown (WEBWCB) file is derived from the SMF type 120, subtype 9 record, CPU Usage Breakdown section. The section is optional, and may only contain data for certain web request activity of interest. There is some overhead in collecting the data for this section, so it may be populated for select web activity for short periods of time that are of particular interest to study in more detail.

When the CPU Usage Breakdown section is populated, it may produce 1 to 30 records per web request. Each time the web request calls a web or EJB container, Websphere can capture the metrics contained in this file for deeper study into where the request is spending its time within the application.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

5.2.11.1 WEBWCB File Organization

This table (Figure 5-12) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your site.

Timespan	Level of Data Granularity				
DETAIL	SYSID MONTH	WEBSRVNM DAY	WEBINAME HOUR	WEBRTYPE ENDTS	YEAR
DAYS	N/A				
WEEKS	N/A				
MONTHS	N/A				
YEARS	N/A				
TABLES	N/A				

Generation Date: Thu, Feb 14, 2013

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-12. WEBWCB Sort Sequence and Data Granularity

5.2.11.2 WEBWCB Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Feb 14, 2013

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

X.....E	DAY	- Day of Month
X.....E	HOUR	- Hour of Day
X.....E	MONTH	- Month of Year
X.....E	SYSID	- System Identifier
X.....E	WEBINAME	- WebSphere Server Instance Name

```
 X.....E  WEBRTYPE - Web Request Type
 X.....E  WEBSRVNM - WebSphere Server Name
 X.....E  WEEK     - Week of Year
 X.....E  YEAR     - Year of Century
 X.....E  ZONE     - Time Zone

Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 X.....E  ENDTS    - End Time Stamp
 X.....E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp
 X.....E  SYSNAME  - System Name
 X.....E  SYSPLEX  - Sysplex Name
 X.....E  WEBCELL  - WebSphere Cell Name
 X.....E  WEBCY    - Enclave Token
 X.....E  WEBNODE  - WebSphere Node Name

Retained Data Elements

 X.....E  WCBBT    - Controller Jobname
 X.....E  WCBBU    - Controller Job ID
 X.....E  WCBBV    - Controller STOKEN
 X.....E  WCBBW    - Controller ASID
 X.....   WCBBX    - Controller UUID
 X.....   WCBBY    - Server UUID
 X.....E  WCBBZ    - Daemon Group Name
 X.....E  WCBCG    - Dispatched Servant PID
 X.....E  WCBCH    - Dispatched Servant ID
 X.....E  WCBEU    - Container Type
 X.....E  WCBEY    - AMC/Web Application Name
 X.....E  WCBFA    - Method/Servlet Name

Accumulated Data Elements

 X.....E  WCBEV    - Object CPU Time
 X.....E  WCBEW    - Number of Object Executions
 X.....E  WCBFI    - Object Elapsed Time
```

5.2.13.3 WEBWCB Usage Considerations

The WEBWCB file is derived from SMF type 120 subtype 09 records, CPU usage breakdown section. The CPU usage breakdown section is not activated by default. Typically the breakdown section is activated for a sort time and focused on web request activity that required special study, as there is some overhead in collecting the information.

When the web request application calls another web part that requires the web container or EJB container, Websphere can gather the usage metrics for each of these parts. Usage on up to 30 of these called parts can be recorded for the web request.

If you study the WCB file for particular web requests, you will be able to determine which parts of the application are consuming the most CPU. You will also be able to identify the AMC (Application/Module/Component) or web application name, and the method or servlet name used by each part of the web request.

Most of the information needed for analysis is in the WCB file. However, if you need other information from the other files populated from the SMF 120 subtype 9 record, you can merge the files by SYSID and WEBCY (enclave token) to correlate the data.

5.2.11.4 WEBWCB Retrieval Examples

This section presents a typical WEBWCB retrieval example.

```
Produce a detail report showing the web request CPU
breakdown for a particular time period.  Substitute any
subsetting logic to extract the desired data for a study.

DATA TEMP;
  SET &WEBX..WEBWCB01;
    IF WEBCY    = '000000E000622D85'X;  /* <-- Any   */
/*  IF HOUR=10;              /*  subsetting criteria */
RUN;                         /*  to focus the report */

PROC SORT DATA=TEMP;
  BY SYSID WEBCY ENDTS;
RUN;

PROC PRINT DATA=TEMP LABEL;
  BY SYSID WEBCY;
  PAGEBY WEBCY;
  SUMBY WEBCY;
  SUM WCBEV WCBFI WCBEW;
  VAR WEBSRVNM WEBINAME WEBRTYPE
      WCBBT WCBBT WCBEU WCBEV WCBFI WCBEW
      WCBEY WCBFA ENDTS;
  TITLE 'Web Request CPU Breakdown Study';
RUN;
```

5.2.12 Web User Data File (WEBWUD)

The Web User Data (WEBWUD) file is derived from the SMF type 120, subtype 9 records, User Data section. Through a Websphere API, the application can record any information deemed appropriate to further identify the application and functions being performed by the Web Request into the SMF record.

This file is optional, and can be configured to be stored on disk in the Detail database, on Tape only, or both.

It is possible for the application to populate up to five 2048 byte user fields that are formatted according to the users choice. As the User Areas are found in the SMF record, they are written to the WEBWUD file. This file can be used to merge back into other WEB files when this application data is required.

Note: The information in these user fields is available during the processing of the SMF data to populate user account code fields in the Web Request Activity (WEBWRA) and WEB Request Asynch Activity (WEBWAA) files.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WEBWUD File Organization
2 - WEBWUD Data Elements List
3 - WEBWUD Usage Considerations
4 - WEBWUD Retrieval Examples

5.2.12.1 WEBWUD File Organization

This table (Figure 5-13) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your site.

Timespan	Level of Data Granularity
DETAIL	SYSID WEBSRVNM WEBINAME WEBRTYPE ENDTS
DAYS	N/A
WEEKS	N/A
MONTHS	N/A
YEARS	N/A
TABLES	N/A

Generation Date: Thu, Feb 14, 2013

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-13. WEBWUD Sort Sequence and Data Granularity

5.2.12.2 WEBWUD Elements List

The table below identifies data elements contained in this
file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element
is supported. The timespans are indicated by
the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with
the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are
defined by each installation when CA MICS is installed.
Therefore, this table has been generated as part of the
installation process to accurately reflect the CA MICS
system at your installation.

GENERATION DATE: Thu, Feb 14, 2013

Note: Essential data elements are identified by an "E"
under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

X.....E	ENDTS	- End Time Stamp
X.....E	SYSID	- System Identifier
X.....E	WEBINAME	- WebSphere Server Instance Name

```
 X.....E  WEBRTYPE - Web Request Type
 X.....E  WEBSRVNM - WebSphere Server Name

Common Data Elements

 X.....E  DAY      - Day of Month
 X.....E  DAYNAME  - Name of Day of Week
 X.....E  HOUR     - Hour of Day
 X.....E  MONTH    - Month of Year
 X.....E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp
 X.....E  SYSNAME  - System Name
 X.....E  SYSPLEX  - Sysplex Name
 X.....E  WEBCELL  - WebSphere Cell Name
 X.....E  WEBCY    - Enclave Token
 X.....E  WEBNODE  - WebSphere Node Name
 X.....E  WEEK     - Week of Year
 X.....E  YEAR     - Year of Century
 X.....E  ZONE     - Time Zone

Retained Data Elements

 X.....E  WUDFF    - User Data Type
 X.....E  WUDFH    - User Data
```

5.2.12.3 WEBWUD Usage Considerations

The WEBWUD file contains the contents of the user data sections in the SMF 120 subtype 9 records. There can be up to five 2K byte free-form user sections. It was not practical to put this information into the WEB Request Activity (WEBWRA) file. However this information might be needed for special studies.

The WEBWUD file is typically merged with another file like the WEB Request Activity (WEBWRA) file typically by SYSID and Enclave Token (WEBCY) to associate the user data to its associated web request record.

The user data field (WUDFH) contains information passed to the SMF record from the application through an API that might identify the following about the application: what the request asked for, what was retrieved, the business purpose, and so on.

This information is only valid at the Detail level, and merged with other files at the Detail level.

5.2.12.4 WEBWUD Retrieval Examples

This section presents a typical WEBWUD retrieval example.

```
Produce a summary report showing the web request activity
with the application and application function extracted
from the user data field.

Note:  The information below is assuming for example sake
that the application name and application function
performed are in the second and third positions in the
user date field and the fields are separated by colons.

/* extract desired data from WEBWRA Detail file and
   sort it by SYSID and Enclave Token  */
DATA WRA;
  SET &WEBX..WEBWRA01;
    IF WEBSVRNM = 'myserver';
RUN;
PROC SORT DATA=WRA;
  BY SYSID WEBCY;
RUN;

/* extract desired data from WEBWUD Detail file and   */
/* sort it by SYSID and Enclave Token                 */

DATA WUD;
  SET &WEBX..WEBWUD01;
    IF WEBSVRNM = 'myserver';
RUN;
PROC SORT DATA=WUD;
  BY SYSID WEBCY;
RUN;

/* merge the WRA and WUD files and extract from the   */
/* user data field WUDFH the application and function */
/* performed                                          */

DATA TEMP;
  MERGE WRA  (IN=WRA)
        WUD  (IN=WUD);
    BY SYSID WEBCY;
 IF WUD THEN DO;
   APPL = SCAN(WUDFH,2,':'); /* get application  */
   FUNC = SCAN(WUDFH,3,':'); /* get function     */
 END;
 IF WUD THEN OUTPUT TEMP;
 ELSE RETURN;
```

```
RUN;

/* Summarize the merged file  /*

%LET BY= SYSID WEBSVRNM APPL FUNC;
%LET BREAK = FUNC;
PROC SORT DATA=TEMP;
  BY &BY
RUN;

DATA TEMP2;
  SET TEMP;
  %WRASUM(SUMBY=&BY,SUMBREAK=&BREAK);
RUN;

/* Print the report */

PROC PRINT DATA=TEMP2 LABEL;
  ID SYSID APPL FUNC;
  VAR WRACPUTM WRASUPTM WRAZAPTM WRAPRQTM WRADSPTM WRAQTM
      WRATRSTM;
  LABEL APPL='Application Name'
        FUNC='Application Function';
  TITLE 'Web Request Application Report';
RUN;
```

5.2.13 Web Asynch Activity File (WEBWAA)

The Web Request Asynch Activity (WEBWAA) file is derived from the SMF type 120, subtype 9 records, Asynchronous Data section. The Asynchronous request activity data is written each time an Asynch request is received by the WebSphere Application Server (WAS). The file provides detailed information about the type of request, the servant where the request ran, and the CPU time consumed by the request. In addition, information about the execution context of the web request that created the asynchronous work is available for use in correlating with the normal web request that created it.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WEBWAA File Organization
2 - WEBWAA Data Elements List
3 - WEBWAA Usage Considerations
4 - WEBWAA Retrieval Examples

5.2.13.1 WEBWAA File Organization

This table (Figure 5-14) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your site.

Timespan	Level of Data Granularity					
DETAIL	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	MONTH	DAY	
	HOUR	ENDTS				
DAYS	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	MONTH	DAY	
	HOUR					
WEEKS	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	WEEK	ZONE	
	HOUR					
MONTHS	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	MONTH	ZONE	
YEARS	SYSID	WEBSRVNM	WEBINAME	WEBRTYPE	WEBACT1	
	WEBACT2	WEBACT3	YEAR	ZONE		
TABLES	N/A					

Generation Date: Thu, Feb 14, 2013

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are

```
kept on the file on auxiliary storage or not
is controlled by the complex definition of the
DERIVED option.
```

Figure 5-14. WEBWAA Sort Sequence and Data Granularity

5.2.13.2 WEBWAA Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL
D - DAYS
W - WEEKS
M - MONTHS
Y - YEARS
T - TABLES AREA
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Thu, Feb 14, 2013

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)

Sequence/Summary Data Elements

XD....E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XDWM..E	MONTH	- Month of Year
XDWMY.E	SYSID	- System Identifier
XDWMY.E	WEBACT1	- DIVISION

```
 XDWMY.E  WEBACT2  - DEPARTMENT
 XDWMY.E  WEBACT3  - SERVER APPLICATION
 XDWMY.E  WEBINAME - WebSphere Server Instance Name
 XDWMY.E  WEBRTYPE - Web Request Type
 XDWMY.E  WEBSRVNM - WebSphere Server Name
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone

Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  ORGSYSID - Originating System Identification
 X.....E  STARTTS  - Start Time Stamp
 XDWMY.E  SYSNAME  - System Name
 XDWMY.E  SYSPLEX  - Sysplex Name
 XDWMY.E  WEBCELL  - WebSphere Cell Name
 X.....E  WEBCY    - Enclave Token
 XDWMY.E  WEBNODE  - WebSphere Node Name

Retained Data Elements

 X.....E  WAABT    - Controller Jobname
 X.....E  WAABU    - Controller Job ID
 X.....E  WAABV    - Controller STOKEN
 X.....E  WAABW    - Controller ASID
 X.....   WAABX    - Controller UUID
 X.....   WAABY    - Server UUID
 X.....E  WAABZ    - Daemon Group Name
 X.....E  WAAGL    - Time Request Received
 X.....E  WAAGN    - Time Request Dispatched
 X.....E  WAAGO    - Time Dispatch Completed
 X.....E  WAAGP    - Dispatched Servant PID
 X.....E  WAAGQ    - Job Name for Dispatch Servant
 X.....E  WAAGR    - Job ID for Dispatch Servant
 X.....E  WAAGS    - Dispatched Servant Token
 X.....E  WAAGT    - ASID for Dispatch Servant
 X.....E  WAAGU    - Execution Context Task ID
 X.....E  WAAGV    - Execution Context TCB Address
 X.....E  WAAGW    - Execution Context TCB TTOKEN
 X.....E  WAAGX    - Dispatch Task ID
 X.....E  WAAGY    - Address of the Dispatch TCB
 X.....E  WAAGZ    - TTOKEN for Dispatch TCB
 X.....E  WAAHA    - Execution Context Enclave Token
 X.....E  WAAHB    - Dispatch Enclave Token
 X.....E  WAAHC    - WLM Tran Class Used to Create Enclave
 X.....E  WAAHM    - zAAP CPU Normalization Factor IWMEQTME
 X.....E  WAAHQ    - Work Package / Class Name
```

```
 X.....E  WAAHS    - Work Manager Name
 X.....E  WAAHU    - User Identity
 X.....E  WAASMFTS - SMF Record Timestamp
 XDWM..E  WAASPCNF - Specialty Engine Normalization Factor

Accumulated Data Elements

 XDWMY.E  WAAASYNC - Total Asynch Web Requests
 XDWMY.E  WAAAZZTM - zIIP/zAAP CPU Time
 .DWMY.E  WAACOST  - Processing Cost
 XDWMY.E  WAACPUTM - CP CPU Time
 XDWMY.E  WAAELPTM - Total Asynch Request Elapsed Time
 XDWMY.E  WAAEXCTM - Total Asynch Request Execution Time
 XDWMY.E  WAAHE    - Times Enclave Created
 XDWMY.E  WAAHEJ   - Times Enclave Joined
 XDWMY.E  WAAHF    - Times Enclave was Scheduled
 X.....E  WAAHG    - CPU Time IWMEQTME Macro
 X.....E  WAAHH    - zAAP CPU Time IWMEQTME Macro
 X.....E  WAAHI    - zAAP CPU Time on CP IWMEQTME Macro
 X.....E  WAAHJ    - zIIP CPU Time on CP IWMEQTME Macro
 X.....E  WAAHK    - zIIP CPU Qualified Time IWMEQTME Macro
 X.....E  WAAHL    - zIIP CPU Time IWMEQTME Macro
 X.....E  WAAHN    - CPU Time TIMEUSED Macro
 X.....E  WAAHO    - Specialty CPU Time TIMEUSED Macro
 XDWMY.E  WAANZZTM - Normalized zIIP/zAAP CPU Time

Derived Data Elements

 XDWMY.E  WAAAVETM - Avg Asynch Request Elapsed Time
 XDWMY.E  WAAAVXTM - Avg Asynch Request Execution Time
```

5.2.13.3 WEBWAA Usage Considerations

The WEBWAA file contains the contents of the Asynchronous data section in the SMF 120 subtype 9 records. This section became available with Websphere Application Server version 8.

Most data elements in the WEBWAA file are named WAAxx, where xx correlates to the raw data metric naming convention adopted by IBM when designing the SMF type 120 subtype 9 record. For example:

IBM metric Name	WEBWAA file data element name
SMF1209GV	WAAGV
SMF1209GW	WAAGW

and so on.

This convention allows users familiar with the SMF type 120 subtype 9 record metrics to easily identify the corresponding CA MICS WEBWAA file data elements.

A normal web request can spawn or create asynchronous work that does not require a response back to the original web requestor. Also, an asynchronous transaction can create another asynchronous piece of work. A number of elements are available to trace back to the execution context of the creating web request, whether it is a normal web request, or another asynchronous piece of work. If the asynchronous work executed on the same enclave, use the common enclave token date element (WEBCY) to identify the related web requests. If the work was not done on the same enclave, use WAAHA to match with WEBCY of the creating request. Several other execution context elements are also available to help identify the creating web request:

- WAAHA - Execution Context Enclave Token
- WAAGU - Execution Context Task ID
- WAAGV - Execution Context TCB Address
- WAAGW - Execution Context TCB TTOKEN

Understanding the WEBWAA File CPU Time Data Elements

The following discussion explains how the WEBWAA file CPU time measurements are calculated from the various raw metrics found in the SMF type 120 subtype 9 record Asynchronous data section. The Asynchronous work can run on the same enclave as the web request that submitted it, or from another

asynchronous request that submitted it. The CPU metrics acquired from the IWMEQTME macro will contain accumulated CPU time from the prior request using the same enclave. Due to the disconnect between the web request and asynchronous request, the CPU times acquired from the TIMEUSED macro, which provide just the CPU time used for individual asynchronous work are used. This includes CPU time on the CP engine and CPU time on a specialty engine.

The raw data CPU time-related metrics found in the SMF type 120 record Asynchronous section are kept in the WEBWAA file, using the convention previously described, but additional, more traditionally named CPU time-related data elements are added. These elements simplify CPU resource use analysis and chargeback; they are listed below:

WAACPUTM - CP CPU Time

WAAAZZTM - zIIP/zAAP CPU Time
WAANZZTM - Normalized zIIP/zAAP CPU Time

The CA MICS WEBWAA file data elements listed previously are derived from the following raw CPU time related metrics found in the SMF type 120 subtype 9 record. These various CPU time related metrics are obtained by WebSphere issuing two different IBM Macro calls, TIMEUSED and IWMEQTME. The enclave CPU metrics from the IWMEQTME macro might contain CPU time from threads or processes. As a result, we rely solely on the CPU times from the TIMEUSED macro for Asynchronous work.

--Obtained by WebSphere using the-- TIMEUSED Macro			--Obtained by WebSphere using the-- IWMEQTME Macro		
IBM Metric Name	CA MICS Name	Description	IBM Metric Name	CA MICS Name	Description
SMF1209HN	(WAAHO) -	Total CPU Time	SMF1209HG	(WAAHG) -	Total CPU Time
			SMF1209HH	(WAAHH) -	zAAP CPU Time
SMF1209HO	(WAAHO) -	Specialty Eng CPU Time	SMF1209HI	(WAAHI) -	zAAP Time on CP
			SMF1209HJ	(WAAHJ) -	zIIP Time on CP
			SMF1209HK	(WAAHK) -	zIIP Eligible
			SMF1209HL	(WAAHL) -	zIIP CPU Time
			SMF1209HM	(WAAHM) -	zAAP Factor Normalization

5.2.13.4 WEBWAA Retrieval Examples

This section presents a typical WEBWAA retrieval example.

```
Produce a summary report showing the asynchronous activity
for applications running in a server of a cluster by
hour.

PROC SORT DATA=&WEBD..WEBWAA01 OUT=TEMP;
  BY WEBSRVNM WEBINAME HOUR;
RUN;

DATA TEMP;
  SET TEMP;
    BY WEBSRVNM WEBINAME HOUR;
    %WAASUM;
END;
RUN;

PROC PRINT DATA=TEMP LABEL;
  ID WEBSRVNM WEBINAME HOUR;
  VAR WAAASYNC WAAELPTM WAAEXCTM WAAAVETM WAAAVXTM;
  TITLE 'Web Asynchronous Response Time Report';
RUN;

PROC PRINT DATA=TEMP LABEL;
  ID WEBSRVNM WEBINAME HOUR;
  VAR WAAASYNC WAACPUTM WAAAZZTM WAANZZTM;
  TITLE 'Web Asynchronous CPU Usage Report';
RUN;
```

5.3 Web Log Information Area (WLG) Files

This section identifies each file in the WLG Information Area and defines its level of summarization and data sequencing as the file appears in the applicable time-span levels. In addition, it presents the list of data elements contained in each WLG Information Area file.

The files in the WEB Log Files Information Area are:

1 - HTTP Access Log File (WLGSAL)
2 - HTTP Agent Log File (WLGAGL)
3 - HTTP Referer Log File (WLGRFL)
4 - HTTP Error Messages Log File (WLGERL)
5 - HTTP Access Method Summary File (WLGAMS)
6 - HTTP Access Path Summary File (WLGAPS)
7 - WLE Configuration File (WLGCFG)

5.3.1 Server Access Log File (WLGSAL)

The HTTP Access Log (WLGSAL) File is derived from the HTTP Access, FRCA Access, Proxy, and Proxy Cache log data. The data contained in the logs provides information about who accesses WEB server pages, when the access took place, what the access was, whether the access completed successfully or encountered errors, and the amount of data transferred. This log file contains event data and is collected and reported each time a designated system event occurs. It represents the status of the system at the time the event occurred. Finally, the data in the logs determines the quantity of data transmitted.

The HTTP Access Log data, along with the data from the other logs, help you to monitor individual access activity, cache usage, and errors.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WLGSAL File Organization
2 - WLGSAL Data Elements List
3 - WLGSAL Usage Considerations
4 - WLGSAL Retrieval Examples

5.3.1.1 WLGSAL File Organization

The table below (Figure 5-15) identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in that time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBIP
	WEBPORT	YEAR	MONTH	DAY	HOUR
	ENDTS				
DAYS	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBIP
	WEBPORT	YEAR	MONTH	DAY	HOUR
WEEKS	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBIP
	WEBPORT	YEAR	WEEK	ZONE	HOUR
MONTHS	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBIP
	WEBPORT	YEAR	MONTH	ZONE	
YEARS	SYSID	WEBACT1	WEBACT2	WEBACT3	WEBIP
	WEBPORT	YEAR	ZONE		
TABLES	N/A				

Generation Date: Tue, May 12, 2009

```
NOTE:  This file was generated with ESSENTIAL=ALL
       option in effect.  All data elements defined in
       the file are generated.
NOTE:  This file was generated with DERIVED=DEFault
       option in effect.  Whether data elements are
       kept on the file on auxiliary storage or not
       is controlled by the complex definition of the
       DERIVED option
```

Figure 5-15. WLGSAL Sort Sequence and Data Granularity

5.3.1.2 WLGSAL Elements List

```
_ins000e.html

GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *  Element    Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 XD....E  DAY      - Day of Month
 XDW...E  HOUR     - Hour of Day
 XDWM..E  MONTH    - Month of Year
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBACT1  - DIVISION
 XDWMY.E  WEBACT2  - DEPARTMENT
 XDWMY.E  WEBACT3  - SERVER APPLICATION
 XDWMY.E  WEBIP    - Web IP Address
 XDWMY.E  WEBPORT  - Web Server Port Number
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone
```

```
Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  MICSVER  - CA MICS Version Number
 XDWMY.E  STARTTS  - Start Time Stamp
 XDWMY.E  WEBGROUP - Web Group Name
 XDWMY.E  WEBNAME  - Web Server Name
 XDWMY.E  WEBSNAME - Web Server Short Name
 X.....E  WEBSRVRT - Directory for Server Root
 XDWMY.E  WLGLTYPE - Log File Type
 XDWMY.E  WLGSRVNM - WLE Assigned Server Name

Retained Data Elements

 X.....E  SALAUTHI - AUTH Id if Present
 XDWMY.E  SALGMTOF - GMT Offset
 X.....E  SALHSTIP - Host Name or IP Address
 X.....E  SALPATH  - Path to File
 X.....E  SALREQME - Request Method (GET, POST, etc.)
 X.....E  SALREQST - Request Status
 XDWMY.E  SALSRVNM - Server Name
 XDWMY.E  SALSRVRL - Server Release

Accumulated Data Elements

 XDWMY.E  SALACCNT - FRCA (Cache) Access Log Count
 XDWMY.E  SALACNT  - Server Access Log Count
 XDWMY.E  SALBYTET - Bytes Transferred
 XDWMY.E  SALCOST  - Processing Charges
 XDWMY.E  SALKBYTT - KiloBytes Transferred
 XDWMY.E  SALMBYTT - MegaBytes Transferred
 XDWMY.E  SALPCCNT - Proxy Cache Access Log Count
 XDWMY.E  SALPCNT  - Proxy Access Log Count
```

5.3.1.3 WLGSAL Usage Considerations

The WLGSAL file records detail auditing of events as they occur. This file provides information such as who accessed the web page, what web page was accessed, when the web page was accessed, and the quantity of data transferred.

The "Host Name or IP Address" can be up to 254 bytes, if the DNS resolver is active at your site. The data for this field is divided into two CA MICS elements, SALHSTIP (first 200 bytes) and SALHSTI2 (remaining 54 bytes). SALHSTI2 is turned off by default in the file. If you have a "Host Name or IP Address" field longer than 200 bytes, the CA MICS Web Analyzer daily processing will issue warning messages in MICSLOG.

The following checklist must be completed in order to activate the element (SALHSTI2) and to retrofit databases:

1. Change the DETAIL time-span flag from N to 0 (zero) as shown below:

   ```
   NAME SALHSTI2    00  0 N N N N   Host Name or IP Address 2
   ```

2. Submit sharedprefix.MICS.CNTL(WEBCGEN)

3. Retrofit all your WLGSAL files in online databases where this file is active. You only need to add the SALHSTI2 element.

5.3.1.4 WLGSAL Retrieval Examples

This section presents a typical WLGSAL retrieval example.

Produce a report that provides detail about a specific user's web page access activity.

```
PROC PRINT DATA=&PWEBX..WLGSAL01;
 WHERE SALHSTIP = '123.45.678.9';
RUN;
```

5.3.2 HTTP Agent Log (WLGAGL) File

The HTTP Agent Log (WLGAGL) File provides information about the web browser used to access the HTTP Server web page. The file is derived from the HTTP Agent Log data. This file allows you to identify what browser type is most often used to access web pages on the web server. Once the most commonly used browser type for your web server is identified, you can specifically tailor your server for that browser.

The following sections describe the file organization, list data elements maintained, and provide usage hints.

1 - WLGAGL File Organization
2 - WLGAGL Data Elements List
3 - WLGAGL Usage Considerations
4 - WLGAGL Retrieval Examples

5.3.2.1 WLGAGL File Organization

The table below (Figure 5-16) identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	N/A				
DAYS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	DAY	HOUR	AGLBRWSR		
WEEKS	SYSID	WEBIP	WEBPORT	YEAR	WEEK
	ZONE	HOUR	AGLBRWSR		
MONTHS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	ZONE	AGLBRWSR			
YEARS	N/A				
TABLES	N/A				

Generation Date: Tue, May 12, 2009

```
NOTE:  This file was generated with ESSENTIAL=ALL
       option in effect.  All data elements defined in
       the file are generated.
NOTE:  This file was generated with DERIVED=DEFault
       option in effect.  Whether data elements are
       kept on the file on auxiliary storage or not
       is controlled by the complex definition of the
       DERIVED option.
```

Figure 5-16. WLGAGL Sort Sequence and Data Granularity

5.3.2.2 WLGAGL Elements List

```
_ins000e.html


GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *  Element    Description (LABEL)
 -------  --------   ---------------------------------------


Sequence/Summary Data Elements

 .DWM..E  AGLBRWSR - Browser Used to Access
 .D....E  DAY      - Day of Month
 .DW...E  HOUR     - Hour of Day
 .DWM..E  MONTH    - Month of Year
 .DWM..E  SYSID    - System Identifier
 .DWM..E  WEBIP    - Web IP Address
 .DWM..E  WEBPORT  - Web Server Port Number
 .DW...E  WEEK     - Week of Year
 .DWM..E  YEAR     - Year of Century
 .DWM..E  ZONE     - Time Zone

Common Data Elements

 .D....E  DAYNAME  - Name of Day of Week
 .DWM..E  ENDTS    - End Time Stamp
 .DWM..E  MICSVER  - CA MICS Version Number
 .DWM..E  STARTTS  - Start Time Stamp
 .DWM..E  WEBGROUP - Web Group Name
 .DWM..E  WEBNAME  - Web Server Name
 .DWM..E  WEBSNAME - Web Server Short Name
 .DWM..E  WLGLTYPE - Log File Type
 .DWM..E  WLGSRVNM - WLE Assigned Server Name

Retained Data Elements

 .DWM..E  AGLGMTOF - GMT Offset

Accumulated Data Elements

 .DWM..E  AGLCNT   - Agent Log Count
```

5.3.2.3 WLGAGL Usage Considerations

By default, the WLGAGL file is only available at the DAYS and higher timespans. This file is used to quantify hits by browser type. Knowing the most commonly used browser(s) to access server web pages will allow you to target these when modifying or creating applications.

Special considerations or techniques related to using the WLGAGL file are provided below.

1. The WLGAGL file is turned off at the DETAIL timespan in the sharedprefix.MICS.GENLIB(WEBGENIN) member. The file line in WEBGENIN appears as:

   - FILE AGL 00 1 N E Y Y N Y Y Y Agent Log file

   In order to turn on the file at the DETAIL timespan, it will be necessary to change the "N" in the DETAIL timespan to a "Y". The "E" in the DAYS timespan must be retained, as the file requires special processing. The new file line in WEBGENIN appears as:

   - FILE AGL 00 1 Y E Y Y N Y Y Y Agent Log file.

   After making these change you must submit the following jobs:

   sharedprefix.MICS.CNTL(WEBCGEN)

   prefix.MICS.CNTL(COPYLIBU)

   prefix.MICS.CNTL(CYCLEGEN)

As demonstrated above, this file can be activated at the detail level by modifying the FILE statement in sharedprefix.MICS.GENLIB(WEBGENIN), however the detail information will be of limited value.

5.3.2.4 WLGAGL Retrieval Examples

This section present a typical WLGAGL retrieval example.

Print a report listing each server's top 5 browsers.

```
PROC SORT DATA=&PWEBD..WLGAGL01 OUT=WLGAGL01;
BY WEBNAME AGLBRWSR;

PROC SUMMARY DATA=WLGAGL01 NWAY;
  CLASS WEBNAME AGLBRWSR;
  VAR AGLCNT;
  OUTPUT OUT=NEW SUM=AGLCNT;
RUN;

OPTIONS OBS=5;

PROC PRINT;
```

5.3.3 HTTP Referer Log (WLGRFL) File

The HTTP Referer Log (WLGRFL) File is derived from the HTTP Referer Log. It provides information about web pages that referred (linked to) the requested web page. The file contains event data that is logged every time an entry is logged in the Access Log. The information contained in the file are who referred the web page, what web page was requested, and when the web page request was made.

The WLGRFL file, by default, is turned off at the DETAIL time-span. The majority of the links to the web pages could be made from the local server where the web pages are stored, thus making the file more voluminous with information that may not be desirable.

The HTTP Referer Log data, along with the data from the other logs, help you rank specific web sites based on their referral frequency.

The following sections describe the file organization, list data elements maintained, and provide usage hints.

1 - WLGRFL File Organization
2 - WLGRFL Data Elements List
3 - WLGRFL Usage Considerations
4 - WLGRFL Retrieval Examples

5.3.3.1 WLGRFL File Organization

The table below (Figure 5-17) identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	N/A				
DAYS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	DAY	HOUR	RFLHSTIP	RFLLPORT	
WEEKS	SYSID	WEBIP	WEBPORT	YEAR	WEEK
	ZONE	HOUR	RFLHSTIP	RFLLPORT	
MONTHS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	ZONE	RFLHSTIP	RFLLPORT		
YEARS	N/A				
TABLES	N/A				

Generation Date: Tue, May 12, 2009

```
NOTE:  This file was generated with ESSENTIAL=ALL
       option in effect.  All data elements defined in
       the file are generated.
NOTE:  This file was generated with DERIVED=DEFault
       option in effect.  Whether data elements are
       kept on the file on auxiliary storage or not
       is controlled by the complex definition of the
       DERIVED option.
```

Figure 5-17. WLGRFL Sort Sequence and Data Granularity

5.3.3.2 WLGRFL Elements List

```
_ins000e.html


GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *  Element    Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 .D....E  DAY      - Day of Month
 .DW...E  HOUR     - Hour of Day
 .DWM..E  MONTH    - Month of Year
 .DWM..E  RFLHSTIP - Host Name or IP Address
 .DWM..E  RFLLPORT - Client Port
 .DWM..E  SYSID    - System Identifier
 .DWM..E  WEBIP    - Web IP Address
 .DWM..E  WEBPORT  - Web Server Port Number
 .DW...E  WEEK     - Week of Year
 .DWM..E  YEAR     - Year of Century
 .DWM..E  ZONE     - Time Zone
```

```
Common Data Elements

 .D....E  DAYNAME  - Name of Day of Week
 .DWM..E  ENDTS    - End Time Stamp
 .DWM..E  MICSVER  - CA MICS Version Number
 .DWM..E  STARTTS  - Start Time Stamp
 .DWM..E  WEBGROUP - Web Group Name
 .DWM..E  WEBNAME  - Web Server Name
 .DWM..E  WEBSNAME - Web Server Short Name
 ......E  WEBSRVRT - Directory for Server Root
 .DWM..E  WLGLTYPE - Log File Type
 .DWM..E  WLGSRVNM - WLE Assigned Server Name

Retained Data Elements

 .DWM..E  RFLGMTOF - GMT Offset
 ......E  RFLPATH  - Path to File

Accumulated Data Elements

 .DWM..E  RFLCNT   - Referer Log Count
```

5.3.3.3 WLGRFL Usage Considerations

The WLGRFL file defines who referred a request for the resources that were accessed. Information contained in this file is similar to information contained in the WLGSAL file. (See section 5.3.1 of this guide for information on the WLGSAL file.) Due to the information being of a similar nature, the DETAIL time-span in this file is not kept, rather only time-spans at the higher levels.

Special considerations or techniques related to using the WLGRFL file are provided below:

1. The WLGRFL file is turned off at the DETAIL time-span in the sharedprefix.MICS.GENLIB(WEBGENIN) member. The file line in WEBGENIN appears as:

   o FILE RFL 00 1 N E Y Y N Y Y Y Referer Log file

In order to turn on the file at the DETAIL time-span, it will be necessary to change the "N" in the DETAIL time-span to a "Y". The "E" in the DAYS time-span must be retained, as the file requires special processing. The new file line in WEBGENIN appears as:

```
o  FILE RFL   00  1   Y E Y Y N Y Y Y Referer Log file.
```

After making these change you must submit the following jobs:

sharedprefix.MICS.CNTL(WEBCGEN)

prefix.MICS.CNTL(COPYLIBU)

prefix.MICS.CNTL(CYCLEGEN)

2. The "Host Name or IP Address" can be up to 254 bytes, if the DNS resolver is active at your site. The data for this field is divided into two CA MICS elements, RFLHSTIP (first 200 bytes) and RFLHSTI2 (remaining 54 bytes). RFLHSTI2 is turned off by default in the file. If you have a "Host Name or IP Address" field longer than 200 bytes, the CA MICS Web Analyzer daily processing will issue warning messages in MICSLOG. If you want to activate

   The following checklist must be completed in order to activate the element and to retrofit databases:

   A. Change the time-span flags as shown below for RFLHSTI2 and the two elements that must follow in sort sequence, RFLLPORT, ENDTS.

   ```
   NAME RFLHSTI2   00  9 9 9 8 7 Host Name or IP Address 2
   NAME RFLLPORT   00  10 10 10 9 8 Client Port
   NAME ENDTS      99  11 0 0 0 0
   ```

   B. Submit sharedprefix.MICS.CNTL(WEBCGEN)

   C. Retrofit all your WLGRFL files in online databases where this file is active. You only need to add the RFLHSTI2 element.

5.3.3.4 WLGRFL Retrieval Examples

This section presents a typical WLGRFL retrieval example.

Show activity of a specific referer for a day.

```
PROC PRINT DATA=&PWEBD..WLGRFL01;
 WHERE RFLHSTIP = '123.45.678.9';
RUN;
```

5.3.4 HTTP Error Messages Log (WLGERL) File

The HTTP Error Messages Log (WLGERL) File is derived from the HTTP Server Error Log. This file records information about the error message and the message description in the text. Complete details of HTTP error messages can be found in the WebSphere HTTP Server documentation.

The following sections describe the file organization, list data elements maintained, and provide usage hints.

1 - WLGERL File Organization
2 - WLGERL Data Elements List
3 - WLGERL Usage Considerations
4 - WLGERL Retrieval Examples

5.3.4.1 WLGERL File Organization

The table below (Figure 5-18) identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	DAY	HOUR	ENDTS	ERLMSGNO	
DAYS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	DAY	HOUR	ERLMSGNO		
WEEKS	SYSID	WEBIP	WEBPORT	YEAR	WEEK
	ZONE	HOUR	ERLMSGNO		
MONTHS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	ZONE	ERLMSGNO			
YEARS	SYSID	WEBIP	WEBPORT	YEAR	ZONE
	ERLMSGNO				
TABLES	N/A				

Generation Date: Tue, May 12, 2009

```
NOTE:  This file was generated with ESSENTIAL=ALL
       option in effect.  All data elements defined in
       the file are generated.
NOTE:  This file was generated with DERIVED=DEFault
       option in effect.  Whether data elements are
       kept on the file on auxiliary storage or not
       is controlled by the complex definition of the
       DERIVED option.
```

Figure 5-18. WLGERL Sort Sequence and Data Granularity

5.3.4.2 WLGERL Elements List

```
_ins000e.html
GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *   Element   Description (LABEL)
 -------   --------  ----------------------------------------


Sequence/Summary Data Elements

 XD....E  DAY      - Day of Month
 XDWMY.E  ERLMSGNO - Error Message Number
 XDW...E  HOUR     - Hour of Day
 XDWM..E  MONTH    - Month of Year
 XDWMY.E  SYSID    - System Identifier
 XDWMY.E  WEBIP    - Web IP Address
 XDWMY.E  WEBPORT  - Web Server Port Number
 XDW...E  WEEK     - Week of Year
 XDWMY.E  YEAR     - Year of Century
 XDWMY.E  ZONE     - Time Zone
```

```
Common Data Elements

 XD....E  DAYNAME  - Name of Day of Week
 XDWMY.E  ENDTS    - End Time Stamp
 XDWMY.E  MICSVER  - CA MICS Version Number
 XDWMY.E  STARTTS  - Start Time Stamp
 XDWMY.E  WEBGROUP - Web Group Name
 XDWMY.E  WEBNAME  - Web Server Name
 XDWMY.E  WEBSNAME - Web Server Short Name
 XDWMY.E  WLGSRVNM - WLE Assigned Server Name

Retained Data Elements

 XDWMY.E  ERLGMTOF - GMT Offset
 XDWMY.E  ERLMSGTX - Error Message Description Text

Accumulated Data Elements

 XDWMY.E  ERLCNT   - Error Log Count
```

5.3.4.3 WLGERL Usage Considerations

The WLGERL file provides a description of WebSphere server error messages with complete message text and the time the error occured. Data in the SMF type 103 record, as recorded in the WEBWPR file, summarizes errors by level and provides some specific error response counts (i.e, 200, 302, 401, 403, 404, 407 and 500). Data in the WLGERL file records all logged errors and allows complete analysis of both system and application level errors.

5.3.4.4 WLGERL Retrieval Examples

This section presents a typical WLGERL retrieval example.

Produce a report showing the count of the top ten error messages with message text.

```
PROC FREQ DATA=&PWEBX.WLGERL01 ORDER=FREQ; TABLES
 ERLMSGNO*ERLMSGTX / OUT=TOP10(DROP=PERCENT) NOCUM NOPRINT
 NOPERCENT;
RUN;

OPTIONS OBS=10;

PROC PRINT DATA=TOP10;
RUN;
```

5.3.5 WLE Configuration (WLGCFG) File

The WLE Configuration (WLGCFG) File is derived from the WLE utility configuration file. The key information shows the complete file path for each log file that is defined in the WLE configuration file for processing.

The source of these records is the Web Log Extractor utility configuration file. The configuration record is generated each time the WLE utility is run and is recorded as subtype 9.

The WLGCFG file is known as a 'spin' file. Unlike, most other files, there is only one DETAIL cycle. The WLGCFG file will contain information on the last 'n' different configuration observed for each web server. The CFG limit parameter as specified in WEBOPS (see section 7.3.2 of this guide) controls the number 'n' of configuration observations that will be stored in the DETAIL timespan of the WLGCFG file

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WLGCFG File Organization
2 - WLGCFG Data Elements List
3 - WLGCFG Usage Considerations
4 - WLGCFG Retrieval Examples

5.3.5.1 WLGCFG File Organization

The table below (Figure 5-19) identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity
DETAIL	SYSID WEBIP WEBPORT YEAR MONTH DAY HOUR ENDTS
DAYS	N/A
WEEKS	N/A
MONTHS	N/A
YEARS	N/A
TABLES	N/A

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.
NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-19. WLGCFG Sort Sequence and Data Granularity

5.3.5.2 WLGCFG Elements List

```
_ins000e.html

GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *  Element    Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 X.....E  DAY      - Day of Month
 X.....E  ENDTS    - End Time Stamp
 X.....E  HOUR     - Hour of Day
 X.....E  MONTH    - Month of Year
 X.....E  SYSID    - System Identifier
 X.....E  WEBIP    - Web IP Address
 X.....E  WEBPORT  - Web Server Port Number
 X.....E  YEAR     - Year of Century

Common Data Elements

 X.....E  DAYNAME  - Name of Day of Week
 X.....E  MICSVER  - CA MICS Version Number
 X.....E  STARTTS  - Start Time Stamp
 X.....E  WEBGROUP - Web Group Name
 X.....E  WEBNAME  - Web Server Name
 X.....E  WEBSNAME - Web Server Short Name
 X.....E  WEBSRVRT - Directory for Server Root
 X.....E  WEEK     - Week of Year
 X.....E  WLGSRVNM - WLE Assigned Server Name
 X.....E  ZONE     - Time Zone

Retained Data Elements

 X.....E  CFGACPTH - Access Log Path
 X.....E  CFGAFPTH - FRCA (Cache) Log Path
 X.....E  CFGAGENT - Agent Log Path
 X.....E  CFGERROR - Error Log Path
 X.....E  CFGFPROX - Proxy Cache Log Path
 X.....E  CFGPROXY - Proxy Log Path
 X.....E  CFGREFER - Referer Log Path
```

5.3.5.3 WLGCFG Usage Considerations

The WLGCFG file is only available at the DETAIL timespan, thereby limiting the number of observations and assisting in managing the size of your database(s). This file provides information on the location of log inputs, as specified to the CA MICS Web Log Extractor. The primary benefit of WLGCFG is to analyze a web server's path name changes that may have occur over time.

This files contains only one record for every web server whose data was passed to Web Log Extractor (WLE) utility. The only time you will see another record for a web server is when any one of its log file path name is changed. The file can contain, by default, maximum of seven most recent observation for a web server. During the DAY052 processing of subtype 09 observations, a comparison of the log file path name is made. If the path names are identical to already known path name for a web server, the observation is dropped from further processing. The benefit of dropping the most recent as oppose to the oldest observation is that you can determine when a specific configuration was set.

To keep an observation only when configuration changes are detected allows to keep the file volume very low even when you increase the default for CFGLIMIT parameter from seven to some higher number.

5.3.5.4 WLGCFG Retrieval Examples

This section presents a typical WLGCFG retrieval example.

Print server configuration change activity.

```
PROC PRINT DATA=&PWEBX..WLGCFG01;
RUN;
```

5.3.6 HTTP Access Method Summary (WLGAMS) File

The HTTP Access Method Summary (WLGAMS) File is derived from the HTTP Server Access Log (WLGSAL) File, making it event driven data. This file is only generated at DAYS and higher timespans. It contains summarized data at the access method and status code level.

The HTTP Access Method Summary data can be useful to track occurrences of status codes (302, 402, etc.) for specific access methods. Large values of particular status codes can reflect a problem with the accessed web page. This file helps to monitor the health, throughput, and activity of your Web Server.

The following sections describe the file's organization, list data elements maintained, and provide usage hints.

1 - WLGAMS File Organization
2 - WLGAMS Data Elements List
3 - WLGAMS Usage Considerations
4 - WLGAMS Retrieval Examples

5.3.6.1 WLGAMS File Organization

The table below (Figure 5-20) identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity					
DETAIL	N/A					
DAYS	SYSID	WEBIP	WEBPORT	YEAR	MONTH	
	DAY	HOUR	WLGLTYPE	AMSREQME	AMSREQST	
WEEKS	SYSID	WEBIP	WEBPORT	YEAR	WEEK	
	ZONE	HOUR	WLGLTYPE	AMSREQME	AMSREQST	
MONTHS	SYSID	WEBIP	WEBPORT	YEAR	MONTH	
	ZONE	WLGLTYPE	AMSREQME	AMSREQST		
YEARS	SYSID	WEBIP	WEBPORT	YEAR	ZONE	
	WLGLTYPE	AMSREQME	AMSREQST			
TABLES	N/A					

Generation Date: Tue, May 12, 2009

```
NOTE:  This file was generated with ESSENTIAL=ALL
       option in effect.  All data elements defined in
       the file are generated.
NOTE:  This file was generated with DERIVED=DEFault
       option in effect.  Whether data elements are
       kept on the file on auxiliary storage or not
       is controlled by the complex definition of the
       DERIVED option.
```

Figure 5-20. WLGAMS Sort Sequence and Data Granularity

5.3.6.2 WLGAMS Elements List

```
_ins000e.html

GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *   Element   Description (LABEL)
 -------   --------  ----------------------------------------


Sequence/Summary Data Elements

 .DWMY.E  AMSREQME - Request Method (GET, POST, etc.)
 .DWMY.E  AMSREQST - Request Status
 .D....E  DAY      - Day of Month
 .DW...E  HOUR     - Hour of Day
 .DWM..E  MONTH    - Month of Year
 .DWMY.E  SYSID    - System Identifier
 .DWMY.E  WEBIP    - Web IP Address
 .DWMY.E  WEBPORT  - Web Server Port Number
 .DW...E  WEEK     - Week of Year
 .DWMY.E  WLGLTYPE - Log File Type
 .DWMY.E  YEAR     - Year of Century
 .DWMY.E  ZONE     - Time Zone
```

```
Common Data Elements

 .D....E  DAYNAME  - Name of Day of Week
 .DWMY.E  ENDTS    - End Time Stamp
 .DWMY.E  MICSVER  - CA MICS Version Number
 .DWMY.E  STARTTS  - Start Time Stamp
 .DWMY.E  WEBGROUP - Web Group Name
 .DWMY.E  WEBNAME  - Web Server Name
 .DWMY.E  WEBSNAME - Web Server Short Name
 .DWMY.E  WLGSRVNM - WLE Assigned Server Name

Accumulated Data Elements

 .DWMY.E  AMSACCNT - FRCA (Cache) Access Log Count
 .DWMY.E  AMSACNT  - Server Access Log Count
 .DWMY.E  AMSBYTET - Bytes Transferred
 .DWMY.E  AMSKBYTT - KiloBytes Transferred
 .DWMY.E  AMSMBYTT - MegaBytes Transferred
 .DWMY.E  AMSPCCNT - Proxy Cache Access Log Count
 .DWMY.E  AMSPCNT  - Proxy Access Log Count
 .DWMY.E  AMSTOCHT - Total Cache Hits
 .DWMY.E  AMSTOPHT - Total Proxy Hits
 .DWMY.E  AMSTOTHT - Total Hits

Minimum Data Elements

 .DWMY.E  AMSMNBTT - Min Bytes Transferred

Maximum Data Elements

 .DWMY.E  AMSMXBTT - Max Bytes Transferred

Derived Data Elements

 .DWMY.E  AMSAVABT - Average Bytes Transferred Per Access
 .DWMY.E  AMSAVCBT - Avg Bytes Transferred Per Cache Access
 .DWMY.E  AMSAVPBT - Avg Bytes Transferred Per Proxy Access
 .DWMY.E  AMSAVPCT - Avg Bytes Transferred Per Proxy Cache
 .DWMY.E  AMSPCCHT - Percent Total Cache Hits
 .DWMY.E  AMSPCPCH - Percent Proxy Cache Hits
 .DWMY.E  AMSPCTPH - Percent Total Proxy Hits
```

5.3.6.3 WLGAMS Usage Considerations

The WLGAMS file can be used to classify applications by the methods employed. This file can be used to track errors (status code). The WLGAMS file is derived from the WLGSAL DETAIL observations. The WLGAMS file is not available, nor can it be obtained, at the DETAIL timespan.

5.3.6.4 WLGAMS Retrieval Examples

This section presents a typical WLGAMS file retrieval example.

```
   List server access method summary by the hour;

PROC PRINT DATA=&PWEBD..WLGAMS01;
 ID WEBIP WEBPORT;
 VAR MONTH DAY HOUR AMSREQME AMSACNT AMSACCNT AMSPCNT
     AMSPCCNT;

RUN;
```

5.3.7 HTTP Access Path Summary (WLGAPS) File

The HTTP Access Path Summary (WLGAPS) File is derived from the HTTP Access Log (WLGSAL) File, thus is event driven. This file is only generated at the DAYS and higher time-spans. It contains summarized data at the access path level.

The HTTP Access Path Summary data can be useful in tracking most frequently accessed resources. You can determine the frequency of requests satisfied in cache and determine whether HTTP Server resources should be reconfigured to support high activity web pages. The HTTP Access Path Summary File helps you to monitor the health, throughput, and activity of your Web Server.

```
   NOTE:  The path name can contain symbolic variables and
          parameter values passed by the client to the
          invoked application.  These symbolic variables and
          parameter values, if present, immediately follow
          the actual path name.

The following sections describe the file's organization, list
data elements maintained, and provide usage hints.

    1 - WLGAPS File Organization
    2 - WLGAPS Data Elements List
    3 - WLGAPS Usage Considerations
    4 - WLGAPS Retrieval Examples
```

5.3.7.1 WLGAPS File Organization

The table below (Figure 5-17) identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each site when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your organization.

Timespan	Level of Data Granularity				
DETAIL	N/A				
DAYS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	DAY	HOUR	WEBSRVRT	APSPATH	
WEEKS	SYSID	WEBIP	WEBPORT	YEAR	WEEK
	ZONE	HOUR	WEBSRVRT	APSPATH	
MONTHS	SYSID	WEBIP	WEBPORT	YEAR	MONTH
	ZONE	WEBSRVRT	APSPATH		
YEARS	SYSID	WEBIP	WEBPORT	YEAR	ZONE
	WEBSRVRT	APSPATH			
TABLES	N/A				

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=DEFault option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-17. WLGAPS Sort Sequence and Data Granularity

5.3.7.2 WLGAPS Elements List

```
_ins000e.html

GENERATION DATE: Tue, May 12, 2009

Note:  Essential data elements are identified by an "E"
       under the Timespan asterisk (*) column.

  Time-    Data      Data Element
  Span *   Element   Description (LABEL)
 -------  --------   ----------------------------------------


Sequence/Summary Data Elements

 .DWMY.E  APSPATH  - Path to File
 .D....E  DAY      - Day of Month
 .DW...E  HOUR     - Hour of Day
 .DWM..E  MONTH    - Month of Year
 .DWMY.E  SYSID    - System Identifier
 .DWMY.E  WEBIP    - Web IP Address
 .DWMY.E  WEBPORT  - Web Server Port Number
 .DWMY.E  WEBSRVRT - Directory for Server Root
 .DW...E  WEEK     - Week of Year
 .DWMY.E  YEAR     - Year of Century
 .DWMY.E  ZONE     - Time Zone

Common Data Elements

 .D....E  DAYNAME  - Name of Day of Week
 .DWMY.E  ENDTS    - End Time Stamp
 .DWMY.E  MICSVER  - CA MICS Version Number
 .DWMY.E  STARTTS  - Start Time Stamp
 .DWMY.E  WEBGROUP - Web Group Name
 .DWMY.E  WEBNAME  - Web Server Name
 .DWMY.E  WEBSNAME - Web Server Short Name
 .DWMY.E  WLGSRVNM - WLE Assigned Server Name
```

```
Accumulated Data Elements

 .DWMY.E  APSACCNT - FRCA (Cache) Access Log Count
 .DWMY.E  APSACNT  - Server Access Log Count
 .DWMY.E  APSBYTET - Bytes Transferred
 .DWMY.E  APSKBYTT - KiloBytes Transferred
 .DWMY.E  APSMBYTT - MegaBytes Transferred
 .DWMY.E  APSPCCNT - Proxy Cache Access Log Count
 .DWMY.E  APSPCNT  - Proxy Access Log Count
 .DWMY.E  APSTOCHT - Total Cache Hits
 .DWMY.E  APSTOPHT - Total Proxy Hits
 .DWMY.E  APSTOTHT - Total Hits

Minimum Data Elements

 .DWMY.E  APSMNBTT - Min Bytes Transferred

Maximum Data Elements

 .DWMY.E  APSMXBTT - Max Bytes Transferred

Derived Data Elements

 .DWMY.E  APSPCCHT - Percent Total Cache Hits
 .DWMY.E  APSPCPCH - Percent Proxy Cache Hits
 .DWMY.E  APSPCTPH - Percent Total Proxy Hits
```

5.3.7.3 WLGAPS Usage Considerations

```
The WLGAPS file is used to find the most popular or commonly
hit web pages and is derived from the WLGSAL file.  The
WLGAPS file is not available at the DETAIL timespan.
```

5.3.7.4 WLGAPS Retrieval Examples

This section presents a typical WLGAPS retrieval example.

List the top twenty web pages accessed for each server.

```
PROC SORT DATA=&PWEBD..WLGAPS01 OUT=WLGAPS01;
 BY WEBIP WEBPORT APSPATH;

PROC SUMMARY DATA=WLGAPS01 NWAY;
  CLASS WEBNAME APSPATH ;
  VAR APSACNT APSACCNT APSPCNT APSPCCNT;
  OUTPUT OUT=NEW SUM=APSACNT APSACCNT APSPCNT APSPCCNT;
RUN;

OPTIONS OBS=20;

PROC PRINT;
  ID WEBNAME APSPATH ;
  VAR APSACNT APSACCNT APSPCNT APSPCCNT;
RUN;
```

Chapter 6: DATA SOURCES

IBM's WebSphere Application Server for z/OS or OS/390 is not a product; it is a family of products that is comprised of the IBM Web Application environment. The IBM WebSphere Application Server operates across multiple platforms, including NT, AIX, HP, AS/400, z/OS, and OS/390. Operating under a z/OS or OS/390 platform, the CA MICS Web Analyzer uses data collected from the WebSphere Application Server for z/OS or OS/390 to monitor and improve performance.

The CA MICS Web Analyzer supports the following data sources that can be used together or separately:

- SMF type 103 records (generated by IBM's WebSphere Application Server for z/OS and OS/390, HTTP Server)

- SMF type 120 records (generated by IBM's WebSphere Application Server for z/OS and OS/390)

- WebSphere Server logs (that is, Server Access Log, FRCA Access Log, Proxy Access Log, FRCA Proxy Log, Error Log, Agent Log, Referer Log, and Server Configuration Log)

Note: See section 6.2.1.1 for information on how to activate configuration and/or performance record logging for IBM's WebSphere Application Server for z/OS and OS/390. Refer to IBM's WebSphere Application Server for OS/390 HTTP Server Planning, Installing, and Using Guide for additional details about SMF record type 103 and IBM's WebSphere Application Server for z/OS and OS/390 Operations and Administration Guide for additional details about SMF record type 120.

The CA MICS Web Analyzer uses information logged by IBM's WebSphere Application Server for z/OS and OS/390 and its HTTP Server to monitor your WEB activity. It uses the SMF 103 and SMF 120 records and HTTP Server Log records measurement data to produce files in the Web Information Area (WEB) and Web Log Information Area (WLG) of the CA MICS database. Ask your system administrator if these records are available from your WebSphere Application Server.

The records used by the CA MICS Web Analyzer are discussed later in this chapter. For additional information about data sources, refer to the WebSphere documentation and IBM's "HTTP Server Planning, Installing, and Using" guide.

This section contains the following topics:

6.1 Data Collector Considerations

Before installing the CA MICS Web Analyzer, the CA MICS System Administrator must address the data collection considerations. These include record type identification, configuring your Web Server to collect data, and ensuring availability of WebSphere HTTP Server log data sets. Types of Web Server Configuration modes are discussed below:

1 - WLM Interaction with WebSphere Servers

6.1.1 WLM Interaction with WebSphere Servers

There are three modes that can be used in configuring the WebSphere Application Server for z/OS and OS/390. These configurations are identified as standalone, scalable, and/or multiple servers.

- Standalone mode involves the use of an HTTP or web server address space (a/s) that interfaces with TCP/IP and serves up HTML web pages. Application code, in the form of GWAPI or CGI programs, execute as processes or threads in the same address space. The Workload Manager (WLM) views the web server as a single address space.

- Scalable mode involves an HTTP or web server, known by WLM as the Queue Manager, and one or more Application servers, known to WLM as queue servers. WLM will start and maintain one or more queue server address spaces, based on unique workload definition goal objectives. This mode permits large scale web-based applications to be managed according to client defined business objectives.

- Multiple server mode is a variation of both cases above, whereby multiple web servers are created and utilized by a process known as TCP/IP port sharing. The port-sharing option permits multiple servers to listen on a common IP port for incoming traffic. This option can be applied to both the standalone and scalable cases, depending on need, and could even be combined into a complex case involving multiple solutions for a number of discrete workloads, such as an Internet Service Provider (ISP) environment.

6.2 Record Descriptions

The records shown in Figures 6-1 and 6-2 are used as input by the CA MICS Web Analyzer. These figures show when each record is written and the content of the records.

The WebSphere Application Server for z/OS and OS/390 generates the SMF type 120 record. The HTTP Server generates SMF type 103 records, and is also capable of generating HTTP Server Log Records. These records are pre-processed through the CA MICS Web Log Extractor utility.

The type 103 records provide configuration (subtype 01) and performance (subtype 02) information for IBM's WebSphere Application Server for z/OS and OS/390, HTTP Server. These records are written on a user-defined interval with a default value of 15 minutes, which is initially set in the configuration file.

Additional detailed information about SMF type 103 records can be found in IBM's documentation for the WebSphere Application Server.

The type 120 record provides Server Activity (subtype 01) and Server Interval (subtype 03) information for IBM's WebSphere Application Server for z/OS and OS/390. Server activity records are written as each activity within a server is completed. Server interval records are written at user-defined intervals, with a default value of 60 minutes (one hour). Additional records supported by the CA MICS Web Analyzer Option are the subtype 6 (J2EE Container Interval) and subtype 8 (WebContainer Interval) records.

The SMF type 120 subtype 9 record provides a consolidated set of server activity statistics. Generated for each transaction request, this subtype 9 record contains most of the information provided by the other record type 120 subtypes, as well as additional information not found in the other subtypes. For example, the SMF type 120 subtype 9 record contains information about the use of IBM specialty CPU engines.

One subtype 9 record is created for every external web application request, or internal request, when a controller communicates with a servant region.

A description of these records can be found in the following subsections:

1 - SMF Record Types
2 - HTTP Server Log Record

6.2.1 SMF Type Record

Two types of SMF records are generated by WebSphere Application Server for z/OS and OS/390. WebSphere Application Server generates SMF type 120 records and the HTTP Server generates SMF type 103 records. The following sections discuss these types of SMF records in detail:

1 - SMF Type 103 Record
2 - SMF Type 120 Record

6.2.1.1 SMF Type 103 Record

The record shown in Figure 6-1 is used by the CA MICS Web Analyzer to build the files in the Web Information Area. The figure shows when each record is written and what each record contains.

This type of record is generated by the WebSphere Application Server for z/OS and OS/390. The SMF Type 103 record contains two subtypes. Below is a brief description of the SMF Type 103 record contents and when the record is written.

Record Type	Subtype	Written	Contents
SMF 103	1	When the server is initialized.	Configuration
SMF 103	2	For every interval defined through the SMFrecordinginterval directive in the server configuration file	Performance

Figure 6-1. Web Record Descriptions

```
In order to enable IBMs WebSphere Application Server for
OS/390 to log SMF type 103 records, you must code the
following directives in the HTTPD.CONF configuration file:

SMF                   all/config/perf/none

     where   "all"    log both subtype 01 and 02 records.

             "config" log subtype 01 records.

             "perf"   log subtype 02 records.

             "none"   DEFAULT - no SMF record logging

SMFRecordingInterval  hh:mm

               where  hh:mm  is the SMF recording interval
                             time in hour and minutes.
                             DEFAULT time is 00:15 unless
                             changed in the HTTPD.CONF.  The
                             SMFRecordingInterval should be
                             evenly divisible into an hour
                             and be less than 1 hour (i.e, 5,
                             10, 15, 20, or 30 minutes).

Further details on this and other directives are available in
the WebSphere HTTP Server guide.
```

6.2.1.2 SMF Type 120 Record

The record shown in Figure 6-2 is used by the CA MICS Web Analyzer to build the files in the Web Information Area. The figure shows when each record is written and what each record contains.

This type of record is generated by the WebSphere Application Server for z/OS. Below is a brief description of the SMF Type 120 record contents.

Record Type	Subtype	Written	Contents
SMF 120	1	When an activity within a server completes.	Server activity
SMF 120	3	For every installation-defined SMF interval.	Server activity in an interval
SMF 120	6	For every installation-defined SMF interval.	EJB Container activity in an interval
SMF 120	8	For every installation-defined SMF interval.	Web Container activity in an interval
SMF 120	9*	For every external or internal request activity.	Web Request activity

Figure 6-2. SMF Type 120 Record Descriptions

For information on how to enable IBM's WebSphere Application Server for z/OS to log SMF type 120 records, refer to the IBM documentation for WebSphere Application Server.

IBM introduced the SMF type 120 subtype 9 Web Request Activity record with WAS v7. This record was designed to unify the performance metrics for each web request. Previously, users found it difficult to correlate and piece together request performance metrics using the other SMF type 120 record subtypes. The Subtype 9 record also contains additional performance metrics not found in the other

subtypes.

The Subtype 9 record contains many sections, some of which are user configurable. User configuration was provided to allow users to strike a balance between the collection of detailed performance metrics, and the overhead required for collection. Users may activate some sections for short periods of time to study performance or characteristics of web requests.

The following chart list the sections, and identifies which sections are configurable, (meaning they may or may not be present in the raw data), and the target CA MICS file(s) where some or all of the metrics will be found.

Subtype 9 section	Occurrences	Configurable	File(s)
Platform neutral Server Information	1	No	WEBWRA, WCB
z/OS Server Information	1	No	WEBWRA, WCB
Platform Neutral Request Info	1	No	WEBWRA, WCB
z/OS Request Information	1	No	WEBWRA
z/OS formatted timestamps	0-1	Yes	None
Network Data	0-1	No	WEBWRA
Classification data	0-5	No	WEBWRA
Security data	0-3	Yes	WEBWRA
CPU Usage Breakdown	0-30	Yes	WEBWCB
User Data	0-5	Yes	WEBWUD
Asynchronous			Not

Activity	1	No	Supported

Note: Request identification data elements are found in multiple files. For merging files together at the detail level, use the Enclave Token (WEBCY) value to match observations for a particular request.

6.2.2 HTTP Server Log Records

The record shown in Figure 6-3 is used by the CA MICS Web Analyzer Option. The figure shows when the record is written and what each record contains.

HTTP Server Log records are generated by the HTTP Web Server. The server records activity in various log files each time a web page is accessed. This makes the data in log files event driven. The Web Server can be configured to log many different types of activities. A maximum of eight log files can be activated and generated. Each night at midnight the server closes the current logs and creates new logs for the new day. Following is a list of the log files:

Server Access Log	Contains an activity of requests made to the server.
FRCA Access Log	Contains activity of requests served by the Fast Response Cache Accelerator (FRCA).
Proxy Access Log	Contains activity of requests processed by the proxy server.
FRCA Proxy Log	Contains activity of requests processed by the proxy server's cache.
Agent Log	Contains the browser name used to access the requested resource.

Referer Log	Identifies the web page that referred (linked to) the requested resource.
Error Log	Contains errors encountered by the server's client.
CGI Error Log	Contains standard error output (stderr) from CGI programs.

The CA MICS Web Analyzer Option provides a WEB Log Extractor (WLE) utility to preprocess and reformat the log files. As many as seven log files from each server are accepted for processing in the DAY052 step.

The Web Log Extractor (WLE) utility uses its config file to determine the specific log files for specified servers to process and reformat. You must specify each server and the path name for its log files. The log files and their corresponding SMF-like records generated by the WLE utility are shown in Figure 6-3.

Record name	Written	Record contents
WLE Record	For every log file that is processed through the WLE utility and for every server processed.	Server Access Log (Subtype 1) FRCA Cache Access Log (Subtype 2) Proxy Access Log (Subtype 3) Proxy Cache Log (Subtype 4) Agent Log (Subtype 5) Referer Log (Subtype 6) Error Log (Subtype 7) WLE Configuration (Subtype 9)

NOTE: The data will be logged to the Proxy logs (access and cache) or the FRCA and Server Access logs, never to both.

Figure 6-3. HTTP Server Log Record

6.3 CA MICS and CA SMF Director Interface

CA SMF Director is an SMF management product with special features that can be used to optimize CA MICS processing. CA SMF Director can significantly reduce operational overhead by creating content specific files while simultaneously creating an SMF historical archive during the SMF dump process.

CA MICS DAILY and incremental update operational jobs can be modified to take advantage of this product, provided your site is licensed for and has CA SMF Director installed in your environment.

CA SMF Director provides functions that extract SMF data at dump time or from previously archived SMF files. These functions eliminate the need for any external utilities used for preprocessing of the SMF dump tape for data extraction prior to execution of the DAILY job. In addition, the DAYSMF step of the CA MICS DAILY job may no longer be required. For details on how to eliminate the DAYSMF step, see section 5.10, Removing the DAYSMF Step from the DAILY Job of the PIOM guide.

CA SMF Director SPLIT and EXTRACT statement operands provide a way for you to create content specific files to meet your requirements. The main functions of the operands include the ability to:

- Select system identifiers
- Select or exclude SMF record types and subtypes
- Use conditional statements for granularity
- Define time boundaries for SMF data

To use the SPLIT function of CA SMF Director for CA MICS components, one or more SPLIT statements must be coded. The split is performed at SMF dump time and the files can then be used as input to products in one or more units. The same results can be accomplished by using the EXTRACT function of CA SMF Director, which retrieves data from the previously created history files. For more information on the SPLIT and EXTRACT functions, see the CA SMF Director User Guide at http://ca.com/support.

A complete set of SMF record types and subtypes, for each CA MICS component that inputs SMF data, can be found in sharedprefix.MICS.PARMS(cccSMFD), where ccc is the three-character product identifier. Each PARMS member lists the available SMF record types and subtypes for that component in a format suitable for use in CA SMF Director SPLIT and EXTRACT statements. These members are provided as examples only; the CA SMF Director JCL should not reference sharedprefix.MICS.PARMS(cccSMFD).

Chapter 7: PARAMETERS

This chapter describes how to define the parameters required to install the CA MICS Web Analyzer. Use this chapter as a detailed reference in conjunction with the PIOM.

Defining CA MICS parameters requires an understanding of your site requirements and the ability to translate that understanding into the appropriate CA MICS parameters. You must do the following:

- Arrive at a number of policy decisions.
- Fill out several worksheets.
- Translate the worksheet entries into the corresponding CA MICS parameter library member entries.

These activities represent the major portion of the product installation process. While worksheets are provided, they are optional; it is mandatory however to ensure that product parameters are coded.

This chapter focuses on considerations that are unique to the CA MICS Web Analyzer. Chapters 2 and 3 of the PIOM document the mechanics of the CA MICS installation process and include checklists that describe each installation step.

If you have any questions, contact CA Technical Support.

This section contains the following topics:

7.1 Environmental Considerations

Before coding product parameters, you need to know about the Web Server environment(s) at your installation. Prior to specifying parameters for the product:

- Examine existing web server applications, and internal ownership rules at your site. You may want to define ownership based on file path or other criteria.

- Review current CA MICS parameter specifications:

  - Review database unit specifications to determine which unit or units should include Web server data.

  - Review the SYSID parameter to determine whether it conflicts with the Web server data. Your site's SYSID definition is located in the prefix.MICS.PARMS data set for each database unit.

  - Review the CA MICS ZONE parameter to ensure that it reflects the variations in Web server data. This parameter is located in the prefix.MICS.PARMS data set for each database unit.

  - Examine existing accounting standards for other CA MICS components, if applicable, to ensure that the CA MICS Accounting and Chargeback Product aggregates information consistently.

The CA MICS Web Analyzer processes input data from the SMF type 103 record and optionally from the WebSphere Application Server for OS/390 log data. All input data is processed by the DAY052 step in the CA MICS DAILY job as part of the same process. By reviewing the MICSLOG output of the DAY052 step of the CA MICS DAILY job, you can determine exactly what SMF records and what WLE utility records were present in the input data stream for each Web Server.

Following the CA MICS standards, if an installation fails to provide input from ANY source, the DAY052 step will fail with a U300 ABEND. When you expect this problem, but must run the DAILY job anyway, code SYSPARM=NODATA on the EXEC statement of the DAY052 step to prevent the ABEND.

There is no parameterized provision in CA MICS to cause a similar ABEND when data from one of the required sources (SMF or Log) is missing but data from the other source is present. If this occurs at your installation, consider coding the _USRIHL exit to examine the Input History Log File from the DETAIL timespan at the end of DAILY processing. This routine, and a sample exit which provides this function, are documented in Chapter 10 of this guide.

7.2 Complex Level Parameters

Complex level parameters describe the CA MICS Web Analyzer to the CA MICS system. These parameters are described in the following subsections:

1 - Analyzer Definition Statements (WEBGENIN)
2 - Account Code Specification (WEBACCT)
3 - Account Code Exit Routine (WEBACRT)

7.2.1 Analyzer Definition Statements (WEBGENIN)

A Generation Definition Statement member is provided for the CA MICS Web Analyzer in the WEBGENIN member of the sharedprefix.MICS.GENLIB library. WEBGENIN contains the component definition input to the component generation job in sharedprefix.MICS.CNTL(WEBCGEN). Review this member to check the component definitions and make any desired changes as described below. For more information on the use and modification of Generation Definition Statements, see section 4.2.4 of the System Modification Guide.

Activating Account Codes

By default, the COMP statement keyword is set to ACCT. Accounting changes should be coded in the sharedprefix.MICS.PARMS(WEBACCT and WEBACRT) parameter members discussed later in this chapter. Finally, you must run the WEBCGEN job to make these changes effective.

```
OPTION Statement
---------------

The OPTION statement indicates which groups of data elements
are going to be kept in the database.

The valid keywords are:

  WLG   -  WLE Log Utility Data
  NOWLG -  No WLE Log Utility Data

You must specify one keyword from the pair.  In order to
improve performance, the code to process log data will not be
executed when NOWLG is specified.
```

7.2.2 Account Code Specification (WEBACCT)

```
Account code data elements, a part of the file's sort and
summarization key, have names in the form WEBACTx, where x is
a number from 1 through 9.  If you activate the account code
support, the CA MICS Web Analyzer uses these WEBACTx elements
to associate monitor data with the responsible individual or
group.

The Server Access Log File (WLGSAL) in the CA MICS Web
Analyzer supports account codes.

The account code parameter, sharedprefix.MICS.PARMS(WEBACCT),
defines how many account code fields are maintained in the
CA MICS Web Analyzer's files. (CA MICS supports from one
through nine account codes).  In addition to defining the
number of fields, the WEBACCT parameter defines the length of
each field and its contents (label).

The parameters specified in the WEBACCT member of the
sharedprefix.MICS.PARMS data set define the account field
structure.  To assign values to the account fields, you need
to code an account code exit routine, WEBACRT, described in
Section 7.2.3.

Once you have defined the account codes and their derivation,
you need to activate them by setting an option in the
sharedprefix.MICS.GENLIB(WEBGENIN) member.  Section 7.2.1
describes how to activate the account codes.
```

PREPARATION

Each installation has its own method for associating commands and users with the projects or departments within the organization. Before defining the account codes, you should investigate your installation's accounting standards, in order to:

- Identify the coding system (e.g., your cost center coding system for identifying the division, department, project and employee).

- Identify how the codes are specified. For example, the division may be determined by part of the user ID.

- Identify if and how the codes are verified to ensure that they correspond to a valid definition. We recommend that you validate all account codes and assign unidentified or invalid account codes to a special installation overhead account code (see the documentation of the WEBACRT exit). This approach provides two benefits: it will filter out invalid codes from inclusion in the database, requiring less DASD space for storing the data in the WEB and WLG Information Area files; and you can easily see how much of this unidentifiable activity is taking place.

Consider the following in determining the number of account fields required to meet your installation's reporting and analysis needs:

- Account fields are part of the file keys for the files that support them. As such, at least one record is generated for each combination of values. (More than one record may be generated because other fields also make up the file keys.) A greater number of account fields means that the database will be larger than if fewer account fields are defined, but reporting and analysis can be performed in greater detail.

- If you anticipate a future need to expand the account code structure, establish an extra account code now to eliminate the need to retrofit the database later.

DEFINING THE WEB ACCOUNT CODE PARAMETER

Once prepared, code the contents of the worksheet in the data set sharedprefix.MICS.PARMS(WEBACCT). Observe the following conventions when coding:

- Blank statements are permitted. Comments are coded by beginning the statement with an asterisk (*).
- Account levels must be specified in ascending order, starting with 1. You can specify up to nine account levels, with no gaps between the numbers.
- The statements are free-form but positional.

SAMPLE WEBACCT MEMBER

The sample WEBACCT member is distributed to contain:

```
* SAMPLE ACCOUNT CODE SPECIFICATION
     1 10 'DIVISION'
     2 10 'DEPARTMENT'
     3 10 'SERVER APPLICATION'
```

This sample uses only three account codes: DIVISION, DEPARTMENT, and SERVER APPLICATION. Each account code is ten bytes long.

SPECIFYING THE TIMESPAN MASK OPTION

The timespan mask option is used to deactivate the account code in specified timespans. Specifying a timespan mask requires coding the account code statement using the timespan mask parameter T(......), as follows:

```
          level  T(......)  length  'descriptive title'
```

This example shows the placement and syntax of the timespan mask on the account code statement. Each "." position in the T position represents a timespan, in the order: DETAIL, DAYS, WEEKS, MONTHS, YEARS, and TABLES. For each timespan, a Y indicates that the account level should be active and an N indicates that the account level should be inactive.

This parameter is optional. If it is not coded, the assumed value is T(YYYYYY).

The following rules apply if you code this parameter:

- DETAIL must always be Y. Account codes may not be deactivated in this timespan.
- If DAYS is N, then WEEKS, MONTHS, and YEARS must also be N. If the account code level is inactive in the DAYS timespan, it cannot be active in higher timespans.
- If MONTHS is N, then YEARS must be N. If the account code level is inactive in the MONTHS timespan, it cannot be active in the YEARS timespan.

```
COLLECTING DATA TO SPECIFY ACCOUNT CODES

Use the worksheet in Figure 7-1 to organize the information
needed to code WEBACCT.
+-------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET:  WEB Account Code Specification     |
| PARMS Library Member is WEBACCT                                         |
| Reference:  Section 7.2.2 - CA MICS Web Analyzer Guide                  |
+-------------------------------------------------------------------------+
|                                                                         |
|  ACCOUNT  TIMESPAN  FIELD                                               |
| CODE LEVEL  MASK     LENGTH  DESCRIPTION                                |
|   (1-9)                                                                 |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|     _     T(______)   __    '_____________________________________'    |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
|                                                                         |
+-------------------------------------------------------------------------+
| ....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-------------------------------------------------------------------------+
```

Figure 7-1. WEB Account Code Specification Worksheet

where the fields are:

ACCOUNT CODE LEVEL — the level of importance of the element. Level 1 is the most important field, and level 9 is the least important.

TIMESPAN — account code levels can be deactivated in the

MASK	DAYS, WEEKS, MONTHS, and YEARS timespans. For example, the DETAIL timespan may have account code levels 1, 2, and 3, while the MONTHS timespan has only level 1. Setting account code levels by timespan is accomplished by specifying this additional parameter on the account code statements, as described below.
FIELD LENGTH	the length of the specified element. Valid lengths range from 1 to 30.
DESCRIPTION	the title that describes the account code level. Valid descriptions range from 1 to 40 characters.

7.2.3 Account Code Exit Routine (WEBACRT)

This section explains how to code an account code exit for the CA MICS Web Analyzer. These definitions will be used for every database unit that contains this analyzer.

There are 4 files that contain an account code structure. The Data elements from the raw data are:

File	File Name	Source
WLGSAL	HTTP Access Log File	HTTP Server Log
WEBSAF	WAS Server Activity File	SMF 120 Subtype 1
WEBWRA	Web Request Activity File	SMF 120 Subtype 9
WEBWAA	Web Asynch Activity File	SMF 120 Subtype 9

After you define WEBACCT for the number of account codes you want to carry in the above listed files, code the Account Code Derivation Routine (WEBACRT) in sharedprefix.MICS.PARMS. WEBACRT is a SAS routine that derives the account code data elements during the DAY052 step. For example, if you define three account codes in WEBACCT, then your WEBACRT exit must contain SAS code to derive the data elements WEBACT1, WEBACT2, and WEBACT3. If you do not process the HTTP Server log or the SMF 120 Subtype 1 data, you do not need to include checks for those in the routine.

Every element read from the raw input record is available to the WEBACRT routine whether it is written to the database or not.

Possible elements from the SMF 120 subtype 9 record to base the account code derivations on might include (but not limited to):

WRABT	Controller Jobname
WRACR	Job Name for Dispatch Servant
WEBRTYPE	Web Request Type
WRAEK	Request Origin
USER1-5*	User Data (temporary element)
USERNO1-5	User Data Type (temporary element)

* Temporary elements USER1 through USER5 contain the user data written to the SMF 120 subtype 9 record by a WAS API that possibly identifies some details on what the application is and the type or function of the web request being done by the application, which might be useful in assigning the account code fields. This is a free form

```
  field and might require complex code to pick out the
  specific information needed.  There can be up to 5 of these
  user data sections in the record.  USER1 contains the value
  from the first user data section, USER2, the second, and so
  forth.  Each user data section is identified by a User Data
  Type which is in the corresponding data elements
  USER1-USER5.

You are responsible for testing the accuracy of the exit
routine.  The worksheet for coding the WEBACRT exit is shown
in Figure 7-2.

+-------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET:  WEB Account Code Routine Definition |
|                                                                         |
| PARMS Library Member is WEBACRT                                         |
| Reference Sections: 7.2.3  CA MICS WebSphere Analyzer Guide             |
+-------------------------------------------------------------------------+
| * CHECK INPUT SOURCE                                                    |
| IF FILEID 'WEBWRA' THEN DO;             /* check the input source  */   |
|                                                                         |
|   IF { account data is valid } THEN DO; /* validate account codes  */   |
|                                                                         |
|     WEBACT1=field source 1 ;            /* populate the account    */   |
|     WEBACTn=field source n ;            /* code fields             */   |
|     _____________________________________________________________       |
|     _____________________________________________________________       |
|     _____________________________________________________________       |
|   END;                                                                  |
| END;  /* end if waa fileid  */                                          |
| ELSE IF FILEID 'WEBSAF' THEN DO;        /* check the input source  */   |
|                                                                         |
|   IF { account data is valid } THEN DO; /* validate account codes  */   |
|                                                                         |
|     WEBACT1=field source 1 ;            /* populate the account    */   |
|     WEBACTn=field source n ;            /* code fields             */   |
|     _____________________________________________________________       |
|     _____________________________________________________________       |
|     _____________________________________________________________       |
|   END;                                                                  |
| END;  /* end if saf fileid  */                                          |
| ELSE DO;                                                                |
|                                                                         |
|     WEBACT1='overhead category' ;    /* Nothing qualifies, assign  */   |
|     WEBACTn='overhead category' ;    /* to overhead                */   |
|     _____________________________________________________________       |
|     _____________________________________________________________       |
|     _____________________________________________________________       |
|                                                                         |
```

```
| END;                                                                      |
+---------------------------------------------------------------------------+
```

Figure 7-2. WEB Account Code Routine Definition Worksheet

7.3 Unit Level Parameters

This section contains information on the unit level parameters for the CA MICS Web Analyzer:

1 - Web Parameter Generation (WEBPGEN)
2 - Web Processing Options (WEBOPS)
3 - Web IP/PORT Associated Parameters (WEBSLIST)
4 - INPUTRDR and INPUTWEB PARMS Members
5 - Database Space Modeling (DBMODEL)

7.3.1 Web Parameter Generation (WEBPGEN)

The Web Parameter Generation (WEBPGEN) process activates database unit level options. The WEBPGEN also generates SAS macros which are stored in prefix.MICS.USER.SOURCE($WEBMSTR) and skeleton JCL in prefix.MICS.PARMS(WORKWEB). The job contained in prefix.MICS.CNTL(WEBPGEN) must be run when changes are made to the prefix.MICS.PARMS(WEBOPS) member, or prefix.MICS.PARMS(WEBSLIST) member, as defined in the next section.

7.3.2 Web Processing Options (WEBOPS)

```
 This section shows you how to specify the operational
 statements that control CA MICS Web Analyzer processing.


Operational statements are stored in the prefix.MICS.PARMS
cccOPS member, where ccc is the component identifier, and are
incorporated into the CA MICS system by running the
prefix.MICS.CNTL(cccPGEN) job.

************************************************************
*                                                          *
*  NOTE:  CHANGES to prefix.MICS.PARMS(cccOPS) members     *
*         REQUIRE EXECUTION of prefix.MICS.CNTL(cccPGEN)   *
*         to take effect.                                  *
*                                                          *
*         In addition, any change to parameters that       *
*         impact the DAILY operational job JCL such as,    *
*                                                          *
*         o  changing RESTART NO to RESTART YES,           *
*                                                          *
*         o  WORK parameter changes when RESTART NO is in  *
*            effect,                                       *
*                                                          *
*         o  Specifying TAPEfff (if this product supports  *
*            a DETAIL level TAPE option),                  *
*                                                          *
*         o  or changes to prefix.MICS.PARMS(INPUTccc),    *
*                                                          *
*         will require regeneration of the DAILY job by    *
*         executing prefix.MICS.CNTL(JCLGEND) or by        *
*         specifying DAILY in prefix.MICS.PARMS(JCLGENU)   *
*         and executing prefix.MICS.CNTL(JCLGENU).         *
*                                                          *
*         Refer to the checklist (if provided) for updating *
*         cccOPS parameters and running required generation *
*         jobs.                                            *
************************************************************
 The following sections explain each of the statements and
 parameters specified in prefix.MICS.PARMS(WEBOPS). Once they
 are defined and coded, implementation of the specified
 parameter values occur with the successful execution of the
 parameter generation routine, prefix.MICS.CNTL(WEBPGEN), in
 each unit containing the CA MICS Web Analyzer product.

 The optional WEBOPS statements are described in the following
 sections:
```

```
 1 SUSPEND
 2 CFGLIMIT
 3 WCFLIMIT
 4 DELTA
 5 WORK, MULTWORK, and NOMULT
 6 RESTART
 7 Incremental Update
 8 DETAIL Tape Processing
 9 Global and System Level GMT Time Offsets
```

7.3.2.1 SUSPEND Statement

```
The SUSPEND statement is optional and signifies that a
specific number of suspended observations in the WebSphere
Performance Suspend (WEB_WP) will be kept.

Once the suspend limit is exceeded for an observation's data
in days, it is discarded.  Records (observations) in the
WEB_WP file are used to provide continuity between the end of
one day's server data, and the beginning of the next day's
data.  In general, specific records in the WEB_WP file should
only remain in the file for one day.

The SUSPEND statement has the following format:

SUSPEND n

where:

n  = the number of days (daily executions) that suspended
     observations will be kept.

DEFAULT:  7
```

7.3.2.2 CFGLIMIT Statement

The CFGLIMIT statement is optional. A config record is generated by the Web Log Extractor (WLE) utility every time the utility is executed. If the config record has not changed, the new observation is kept and the old one is discarded. Up to seven observations are kept in the WLGCFG file.

The CFGLIMIT statement has the following format:

CFGLIMIT n

where n is the number of unique observations in the WLE Configuration (WLGCFG) file that is kept for each server.

DEFAULT: 7

7.3.2.3 WCFLIMIT Statement

The WCFLIMIT statement is optional. A config record (SMF type 103 subtype 01) is generated by the WebSphere Application Server for OS/390 only when the web server is initialized. Thus, if the server is active for days or weeks, SMF type 103 subtype 01 records will be processed over that time.

Once the limit has been reached, the oldest observations are discarded.

The WCFLIMIT statement has the following format:

WCFLIMIT n

where n is the number of the most recent observations in the WebSphere Configuration (WEBWCF) file that is kept for each server.

DEFAULT: 7

7.3.2.4 DELTA Statement

The DELTA statement is optional. This statement is used to increase the delta array size if needed. This array is used to calculate the delta values of various CPU time values from the SMF 120 subtype 9 records, where they were processed on the same enclave.

Check the MICSLOG message WEB00259I for the array usage during the DAILY update. If the used value is close to the allocated size, the array size should be increased to avoid overflowing the array.

The DELTA statement has the following format:

DELTA nnn

where nnn is the number of array slots the system will allocate during the Daily update processing.

DEFAULT: 500

7.3.2.5 WORK, MULTWORK, and NOMULT Statements

```
WORK
----

This statement is optional.  It enables sites experiencing
either SAS WORK space allocation problems or out of work
space conditions during DAYnnn or INCRnnn (where nnn is the
job step number), daily or incremental update processing, to
allocate multiple WORK files.

You can allocate multiple WORK files for use during the daily
and/or incremental update job step.  The maximum number of
WORK files you can allocate varies by product.  These
additional work files are used in conjunction with the single
work data set allocated by default using the JCLDEF
parameters WORKUNIT and WORKSPACE.

Because the individual space allocation requirement for each
WORK file is typically much smaller, it is more likely to be
satisfied.

To take advantage of multiple WORK files support, edit
prefix.MICS.PARMS(cccOPS) and insert a WORK statement as
shown below:

WORK n data_set_allocation_parameters

where n is the number of WORK data sets


         NOTE:  The default is one (1).
                The maximum is nine (9).

         data_set_allocation_parameters is one or more data
        set allocation parameters (for example, STORCLAS or
        SPACE) separated by spaces.

You can also specify the WORK parameter as the following:

WORK n XXX pppp ssss

where:

  n    is the number of WORK data sets
  XXX  is TRK or CYL
  pppp is the primary allocation
```

```
  ssss is the secondary allocation

Note:  When allocating any number of SAS WORK data sets, be
aware that one additional SAS WORK data set is automatically
allocated to facilitate sorting.  For example, if you
allocate six SAS WORK data sets, you will actually get seven.

If you omit the data_set_allocation_parameters or the WORK
parameter, the work data sets are allocated according to the
values you specified for the WORKUNIT and WORKSPACE
parameters in prefix.MICS.PARMS(JCLDEF).  Use the
data_set_allocation_parameters to override this default,
either to alter the space allocation or to use System Managed
Storage (SMS) parameters to control data set placement and
characteristics.

Note:  If you allocate insufficient space for the WORK data
sets, DAYnnn and/or INCRnnn processing will fail and can only
be restarted from the beginning.

Note:  If internal step restart is active, you can override
the WORK data set allocation parameters at execution-time
using the //PARMOVRD facility.  For more information about
execution-time override of dynamic data set allocation
parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks,
according to SAS LIBNAME statement syntax.  If you need
multiple lines, repeat the WORK keyword on the continuation
line.

WORK accepts the engine/host options documented in the SAS
Companion for the z/OS environment, including STORCLAS, UNIT,
SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important!  Do not specify the DISP parameter.

Example 1:

WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss),RLSE)

where:

  n        - is the number of WORK data sets.
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to 8 characters.
  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
  XXX      - is TRK or CYL.
```

```
  pppp     - is the primary allocation.
  ssss     - is the secondary allocation.
  RLSE     - specifies that free-space should be released
             when the data set is closed.

Example 2:

WORK n XXX pppp ssss

where:

  n        - is the number of WORK data sets.
  XXX      - is TRK or CYL.
  pppp     - is the primary allocation.
  ssss     - is the secondary allocation.

Example 3 (multiple lines):

WORK n STORCLAS=MICSTEMP UNIT=SYSDA
WORK   SPACE=(xxxx,(pppp,ssss),,,ROUND))

where:

  n        - is the number of WORK data sets.
  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.
  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.
  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
  XXX      - is TRK or CYL.
  pppp     - is the primary allocation.
  ssss     - is the secondary allocation.

Note:  Since there is some performance impact when using
multiple WORK files, you should specify the minimum number of
WORK data sets to meet your work space requirements.  As a
start, try incrementing the number gradually beginning from
the default.


WORK Considerations
--------------------

How Much Space Should You Allocate?

o  First Time Implementation of Multiple Work Files

   If this is the first time you are implementing multiple
```

work files for this product in this unit, review prefix.MICS.PARMS(JCLDEF) and find the WORKSPACE parameter. It will resemble this sample statement:

```
WORKSPACE      TRK 500 250
```

The value shows the current SAS WORK space allocation for the unit as a single data set. It also serves as the default value used in the unit's DAYnnn daily update (and/or INCRnnn incremental update) step unless you provide a WORK parameter.

To achieve the equivalent work space allocation of WORKSPACE TRK 500 250 using multiple WORK data sets that will collectively share the work space requirements of the daily and/or incremental update step, you could code either one of these:

```
WORK 2 SPACE=(TRK,(250,125))
```

```
WORK 5 SPACE=(TRK,(100,50))
```

To determine the total work space, multiply the number of WORK files (n) by the primary (pppp) and secondary (ssss) values specified.

Note: To simplify the example, only the SPACE parameter is shown above. You can follow either with data set allocation parameters like UNIT or STORCLAS as required for your site.

o Adjusting Allocation for Existing Multiple WORK Files

If you have previously implemented multiple WORK file support for this product in this unit, and you want to change either the number of WORK files or the space allocations, examine prefix.MICS.PARMS(cccOPS) and find the existing WORK statement.

- If the existing WORK statement only specifies the number of WORK files but does not contain space allocation information as shown below:

  ```
  WORK 5
  ```

  Then each of the multiple WORK files is allocated using the values from the WORKSPACE parameter of prefix.MICS.PARMS(JCLDEF), as described earlier under First Time Implementation of Multiple Work Files.

To increase workspace, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation in the WORKSPACE parameter, or do both.

To decrease workspace, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation in the WORKSPACE parameter, or do both.

You can also elect to explicitly specify the multiple WORK file space allocation by adding the space allocation values directly to the WORK statement. This will remove the link to the prefix.MICS.PARMS(JCLDEF) WORKSPACE parameter for multiple WORK file space allocation. This is recommended as it serves to clearly document, in one place, how multiple WORK files are allocated.

- If the existing WORK statement does include space allocation as shown in the examples below:

  WORK 5 TRK 200 100

  or

  WORK 5 SPACE=(TRK,(200,100)) STORCLAS=MICSTEMP

  Simply change the values to meet your needs.

  If you need more work space, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation (for example, change TRK 200 100 to TRK 250 120), or do both.

  To decrease work space, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation (for example, change TRK 200 100 to TRK 150 80), or do both.

Note: If internal step restart is NOT active (RESTART NO) and you change the WORK parameter, you must:

- Run cccPGEN
- Run JCLGENU for DAILY (to regenerate DAILY) and, if incremental update is enabled, INCRccc

```
When internal step restart is active, (RESTART YES), then,
when you change WORK and run cccPGEN, changes take effect
immediately.  There is no need to run JCLGENU.


SASWORK
-------

This statement is optional.

The WORK DD statement in the CA MICS procedures allocates
a temporary data set where SAS keeps its temporary data
files and other items that SAS uses during processing of
the current job.

By default, the space allocated is defined in the member
prefix.MICS.PARMS(JCLDEF) with the WORKSPACE and WORKUNIT
parameters, then generated into all the JCL procedures for
a given unit.

With the SASWORK statement you have the option to override
this unit-wide definition to specify the space allocation
individually for the current step.

The format of the SASWORK statement is:

SASWORK data_set_allocation_parameters

where data_set_allocation_parameters is one or more data set
allocation parameters (for example, STORCLAS or SPACE)
separated by spaces.

You can also specify the SASWORK parameter as the following:

SASWORK XXX pppp ssss

where:

  XXX  is TRK or CYL
  pppp is the primary allocation
  ssss is the secondary allocation

If you omit the data_set_allocation_parameters or the SASWORK
statement, the WORK data set is allocated according to the
values you specified for the WORKUNIT and WORKSPACE
parameters in prefix.MICS.PARMS(JCLDEF).  Use the
data_set_allocation_parameters to override this default,
either to alter the space allocation or to use System Managed
Storage (SMS) parameters to control data set placement and
```

```
characteristics.

Specify data set allocation parameters, separated by blanks,
according to SAS LIBNAME statement syntax.  If you need
multiple lines, repeat the SASWORK keyword on the
continuation line.

Example:

SASWORK STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to 8 characters.
  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
  XXX      - is TRK or CYL.
  pppp     - is the primary allocation.
  ssss     - is the secondary allocation.


Note:  If you change the SASWORK parameter, you must:

o  Run cccPGEN
o  Run JCLGENU for DAILY (to regenerate DAILY) and, if
   incremental update is enabled, INCRccc




MULTWORK|NOMULT fff fff ... fff
-------------------------------

Since multiple work files usage impacts performance, this
product provides these optional parameters so you can
restrict multiple work files usage to only those files having
excessive space requirements.

Note:  You can only use one of these optional parameters with
the WORK statement, NOT both.

The MULTWORK parameter restricts the use of multiple WORK
files to ONLY those listed after the MULTWORK keyword.

MULTWORK fff fff ... fff

where fff is the unique three character identifier
```

If you need multiple lines, repeat the MULTWORK on the continuation line.

The NOMULT parameter forces the use of multiple WORK files for all files EXCEPT those specified after the NOMULT keyword.

NOMULT fff fff ... fff

where fff is the unique three character identifier

If you need multiple lines, repeat the NOMULT on the continuation line.

The default is

MULTWORK AGL RFL ERL SAL AMS APS WPR SAF SIF MIJ SRV

if neither MULTWORK nor NOMULT parameters are specified.

The following files are eligible for multiple WORK support:

- o AGL HTTP Agent Log File
- o ERL HTTP Error Messages Log File
- o RFL HTTP Referer Log File
- o SAL HTTP Server Access Log File
- o AMS HTTP Access Method Summary File
- o APS HTTP Access Path Summary File
- o WPR HTTP Performance File
- o SAF WAS Server Activity File
- o SIF WAS Server Interval File
- o MIJ J2EE Method Interval File
- o SRV Servlet Interval File
- o WRA Web Request Activity File
- o WCB Web CPU Usage Breakdown File
- o WUD Web User Data File

The following section discusses changing the WORK option:

1 - Change the Number of Work Files

7.3.2.5.1 Change the Number of Work Files

```
To change the number of work files used in the CA MICS Web
Analyzer processing in Step DAY052, use the checklist that
follows for each unit.

  ************************************************
  *                                              *
  *          CHANGE NUMBER OF WORK FILES         *
  *                                              *
  ************************************************

___ 1.  Update the WORK statement in
        prefix.MICS.PARMS(cccOPS), where (ccc) is the
        component identifier, to specify the number of work
        data sets required.  Below is an example:

        WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))

        where:

        n        - is the number of WORK data sets.
        STORCLAS - specifies a storage class for a new data
                   set.  The name can have up to eight
                   characters.
        SPACE    - specifies how much disk space to provide
                   for a new data set being allocated.
        XXX      - is TRK or CYL.
        pppp     - is the primary allocation.
        ssss     - is the secondary allocation.

        You should specify the minimum number of WORK data
        sets to meet your work space requirements.  As a
        start, try incrementing the number gradually
        beginning from the default.

___ 2.  If this is the first time you are implementing
        multiple work files for this product, then continue
        with Step 2.  If you are just changing the number
        currently in use, or simply the space definitions,
        then proceed to Step 3 of this checklist.

        Browse sharedprefix.MICS.PROTOLIB(DYcccnnn) and
        sharedprefix.MICS.PROTOLIB(cccINCR), where (nnn) is
        the job step number and (ccc) is the product ID for
        this product, checking for the presence of the WORK
        symbolic on the EXEC statement to determine if you
        have previously modified this product to increase the
```

```
        allocation of SAS WORK space.

___ 2a. If you find a WORK symbolic, simply divide the
        primary and secondary allocation values from the WORK
        symbolic by the number of work files specified above
        (value of n on the WORK statement coded in Step 1).

        Coding the resulting values will yield the same
        aggregate space allocation you have been using with a
        single WORK file.  To double your available WORK
        space, carry out the division, double the results and
        use the values in the WORK definition above.

___ 2b. If you did not find a WORK symbolic in PROTOLIB,
        examine prefix.MICS.PARMS(JCLDEF) for each CA MICS
        unit containing this product.  Find the WORKSPACE
        keyword.  The space allocation specified is used for
        a single SAS WORK file.  Perform the same division as
        described in the previous paragraph to determine the
        quantity that will yield equivalent total allocation
        with multiple WORK files.  Then adjust the values
        upward to meet your needs.

___ 3.  Submit the job in prefix.MICS.CNTL(cccPGEN).

___ 4.  If you specified RESTART YES in the product's cccOPS,
        you are done.  Otherwise, you must do Steps 5, 6, and
        7.

___ 5.  Edit prefix.MICS.PARMS(JCLGENU) so that it contains a
        single line that reads:

        DAILY

        or, if incremental update is enabled for this product
        in this unit database, specify:

        DAILY INCRccc

        where ccc is the product ID.

___ 6.  Submit the job in prefix.MICS.CNTL(JCLGENU).  Ensure
        that there are no error messages in MICSLOG or
        SYSTSPRT, that the MICSLOG contains the normal
        termination message, BAS10999I, and that the job
        completes with a condition code of zero.

___ 7.  The following operational job(s) have changed:
```

```
DAILY

INCRccc (if incremental update is enabled)

If your site has implemented the operational CA MICS
processes in a scheduling product, the JCL may have
to be refreshed in that product.  See the scheduling
product's administrator for the exact processes
involved in updating that product's representation of
the CA MICS jobs.
```

7.3.2.6 Internal Step Restart Statements

```
RESTART YES/NO
--------------

This statement is optional.  Specify this to activate
internal step restart for this product's DAILY and/or INCRccc
database update job steps:

RESTART YES


If you do not specify or enable the RESTART parameter, then
this option defaults to the following and internal step
restart is disabled:

RESTART NO


************************************************************
*                                                          *
*  Note:  Changing the RESTART parameter (either from NO   *
*         to YES or from YES to NO) requires regeneration  *
*         of the DAILY operational job by executing        *
*         prefix.MICS.CNTL(JCLGEND) or by specifying       *
*         DAILY in prefix.MICS.PARMS(JCLGENU) and          *
*         executing prefix.MICS.CNTL(JCLGENU).             *
*                                                          *
*         If incremental update is active for this product, *
*         you must also regenerate the INCRccc job.        *
*                                                          *
************************************************************

Internal step restart can significantly reduce time and
resource usage to recover from daily and/or incremental
update processing failures.  CA MICS uses a
checkpoint/restart technique.

o  When internal step restart is activated, the database
   update job step "checkpoints" (or saves) intermediate
   results (work file contents) and the operational
   environment at the end of each processing phase.

o  Then, if required, the database update step can resume
   execution at the beginning of the processing phase in
   which the failure occurred.
```

o Restart is accomplished by restoring the operational environment from the last checkpoint, bypassing completed processing phases, and resuming execution using intermediate results (work files) from the last checkpoint.

Note: When you activate internal step restart (RESTART YES), the following optional restart parameters are enabled. These parameters have no effect if restart is disabled (RESTART NO). For more details, see the individual parameter descriptions later in this section.

- o RESTARTCKPT data_set_allocation_parameters
- o RESTARTWORK data_set_allocation_parameters
- o DYNAMWAIT minutes

Processing Phases:

This product employs two database update processing phases, followed by the two common roll-up phases:

Phase	Description
FORMAT	Read raw input data, convert to SAS format, and output to intermediate work files.
DBUPDATE	Sort intermediate work file contents, eliminate duplicate input data, merge data across optional multiple work files, enhance data content, and create the new DETAIL cycle.
DYSUM	Summarize DETAIL data to create new DAYS cycles and to update current week-to-date and month-to-date cycles.
DYAGE	Cutover new database cycles to production and "age" existing cycles.

RESTART Considerations

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

o Overhead

  Enabling internal step restart adds some overhead to the database update job step -- the cost of taking checkpoints and managing saved materials. Since this overhead is relatively constant and independent of input data volume, you may find that costs outweigh potential savings when input data volume is low, for example in a test unit. For high volume, production units, internal step restart support overhead should be a minor portion of total resource usage.

o Cataloged Work Files

  When internal step restart is enabled, the SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged with permanent dataset names so they will be retained for use in restart if the step abends. These data sets are deleted when the step completes successfully.

  Prior to enabling internal step restart support, these data sets were probably allocated on system "scratch" space with a temporary, system assigned data set names. If your installation standards do not allow "permanent" data sets on DASD volumes used for temporary work space, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct the internal step restart data sets to a generic unit or storage class that allows cataloged data sets.

o Dynamic Allocation

  When internal step restart is active, dynamic allocation is employed for the work data sets. If your installation restricts dynamic allocation of large, cataloged data sets, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct work data set allocation to a generic unit or storage class where dynamic allocation is allowed.

o Data Set Names

  The SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged according to the standard CA MICS unit database data set name conventions. The default DDNAME and data

```
set names are:

  o SAS work data set,
        //cccXWORK DD DSN=prefix.MICS.cccXWORK,.....

  o Internal step restart control data set,
        //cccXCKPT DD DSN=prefix.MICS.cccXCKPT,.....

  o Multiple work file data sets,
        //WORKnn   DD DSN=prefix.MICS.cccWRKnn,.....

Since these data sets conform to the same data set name
conventions as your existing CA MICS data sets, there
should be few, if any, data set name related allocation
issues.  However, it is possible to override the data set
names if required.  Please contact CA MICS Product
Support for assistance if you must alter data set names.



RESTARTCKPT
-----------

This statement is optional.  Specify the following to
override default data set allocation parameters for the
internal step restart checkpoint data set:

RESTARTCKPT   data_set_allocation_parameters

Note:  RESTARTCKPT is ignored when you specify RESTART NO.

The internal step restart checkpoint data set (or cccXCKPT
data set) contains processing status, control, and SAS
environmental information for internal step restart
processing checkpoints.  This includes a copy of the SAS WORK
format and macro catalogs, current macro variable values, and
a description of work files that may be needed to restart
DAYnnn processing.

By default, the cccXCKPT data set is allocated according to
the values you specified for the WORKUNIT and WORKSPACE
parameters in prefix.MICS.PARMS(JCLDEF).  Specify RESTARTCKPT
to override this default, either to alter the space
allocation or to use System Managed Storage (SMS) parameters
to control data set placement and characteristics.

Note:  If you allocate insufficient space for the cccXCKPT
data set, DAYnnn processing will fail and can only be
```

```
restarted from the beginning.

Note:  You can override the RESTARTCKPT data set allocation
parameters at execution-time using the //PARMOVRD facility.
For more information about execution-time override of dynamic
data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks,
according to SAS LIBNAME statement syntax.  If you need
multiple lines, repeat the RESTARTCKPT keyword on the
continuation line.

RESTARTCKPT accepts the engine/host options documented in the
SAS Companion for the z/OS Environment, including STORCLAS,
UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important!  DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

RESTARTCKPT  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp is the primary allocation
             ss is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.

Example 2 (multiple lines):

RESTARTCKPT  STORCLAS=MICSTEMP UNIT=SYSDA
RESTARTCKPT  SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.
```

```
  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
```

```
RESTARTWORK
-----------

This statement is optional.  Specify the following to
override default data set allocation parameters for the
internal step restart WORK data set:

RESTARTWORK   data_set_allocation_parameters

Note:  RESTARTWORK is ignored when you specify RESTART NO.

The internal step restart WORK data set (or cccXWORK data
set) contains the intermediate work files that are not
enabled to multiple work file support, including those files
you may have specified on the optional NOMULT statement.

By default, the cccXWORK data set is allocated according to
the values you specified for the WORKUNIT and WORKSPACE
parameters in prefix.MICS.PARMS(JCLDEF).  Specify RESTARTWORK
to override this default, either to alter the space
allocation or to use System Managed Storage (SMS) parameters
to control data set placement and characteristics.

Note:  If you allocate insufficient space for the cccXWORK
data set, DAYnnn processing will fail and can only be
restarted from the beginning.

Note:  You can override the RESTARTWORK data set allocation
parameters at execution-time using the //PARMOVRD facility.
For more information about execution-time override of dynamic
data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks,
according to SAS LIBNAME statement syntax.  If you need
multiple lines, repeat the RESTARTWORK keyword on the
continuation line.

RESTARTWORK accepts the engine/host options documented in
"SAS Companion for the z/OS Environment", including STORCLAS,
UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.
```

```
Important!  DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

  RESTARTWORK  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp is the primary allocation
             ss is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.

Example 2 (multiple lines):

  RESTARTWORK  STORCLAS=MICSTEMP UNIT=SYSDA
  RESTARTWORK  SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to 8 characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.


 DYNAMWAIT
---------

This statement is optional.  Specify the following:

     DYNAMWAIT   minutes

to override the default amount of time, in minutes, the DAILY
```

```
and/or INCRccc job will wait for an unavailable data set.

Note:  This optional parameter is not normally specified.
       The system default is adequate for most data centers.

Internal Step Restart and Incremental Update facilities use
z/OS dynamic allocation services to create new data sets and
to access existing data sets.  Data set naming conventions
and internal program structure are designed to minimize data
set contention.  However, if data set allocation does fail
because another batch job or online user is already using a
data set, DAILY and/or INCRccc processing will wait 15
seconds and then try the allocation again.  By default, the
allocation will be attempted every 15 seconds for up to 15
minutes.  After 15 minutes, the DAILY or INCRccc job will
abort.

If data set contention in your data center does cause
frequent DAILY or INCRccc job failures, and you are unable to
resolve the contention through scheduling changes, you may
want to use the DYNAMWAIT parameter to increase the maximum
number of minutes the DAILY and/or INCRccc jobs will wait for
the data set to become available.

On the other hand, if your data center standards require
that the DAILY and/or INCRccc jobs fail immediately if
required data sets are unavailable, specify the following:

     DYNAMWAIT 0

Note:  You can override the DYNAMWAIT parameter at
       execution-time using the //PARMOVRD facility.  For
       more information about execution-time override of
       dynamic data set allocation parameters, see the PIOM,
       section 2.3.6.


 The following section discusses enabling this option:

     1 - Enable Internal Step Restart
```

7.3.2.6.1 Enable Internal Step Restart

```
To enable the internal step restart in the CA MICS Web
Analyzer, us the following checklist:

  ************************************************
  *                                              *
  *           ENABLE INTERNAL STEP RESTART       *
  *                                              *
  ************************************************

___ 1.  Edit prefix.MICS.PARMS(cccOPS), where (ccc) is the
        component identifier, and specify:

        RESTART YES

        For additional information on related topic, review
        the documentation for this product on WORK,
        RESTARTWORK, and RESTARTCKPT parameters to override
        default data set allocation parameters.

___ 2.  Submit the job in prefix.MICS.CNTL(cccPGEN).

___ 3.  Edit prefix.MICS.PARMS(JCLGENU) so that it contains a
        single line that reads:

        DAILY

        or, if incremental update is enabled for this product
        in this unit database, specify:

        DAILY INCRccc

        where ccc is the product ID.

___ 4.  Submit the job in prefix.MICS.CNTL(JCLGENU).  Ensure
        that there are no error messages in MICSLOG or
        SYSTSPRT, that the MICSLOG contains the normal
        termination message, BAS10999I, and that the job
        completes with a condition code of zero.

___ 5.  The following operational job(s) have changed:

        DAILY

        INCRccc (if incremental update is enabled)

        If your site has implemented the operational CA MICS
```

```
processes in a scheduling product, the JCL may have
to be refreshed in that product.  See the scheduling
product's administrator for the exact processes
involved in updating that product's representation of
the CA MICS jobs.
```

7.3.2.7 Incremental Update Statements

```
 INCRUPDATE
----------

This statement is optional.  Specify this to enable
incremental update for this product:

INCRUPDATE YES

If you do not specify or enable the INCRUPDATE parameter,
then this option defaults to this and incremental update is
disabled:

INCRUPDATE NO

************************************************************
*                                                          *
*  Note:  Changing the INCRUPDATE parameter (either from NO *
*         to YES or from YES to NO) requires regeneration   *
*         of the DAILY operational job by executing         *
*         prefix.MICS.CNTL(JCLGEND) or by specifying        *
*         DAILY in prefix.MICS.PARMS(JCLGENU) and           *
*         executing prefix.MICS.CNTL(JCLGENU).              *
*                                                          *
*         If you specify INCRUPDATE YES, you must also      *
*         generate the INCRccc, cccIUALC, and cccIUGDG jobs *
*         (where ccc is the 3 character product ID).        *
*         Depending on the options you select, you may also *
*         need to execute the cccIUALC and/or cccIUGDG      *
*         jobs.                                             *
*                                                          *
************************************************************

Incremental update can significantly reduce time and resource
usage in the DAILY job by letting you split out a major
portion of daily database update processing into multiple,
smaller, incremental updates executed throughout the day.

o  Standard CA MICS database update processing involves (1)
   reading and processing raw input data to generate DETAIL
   and DAYS level CA MICS database files, followed by (2)
   summarization of DETAIL/DAYS level data to update
   week-to-date and month-to-date database files.

o  When you activate incremental update:

   -  You can execute the first-stage processing (raw data
```

```
      input to create DETAIL/DAYS files) multiple times
      throughout the day, each time processing a subset of
      the total day's input data.

   -  Then, during the final update of the day (in the
      DAILY job), the incremental DETAIL/DAYS files are
      "rolled-up" to the database DETAIL and DAYS
      timespans, and then summarized to update the
      week-to-date and month-to-date files.

o  Incremental update is independent of your internal step
   restart or DBSPLIT specifications.  You have the option
   to perform incremental updates with or without internal
   step restart support.

o  Incremental update is activated and operates
   independently by product.  The incremental update job
   for this product, INCRccc (where ccc is the product ID),
   can execute concurrently with the incremental update job
   for another product in the same unit database.

o  The CA MICS database remains available for reporting and
   analysis during INCRccc job execution.

************************************************************
*                                                          *
*  Note:  CA MICS is a highly configurable system          *
*         supporting up to 36 unit databases, each of which *
*         can be configured and updated independently.     *
*         Incremental update is just one of the options you *
*         can use to configure your CA MICS complex.       *
*                                                          *
*         All efforts should be made to employ CA MICS     *
*         configuration capabilities to minimize issues    *
*         prior to activating incremental update.  For     *
*         example:                                         *
*                                                          *
*         o  Splitting work to multiple units is an        *
*            effective way to enable parallel database     *
*            update processing                             *
*                                                          *
*         o  Adjusting account code definitions to ensure  *
*            adequate data granularity while minimizing    *
*            total database space and processing time      *
*                                                          *
*         o  Tailoring the database to drop measurements   *
*            and metrics of lesser value to your           *
*            data center, thereby reducing database update *
*            processing and resource consumption           *
```

```
*                                                             *
*        While incremental update is intended to reduce       *
*        DAILY job elapsed time, total resource usage of      *
*        the combined INCRccc and DAILY jobs steps can        *
*        increase due to the additional processing            *
*        required to maintain the incremental update          *
*        "to-date" files and for roll-up to the unit          *
*        database.  The increased total resource usage        *
*        will be more noticeable with small data volumes,     *
*        where processing code compile time is a greater      *
*        percentage of total processing cost.                 *
*                                                             *
***************************************************************
```

Note: When you activate incremental update (INCRUPDATE YES), the following optional incremental update parameters are enabled. These parameters have no effect if incremental update is disabled (INCRUPDATE NO). For more details, see the individual parameter descriptions later in this section.

o INCRDB   PERM/TAPE/DYNAM

o INCRDETAIL data_set_allocation_parameters

o INCRDAYS   data_set_allocation_parameters

o INCRCKPT   data_set_allocation_parameters

o INCRSPLIT USE/IGNORE data_set_allocation_parameters

Incremental update processing reads and processes raw measurement data to create and maintain DETAIL and DAYS level "to-date" files for the current day.

- These incremental update database files are maintained on unique z/OS data sets, independent of the standard CA MICS database files, and independent of any other product's incremental update database files. There is one data set each for DETAIL and DAYS level "to-date" data and a single incremental update checkpoint data set for this product in this unit.

- The incremental update DETAIL and DAYS files can be permanent DASD data sets, or they can be allocated dynamically as needed and deleted after DAILY job processing completes. Optionally, you can keep the incremental update DETAIL and DAYS files on tape, with the data being loaded onto temporary DASD space as needed for incremental update or DAILY job processing.

See the INCRDB PERM/TAPE/DYNAM option for more information.

After activating incremental update, you will use three incremental update facility jobs found in prefix.MICS.CNTL (Note that ccc is the product ID):

o cccIUALC

You execute this job to allocate and initialize the incremental update checkpoint file, and optionally the incremental update DETAIL and DAYS database files. cccIUALC is generally executed just ONE time.

o cccIUGDG

You execute this job to add generation data group (GDG) index definitions to your system catalog in support of the INCRDB TAPE option. cccIUGDG is generally executed just ONE time.

o INCRccc

This is the job you execute for each incremental update. You will integrate this job into your database update procedures for execution one or more times per day to process portions of the total day's measurement data.

Note: The DAILY job is run once at the end of the day. It will perform the final incremental update for the day's data, and then roll-up the incremental DETAIL/DAYS files to the database DETAIL and DAYS timespans and update the week-to-date and month-to-date files.

INCRUPDATE Considerations
---

o Overhead

Incremental update is intended to reduce DAILY job resource consumption and elapsed time by offloading a major portion of database update processing to one or more executions of the INCRccc job. In meeting this objective, incremental update adds processing in the INCRccc and DAILY jobs to accumulate data from each incremental update execution into the composite "to-date" DETAIL and DAYS incremental update files, and also adds processing in the DAILY job to copy the incremental

update files to the unit database DETAIL and DAYS timespans. The amount of this overhead and the savings in the DAILY job are site-dependent, and will vary based on input data volume and on the number of times INCRccc is executed each day.

In addition, activating incremental update will cause additional compile-based CPU time to be consumed in the DAYnnn DAILY job step. The increase in compile time is due to additional code included for each file structure in support of the feature. This increase should be static based on the scope of the CA MICS data integration product in terms of files. This compile-time increase does not imply an increase in elapsed or execution time. Incremental update allows I/O bound, intensive processing (raw data inputting, initial CA MICS transformation, etc.) to be distributed outside of the DAILY job. I/O processing is the largest contributor to elapsed time in large volume applications. Thus, the expected overall impact is a decrease in the actual runtime of the DAYnnn job step.

- Increased "Prime Time" Workload

  By offloading work from the DAILY job to one or more INCRccc executions throughout the day, you are potentially moving system workload and DASD work space usage from the "off-hours," (when the DAILY job is normally executed) to periods of the day where your system resources are in highest demand. You should schedule INCRccc executions carefully to avoid adverse impact to batch or online workloads. For example, if your site's "prime shift" is 8:00 AM to 5:00 PM, you might choose to schedule incremental updates for 7:00 AM (just before "prime shift") and 6:00 PM (just after "prime shift"), with the DAILY job executing just after midnight.

- Increased DASD Usage

  The DASD space required for the incremental update DETAIL and DAYS database files is in addition to the DASD space already reserved for the CA MICS database. By default, the incremental update database files are permanently allocated, making this DASD space unavailable for other applications. In general, you can assume that the incremental update database files will require space equivalent to two cycles of this product's DETAIL and DAYS timespan files.

Alternatively, the incremental update database files can be allocated in the first incremental update of the day and deleted by the DAILY job (see the INCRDB DYNAM option later in this section). This approach reduces the amount of time that the DASD space is dedicated to incremental update, and lets the amount of DASD space consumed increase through the day as you execute each incremental update.

A third option is to store the incremental update database files on tape (see the INCRDB TAPE option). With this approach, the DASD space is required just for the time that each incremental update or DAILY job step is executing. Note that while this alternative reduces the "permanent" DASD space requirement, the total amount of DASD space required while the incremental update or DAILY jobs are executing is unchanged. In addition, the TAPE option adds processing to copy the incremental update files to tape, and to reload the files from tape to disk.

Note: The incremental update checkpoint file is always a permanently allocated disk data set. This is a small data set and should not be an issue.

- Operational Complexity

  Incremental update expands your measurement data management and job scheduling issues. You must ensure that each incremental update and the DAILY job processes your measurement data chronologically; that is, each job must see data that is newer than the data processed by the prior job. By incrementally updating the database, you have more opportunities to miss a log file, or to process a log out of order.

- Interval End Effects

  Each incremental update processes a subset of the day's measurement data, taking advantage of early availability of some of the day's data, for example, when a measurement log fills and switches to a new volume. This can cause a problem if the measurement log split occurs while the data source is logging records for the end of a measurement interval, thus splitting the data for a single measurement interval across two log files. When an incremental update processes the first log file, the checkpoint high end timestamp is set to indicate that this split measurement interval has been processed.

Then, when the rest of the measurement interval's data is encountered in a later update, it can be dropped as duplicate data (because data for this measurement interval end timestamp has already been processed).

Appropriate scheduling of log dumps and incremental updates can avoid this problem. For example, if you plan to run incremental updates at 7:00 AM and 6:00 PM, you could force a log dump in the middle of the measurement interval just prior to the scheduled incremental update executions. This is an extension of the procedure you may already be using for end-of-day measurement log processing. The objective is to ensure that all records for each monitor interval are processed in the same incremental update.

- Dynamic Allocation

  When you activate incremental update and specify TAPE or DYNAM for the INCRDB parameter, dynamic allocation is employed for the incremental update database files. If your site restricts dynamic allocation of large, cataloged data sets, you must use the INCRDETAIL and INCRDAYS parameters to direct incremental update data set allocation to a generic unit or storage class where dynamic allocation is allowed.

- Data Set Names

  The incremental update database files are allocated and cataloged according to standard CA MICS unit database data set name conventions. The DDNAME and default data set names are (where ccc is the product ID):

  - Incremental update checkpoint file,
    `//IUCKPT DD DSN=prefix.MICS.ccc.IUCKPT,.....`

  - Incremental update DETAIL
    `//IUDETAIL DD DSN=prefix.MICS.ccc.IUDETAIL,.....`

  - Incremental update DAYS
    `//IUDAYS DD DSN=prefix.MICS.ccc.IUDAYS,....`

  Since these data sets conform to the same data set name conventions as your existing CA MICS data sets, there should be few, if any, data-set-name-related allocation issues. However, it is possible to override the data set names if required. Contact Technical Support at http://ca.com/support for assistance if you must change

```
  data set names.
 INCRDB
------

This statement is optional.  The default is this:

INCRDB PERM

Note:  INCRDB is ignored when you specify INCRUPDATE NO.

Specify this statement or take the default, to keep the incremental
update database DETAIL and DAYS files on permanently
allocated DASD data sets:

INCRDB PERM

Execute the prefix.MICS.CNTL(cccIUALC) job to allocate the
incremental update database files.

************************************************************
*                                                          *
*  Note:  The incremental update checkpoint file is always  *
*         a permanently allocated DASD data set.           *
*                                                          *
************************************************************


Specify this to offload the incremental update DETAIL and
DAYS files to tape between incremental update executions:

INCRDB TAPE #gdgs UNIT=name

With the TAPE option, the incremental update DETAIL and DAYS
DASD data sets are dynamically allocated at the beginning of
the incremental update job or DAILY job step, and then are
deleted after the job step completes.

o  The first incremental update job of the day allocates
   and initializes the incremental update database files.
   At the end of the job, the DETAIL and DAYS files are
   copied to a new (+1) generation of the incremental
   update tape data sets.  Then the DASD files are deleted.

o  Subsequent incremental update jobs restore the DASD
   incremental update database files from the current, (0)
   generation, incremental update tape data sets before
   processing the input measurement data.  At the end of
   the job, the DETAIL and DAYS files are copied to a new
   (+1) generation of the incremental update tape data
```

sets. Then the DASD files are deleted.

- The DAILY job step also restores the DASD incremental update database files from the (0) generation tape files before processing the input data, but does NOT copy the incremental update database files to tape. Thus, the DAILY job actually creates a new, null (+1) generation.

- Use the #gdgs parameter to specify the maximum number of incremental update tape generations. The minimum is 2 and the maximum is 99, with a default of 5.
  Set the number of generations equal to or greater than the number of incremental updates, including the DAILY job you plan to execute each day. This facilitates restart and recovery if you encounter problems requiring you to reprocess portions of the daily measurement data.

- Use the optional UNIT=name parameter to specify a tape unit name for the incremental update database output tapes. The default is to use the same tape unit as the input tapes.

- A special index must be created in your system catalog for each of the incremental update tape data set generation data groups. The prefix.MICS.CNTL(cccIUGDG) job will generate the statements to create the incremental update GDG index definitions.

  - Before each index is built, it is deleted. These DLTX (or DELETE) statements causes an error message if no entry exists. This is done so that you can change the number of entries without having to delete each of the index entries.

  - DLTX and BLDG (or DELETE and DEFINE) fail if there is a cataloged data set with the same index. IDCAMS (or IEHPROGM) issues a message and gives a return code of 8. This issue is not a problem for non-GDG entries or if the GDG already has the desired number of entries.

  - If you want to change the number of entries kept in a GDG with cataloged data sets, do the following:

    1. Uncatalog any existing entries in the GDG.
    2. Delete the index with a DLTX (or DELETE).
    3. Create the index with a BLDG (or DEFINE).
    4. Catalog any entries that are uncataloged in step 1.

```
o  The incremental update tape data set names are as follows,
   where ccc is the product ID:

   -  Incremental update tape DETAIL file
         tapeprefix.MICS.ccc.IUXTAPE.GnnnnV00

   -  Incremental update tape DAYS file
         tapeprefix.MICS.ccc.IUDTAPE.GnnnnV00

************************************************************
*                                                          *
*  Note:  The INCRDETAIL and INCRDAYS parameters are       *
*         required when you specify INCRDB TAPE.           *
*                                                          *
************************************************************


Specify this parameter to allocate dynamically the
incremental update  DETAIL and DAYS DASD data sets in the first
incremental update of the day, and then delete these data sets
at the end of the DAILY job step:

INCRDB DYNAM


o  With this option, no space is used for the incremental
   update database files during the time between the end of
   the DAILY job step and the beginning of the next day's
   first incremental update.

o  With this approach, you can set the data set allocation
   parameters so that the incremental update DETAIL and DAYS
   data sets start out with a minimum allocation and then
   grow through secondary allocations as more space is
   required for subsequent incremental updates. For
   example, enough space for one incremental update.

************************************************************
*                                                          *
*  Note:  The INCRDETAIL and INCRDAYS parameters are       *
*         required when you specify INCRDB DYNAM.          *
*                                                          *
************************************************************


 INCRDETAIL
----------
```

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional. There is no default.

Specify this to define data set allocation parameters for the incremental update DETAIL data set (IUDETAIL):

INCRDETAIL   data_set_allocation_parameters

Note: INCRDETAIL is ignored when you specify INCRUPDATE NO.

The incremental update DETAIL data set (IUDETAIL) contains the current incremental update detail-level database files, and the DETAIL "to-date" data for the current daily update cycle. You should allocate DASD space equivalent to two cycles of this product's DETAIL timespan data.

If you specified INCRDB PERM (the default), your INCRDETAIL parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.

- Omit the INCRDETAIL parameter if you prefer to specify data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDETAIL parameter specifications are used in incremental update DETAIL data set dynamic allocation during incremental update or DAILY job step execution.

- The INCRDETAIL parameter is required for the TAPE or DYNAM option.

- Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRDETAIL keyword on the continuation line.

- INCRDETAIL accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including

```
   STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and
   VOLSER.

   Important!  DO NOT SPECIFY THE DISP PARAMETER.

o  You can override the INCRDETAIL data set allocation
   parameters at execution-time using the //PARMOVRD
   facility.  For more information about execution-time
   override of dynamic data set allocation parameters, see
   the PIOM, section 2.3.6.

Example 1:

INCRDETAIL   STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp   is the primary allocation
             ss   is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.

Example 2 (multiple lines):

INCRDETAIL   STORCLAS=MICSTEMP UNIT=SYSDA
INCRDETAIL   SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.
```

```
INCRDAYS
--------

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional.  There is no default.


Specify this to define data set allocation parameters for the
incremental update DAYS data set (IUDAYS):

INCRDAYS      data_set_allocation_parameters

Note:  INCRDAYS is ignored when you specify INCRUPDATE NO.

The incremental update DAYS data set (IUDAYS) contains the
current incremental update days-level database files, and the
DAYS "to-date" data for the current daily update cycle.  You
should allocate DASD space equivalent to two cycles of this
product's DAYS timespan data.

If you specified INCRDB PERM (the default), your INCRDAYS
parameter specifications are used in generating the cccIUALC
job (where ccc is the product ID).

o  You will execute the cccIUALC job to allocate and
   initialize the incremental update database and checkpoint
   files.

o  Omit the INCRDAYS parameter if you prefer to specify
   data set allocation parameters directly in the generated
   prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDAYS
parameter specifications are used in incremental update DAYS
data set dynamic allocation during incremental update or
DAILY job step execution.

o  The INCRDAYS parameter is required for the TAPE or DYNAM
   option.

o  Specify data set allocation parameters, separated by
   blanks, according to SAS LIBNAME statement syntax.  If
   you need multiple lines, repeat the INCRDAYS keyword on
   the continuation line.
```

```
o  INCRDAYS accepts the engine/host options documented in the
   SAS Companion for the z/OS Environment, including
   STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and
   VOLSER.

   Important!  DO NOT SPECIFY THE DISP PARAMETER.

o  You can override the INCRDAYS data set allocation
   parameters at execution-time using the //PARMOVRD
   facility.  For more information about execution-time
   override of dynamic data set allocation parameters, see
   the PIOM, Section 2.3.6.

Example 1:

INCRDAYS     STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp   is the primary allocation
             ss   is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.

Example 2 (multiple lines):

INCRDAYS     STORCLAS=MICSTEMP UNIT=SYSDA
INCRDAYS     SPACE=(xxxx,(pp,ss),,,ROUND)

where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
```

a new data set being allocated.

INCRCKPT

This statement is optional. Specify this to override default data set allocation parameters for the incremental update checkpoint data set:

INCRCKPT data_set_allocation_parameters

Note: INCRCKPT is ignored when you specify INCRUPDATE NO.

The incremental update checkpoint data set tracks incremental update job status and the data that has been processed during the current daily update cycle. The incremental update checkpoint is used to detect and block the input of duplicate data during incremental update processing. This data set will be exactly the same size as prefix.MICS.CHECKPT.DATA (the unit checkpoint data set), usually 20K to 200K depending on the prefix.MICS.PARMS(SITE) CKPTCNT parameter (100-9999).

Your INCRCKPT parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- You will execute the cccIUALC job to allocate and initialize the incremental update checkpoint file. If you specified INCRDB PERM, then the cccIUALC job will also allocate the incremental update DETAIL and DAYS database files.

- By default the incremental update checkpoint data set is allocated as SPACE=(TRK,(5,2)) using the value you specified for the prefix.MICS.PARMS(JCLDEF) DASDUNIT parameter.

- Omit the INCRCKPT parameter if you prefer to override data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRCKPT keyword on the continuation line.

INCRCKPT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS,

```
UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important!  DO NOT SPECIFY THE DISP PARAMETER.


Example 1:

INCRCKPT     STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)


where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated, where:

             xxxx is TRK, CYL, or blklen
             pp   is the primary allocation
             ss   is the secondary allocation

             and ROUND specifies that the allocated space be
             "rounded" to a cylinder boundary when the unit
             specified was a block length.  ROUND is ignored
             with the TRK or CYL options.


Example 2 (multiple lines):

INCRCKPT     STORCLAS=MICSTEMP UNIT=SYSDA
INCRCKPT     SPACE=(xxxx,(pp,ss),,,ROUND)


where:

  STORCLAS - specifies a storage class for a new data set.
             The name can have up to eight characters.

  UNIT     - specifies the generic unit for a new data set.
             The name can have up to eight characters.

  SPACE    - specifies how much disk space to provide for
             a new data set being allocated.


DYNAMWAIT
---------
```

This statement is optional. Specify the following:

```
DYNAMWAIT   minutes
```

to override the default amount of time, in minutes, the DAILY and/or INCRccc job will wait for an unavailable data set.

Note: This optional parameter is not normally specified. The system default is adequate for most data centers.

Internal Step Restart and Incremental Update facilities use z/OS dynamic allocation services to create new data sets and to access existing data sets. Data set naming conventions and internal program structure are designed to minimize data set contention. However, if data set allocation does fail because another batch job or online user is already using a data set, DAILY and/or INCRccc processing will wait 15 seconds and then try the allocation again. By default, the allocation will be attempted every 15 seconds for up to 15 minutes. After 15 minutes, the DAILY or INCRccc job will abort.

If data set contention in your data center does cause frequent DAILY or INCRccc job failures, and you are unable to resolve the contention through scheduling changes, you may want to use the DYNAMWAIT parameter to increase the maximum number of minutes the DAILY and/or INCRccc jobs will wait for the data set to become available.

On the other hand, if your data center standards require that the DAILY and/or INCRccc jobs fail immediately if required data sets are unavailable, specify the following:

```
DYNAMWAIT 0
```

Note: You can override the DYNAMWAIT parameter at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

The following section discusses enabling this option:

1 - Implement Incremental Update

7.3.2.7.1 Implement Incremental Update

```
To implement incremental update in the CA MICS Web Analyzer,
use the following checklist:


  ***************************************************
  *                                                 *
  *           IMPLEMENT INCREMENTAL UPDATE          *
  *                                                 *
  ***************************************************

___ 1.  Edit prefix.MICS.PARMS(cccOPS), where (ccc) is the
        component identifier:

        o  Specify the following:

           INCRUPDATE  YES

        o  If you want to store the incremental update
           database files on tape between incremental
           updates, specify this:

           INCRDB  TAPE  #gdgs

        o  If you want to allocate the incremental update
           database files during the first incremental update
           of the day and delete these data sets at the end
           of the DAILY job step, specify this:

           INCRDB  DYNAM

        o  If you specified INCRDB TAPE or INCRDB DYNAM, then
           you must also specify this:

           INCRDETAIL  data_set_allocation_parameters
           INCRDAYS    data_set_allocation_parameters

        o  If you want the incremental update job for this
           product to get input measurement data from the
           output of the SPLITSMF job, specify this:

           INCRSPLIT  USE  data_set_allocation_parameters

        o  For additional information on related topic,
           review the documentation for this product on
           INCRCKPT, INCRDETAIL, INCRDAYS, or INCRSPLIT
           parameters to override default data set allocation
```

parameters.

___ 2. Submit the job in prefix.MICS.CNTL(cccPGEN).

___ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it contains two or more lines reading:

```
DAILY
INCRccc cccIUALC cccIUGDG
```

___ 4. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.

___ 5. Edit the job in prefix.MICS.CNTL(cccIUALC).

- Inspect and/or specify data set allocation parameters for the incremental update database and checkpoint files. If you specified INCRDB TAPE or INCRDB DYNAM, the cccIUALC job will only allocate the incremental update checkpoint data set.
- Submit the job. Ensure that there are no error messages in MICSLOG or SASLOG, and that the job completes with a condition code of zero.

___ 6. If you specified INCRDB TAPE, submit the job in prefix.MICS.CNTL(cccIUGDG) to define generation group indexes for the incremental update DETAIL and DAYS tape data sets. Examine SASLOG, MICSLOG, and SYSPRINT to verify that the generation group indexes were correctly defined.

Note: You may see error messages for the DLTX (or DELETE) statements. This is not a problem. cccIUGDG deletes each index prior to defining it, and an error message is issued if the index does not yet exist (e.g., if this is the first time you ran the cccIUGDG job).

___ 7. The following operational job(s) have changed:

```
DAILY  INCRccc
```

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling

product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

___ 8. Implement operational procedures for gathering input measurement data and executing incremental updates (INCRccc) during the day.

You may also need to modify operational procedures for the DAILY job to ensure that processing is limited to input measurement data that has not been input to one of the day's incremental update executions.

7.3.2.8 DETAIL Tape Processing Overview

For certain high volume files, the CA MICS architecture provides the ability to create optional DETAIL timespan data sets. These optional data sets are separate from the normal CA MICS database and have the following characteristics:

- They are created during the component step execution: during either an incremental update or the DAILY update run. Each DETAIL tape file has a separate DD statement in the DAILY or incremental update job that specifies the data set name and other data set characteristics.

- The data set name (DSN) for each DETAIL tape in a component step is:

  tapeprefix.tmics.DETAIL.iiifff(+1)

  where:

  - tapeprefix is the tape prefix value specified with the TAPEPREFIX statement in prefix.MICS.PARMS(JCLDEF)

  - tmics is either null when NOMICSLEVEL is specified, or MICS if MICSLEVEL is specified with the TAPEPREFIX statement in prefix.MICS.PARMS(JCLDEF)

  - iii is the three character CA MICS information area associated with the file

  - fff is the three character CA MICS file identifier

  Example: For the CA MICS DB2 Analyzer, if DETAIL tape processing were activated for the DB2DSU file, the following prefix.MICS.PARMS(JCLDEF) statement:

  TAPEPREFIX CPLXA.U1T NOMICSLEVEL

  yields the following DETAIL tape DSN:

  CPLXA.U1T.DETAIL.DB2DSU(+1)

- There is no cycle aging associated with these data sets. Instead, each DETAIL tape data set created is an 01 cycle (for example, DB2DSU01), and multiple instances are

managed using Generation Data Groups (GDGs).

- The creation of DETAIL tape data sets is independent of any sharedprefix.MICS.GENLIB customization choices. You can have one or more cycles of a file written to and managed in the unit DETAIL timespan database, and also use the DETAIL tape feature to create independent data sets for that same file.

- The decision to create DETAIL tape data sets is made at the unit level. You can choose to create DETAIL tape data sets for a component file in unit A, but not in unit B.

- While the expectation is that these files will be written to tape, you can make JCL choices to target other media such as DASD. Note that the SAS sequential tape engine will be used to write these files for both DASD and tape data sets.

- Unique output exits are provided to permit manipulation and subsetting of the records written to the DETAIL tape data sets. This manipulation and subsetting, however, has no impact on files written to the standard CA MICS database.

The activation of DETAIL tape data sets is accomplished by specifying a TAPEfff statement in prefix.MICS.PARMS(cccOPS).

Customization of the JCL used to create each data set is accomplished using a cascading hierarchy of parameters beginning with choices made with the TAPEPARM statement in prefix.PARMS(JCLDEF).

The following sections provide the information required to activate DETAIL tape processing:

1 - Component Files Supporting DETAIL Tape Activation
2 - DETAIL Tape JCL Customization Overview
3 - The TAPEfff Statement
4 - The DETAILTAPEPARM Statement
5 - DETAILTAPESMSPARM and TAPEfffSMS Statements
6 - DETAIL Tape User Exits
7 - Activating DETAIL Tape Checklist
8 - Deactivating DETAIL Tape Checklist

7.3.2.8.1 WEB DETAIL Tape Supported Files

The CA MICS WEB Analyzer supports DETAIL tape processing for the following files:

- WEBWRA - Web Request Activity File
- WEBWUD - Web User Data File
- WEBWAA - Web Asynch Activity File

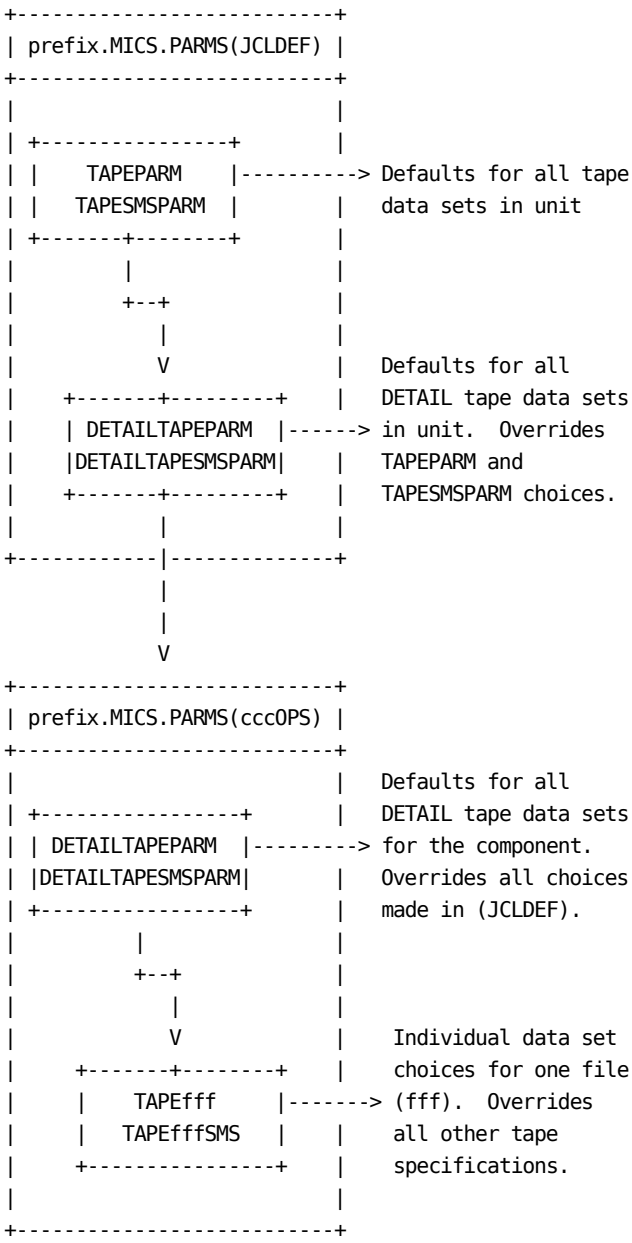
7.3.2.8.2 Overview of JCL Customization Parameters

```
The JCL for DETAIL tape data sets is constructed using tape-
related parameters that are specified in prefix.MICS.PARMS(JCLDEF) and
prefix.MICS.PARMS(cccOPS).

Each DETAIL tape data set has a unique DD in the component
step.  The JCL for each DETAIL tape data set is customized
according to user-specified JCL parameters such as the number
of generation data group (GDG) entries that are maintained, unit
names, volume count, retention, and expiration dates, and
 System Managed Storage (SMS) keywords.For example,
STORCLAS=storclas.

CA MICS provides a hierarchical means to specify tape-related
JCL parameters suitable for each data center.

The hierarchy, as it applies to DETAIL tape specification, is
described:

o Unit Level - TAPEPARM and TAPESMSPARM

  Used for all tape data sets created by all components in
  the unit.  Specified in prefix.MICS.PARMS(JCLDEF)

o Unit DETAIL Tape Level - DETAILTAPEPARM and
                           DETAILTAPESMSPARM

  Used for all DETAIL tape data sets created by all
  components in the unit.  Overrides choices that were
  made in TAPEPARM and TAPESMSPARM.
  Specified in prefix.MICS.PARMS(JCLDEF).

o Component Level - DETAILTAPEPARM and DETAILTAPESMSPARM

  Used for all DETAIL tape data sets created by a particular
  component.  Overrides all choices that were made in JCLDEF.
  Specified in prefix.MICS.PARMS(cccOPS), where ccc
  represents the component.

o File Level - TAPEfff and TAPEfffSMS

  Used for a particular DETAIL tape data set.  The fff
  identifies the specific DETAIL file.  Overrides all choices
  made in JCLDEF and choices that were made with
  DETAILTAPEPARM and DETAILTAPESMSPARM in
  prefix.MICS.PARMS(cccOPS).  Specified in
```

prefix.MICS.PARMS(cccOPS), where ccc represents the component.

The JCL parameter specification hierarchy that is used for DETAIL tape customization is shown graphically in the following diagram:

DETAIL Tape JCL Parameter Hierarchy



```
                    |
                    V

// DD statements and GDG generation
```

7.3.2.8.3 The TAPEfff Statement

```
The TAPEfff statement is used to:

o Activate DETAIL tape data set creation for the file
  indicated by fff.  The component files eligible for DETAIL
  tape processing are listed in a previous section.

o Override, for the indicated file, any JCL-related
  tape specifications made in prefix.MICS.PARMS(JCLDEF)
  and prefix.MICS.PARMS(cccOPS) with TAPEPARM or
  DETAILTAPEPARM statements.

The syntax is as follows:

TAPEfff <optional parameters>

where fff is the unique 3-character file identifier

The optional parameters include both keyword parameters and
keyword=value specifications.  You can include multiple
TAPEfff statements if required in order to specify the
desired optional parameters.  If you specify a parameter
multiple times, the last instance defines the value in
effect.

You can specify as many parameters as you want on each
statement, but each TAPEfff statement must end by column 72.
You may also choose to specify only one parameter per TAPEfff
statement.  Either approach is acceptable.

The optional parameters available for the TAPEfff statement
are listed below:

TAPEfff GDG=nnn UNIT=unitname VOLCOUNT=n
TAPEfff EXPDT=expdt / RETPD=nnn / NOEXPDT
TAPEfff COMP/NOCOMP
TAPEfff STACK/NOSTACK AFF/NOAFF
TAPEfff MODEL='modeldsn' / MODEL=NOMODEL
TAPEfff DISK / TAPE  SPACE=spacedef
TAPEfff VOLSER=volser

The following is a sample TAPEfff statement:

TAPEfff GDG=3 VOLCOUNT=7

In this sample, only the number of generation data groups
```

(GDGs) and VOLCOUNT parameters are explicitly specified for the file fff. All other JCL related parameters for the data set will be set according to specifications made with TAPEPARM and DETAILTAPEPARM statements according to the hierarchy described in a previous section.

The optional parameters for the TAPEfff statement are identical to those supported by the DETAILTAPEPARM statement. A detailed description of the optional parameters is presented immediately following the DETAILTAPEPARM statement description.

7.3.2.8.4 The DETAILTAPEPARM Statement

In prefix.MICS.PARMS(cccOPS), the optional DETAILTAPEPARM statement is used to specify parameters defining processing options for CA MICS DETAIL tape data sets at the component (ccc) level.

When specified in prefix.MICS.PARMS(cccOPS), the DETAILTAPEPARM parameter specifications override unit level tape data set defaults established by TAPEPARM and DETAILTAPEPARM statements in prefix.MICS.PARMS(JCLDEF).

Refer to chapter 2 of the CA MICS Planning, Installation, Operations, and Maintenance Guide for unit level TAPEPARM and DETAILTAPEPARM statements in prefix.MICS.PARMS(JCLDEF).

The primary reason to include a DETAILTAPEPARM statement in prefix.MICS.PARMS(cccOPS) is to specify unique parameters for the DETAIL tapes created for a particular component (ccc).

For example, your site may want to direct the CA MICS DETAIL tapes for a particular component (ccc) to a unique tape volume pool, or to a tape device type such as a virtual tape unit, or to a DASD pool backed by an archive facility. If so, this could be accomplished by including the appropriate DETAILTAPEPARM statements in prefix.MICS.PARMS(cccOPS).

DETAILTAPEPARM <parameters>

Note: In prefix.MICS.PARMS(cccOPS), all DETAILTAPEPARM statements must precede any TAPEfff or TAPEfffSMS statements.

The general form of the DETAILTAPEPARM statement is the keyword DETAILTAPEPARM, followed by one or more DETAILTAPEPARM parameters. DETAILTAPEPARM parameters include both keyword parameters and keyword=value specifications. You can specify the DETAILTAPEPARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the DETAILTAPEPARM parameters:

- Each parameter is described in detail below.

- While the following syntax sample shows only one or two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length).

```
DETAILTAPEPARM GDG=nnn UNIT=unitname VOLCOUNT=n
DETAILTAPEPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
DETAILTAPEPARM COMP/NOCOMP
DETAILTAPEPARM STACK/NOSTACK AFF/NOAFF
DETAILTAPEPARM MODEL='modeldsn' / MODEL=NOMODEL
DETAILTAPEPARM DISK / TAPE  SPACE=spacedef
DETAILTAPEPARM VOLSER=volser
```

The following is a sample DETAILTAPEPARM statement:

```
DETAILTAPEPARM GDG=3 VOLCOUNT=7
```

In this example, the default parameter values are used for most of the DETAILTAPEPARM parameters.

The following describes each of the DETAILTAPEPARM parameters:

- GDG=nnn

  Defaults to 7.

  The number of entries to be created in the GDG for each DETAIL tape data set.

  Overrides the TAPEPARM statement GDG=nnn.

- UNIT=unitname

  Defaults to the value specified on the TAPEPARM statement or to UNIT=3490 if UNIT=unitname is not specified on the TAPEPARM statement.

  The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement UNIT=unitname Value, the old-style TAPEUNIT statement, or both.

- VOLCOUNT=n

  Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, the z/OS default volume count of 5 applies.

The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

- EXPDT=expdt

Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is used which equals the GDG=nnn value.

Specify an expiration date definition which replaces the derived default retention period (RETPD) specifications. This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

Note: For some tape management systems, certain EXPDT values makes the tape available for scratching immediately. Such values are NOT suitable for CA MICS tapes.

The EXPDT=expdt parameter is optional, and, if omitted, CA MICS derives a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT

parameters are mutually exclusive.

- NOEXPDT

  Defaults to the value specified on the TAPEPARM statement.

  Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates, retention periods, or both.

  Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

  Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

- COMP / NOCOMP

  Defaults to the value specified on the TAPEPARM statement. If COMP or NOCOMP is not specified, the z/OS default for the selected tape unit type applies.

  This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

  Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

- STACK / NOSTACK

  Defaults to NOSTACK for DETAIL tapes.

  Specify STACK to place multiple CA MICS tape files on a single tape volume. This mode of operation uses a minimum number of tape volumes

and tape mounts are minimized. CA MICS JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

Specify NOSTACK (the default) to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. This option is useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the TAPEPARM statement STACK/NOSTACK parameter.

- AFF / NOAFF

  Defaults to the value specified on the TAPEPARM statement or to AFF.

  Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

  Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

  Note, if you specify NOAFF, then NOSTACK is automatically set.

  Overrides the TAPEPARM statement AFF/NOAFF parameter.

- MODEL='modeldsn' / MODEL=NOMODEL

  Defaults to the value specified on the TAPEPARM statement.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

- DISK / TAPE

  Defaults to TAPE.

  Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

  When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

  Overrides the TAPEPARM statement DISK/TAPE parameter.

- SPACE=spacedef

  No default.

  Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

  Overrides the TAPEPARM statement SPACE= parameter.

- VOLSER=volser

  No default.

```
Defines the volume serial number of a disk for a
data set defined with the DISK parameter.  Specify
this parameter according to JCL language syntax.

Overrides the TAPEPARM statement VOLSER= parameter.
```

7.3.2.8.5 DETAILTAPESMSPARM and TAPEfffSMS Statements

In prefix.MICS.PARMS(cccOPS), the optional DETAILTAPESMSPARM and TAPEfffSMS statements allow you to specify System Managed Storage (SMS) keywords (for example,STORCLAS=storclass) for DETAIL tape data sets at the component and individual file level, respectively.

DETAILTAPESMSPARM

When specified in prefix.MICS.PARMS(cccOPS), the DETAILTAPESMSPARM parameter specifications override unit level tape data set SMS defaults established by TAPESMSPARM and DETAILTAPESMSPARM statements in prefix.MICS.PARMS(JCLDEF).

Refer to chapter 2 of the CA MICS Planning, Installation, Operations, and Maintenance Guide for unit level TAPESMSPARM and DETAILTAPESMSPARM statements in prefix.MICS.PARMS(JCLDEF).

The primary reason to include a DETAILTAPESMSPARM statement in prefix.MICS.PARMS(cccOPS) is to specify unique SMS parameters for the DETAIL tapes created for a particular component (ccc).

For example, your site may want to direct the CA MICS DETAIL tapes for a particular component (ccc) to a unique storage class.

Sample DETAILTAPESMSPARM statement:

```
DETAILTAPESMSPARM STORCLAS=DTPCLASS
```

In this example, the JCL for all DETAIL tape data sets for the component will include the SMS STORCLAS=DTPCLASS assignment.

TAPEfffSMS

The TAPEfffSMS statement is only permitted in prefix.MICS.PARMS(cccOPS). It is not supported in prefix.MICS.PARMS(JCLDEF). TAPEfffSMS keyword specifications override, for the indicated file, any SMS related tape specifications made in prefix.MICS.PARMS(JCLDEF) and prefix.MICS.PARMS(cccOPS) with TAPESMSPARM or DETAILTAPESMSPARM statements.

The primary reason to include a TAPEfffSMS statement in prefix.MICS.PARMS(cccOPS) is to specify unique SMS parameters for a specific DETAIL tape file (fff).

Sample TAPEfffSMS statement:

```
    TAPEfffSMS STORCLAS=fffCLASS
```

In this example, the JCL for the DETAIL tape data set for file ffft will include the SMS STORCLAS=fffCLASS assignment.

7.3.2.8.6 DETAIL Tape User Exits

A user exit is invoked just prior to the output of each record to a DETAIL tape file. The exits are named _USRTfff, where fff matches the file identifier for the DETAIL tape file.

Refer to chapter 10 of this guide for a complete description of the DETAIL tape user exits.

7.3.2.8.7 Activating DETAIL Tape Checklist

___ 1. Add or modify the desired TAPEfff and TAPEfffSMS statements in prefix.MICS.PARMS(cccOPS), where fff is the three-character file identifier for the file to be written to tape.

___ 2. Submit prefix.MICS.CNTL(cccPGEN).

___ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it reads:

cccGDGGN DAILY

If Incremental Update is active, add this:

INCRccc

___ 4. Submit prefix.MICS.CNTL(JCLGENU) to regenerate the JCL for DAILY and cccGDGGN.

Note: If you want to use specific JCL and SMS parameter values for your DETAIL tape data sets, make sure you have coded the appropriate DETAILTAPEPARM and/or DETAILTAPESMSPARM statements in prefix.MICS.PARMS(JCLDEF) for unit level, or prefix.MICS.PARMS(cccOPS) for component level DETAIL tape JCL parameters. Refer to the previous sections for a detailed discussion on JCL parameters available for DETAIL tape data sets.

___ 5. Submit prefix.MICS.CNTL(cccGDGGN) to define the tape GDG.

Note: If you want to override the number of entries kept in a GDG with cataloged data sets for a TAPEfff file, you must delete and redefine the index with the new limit for GDG entries. Follow the checklist in section 3.5.6.3 of the PIOM.

___ 6. You have just activated DETAIL tape for one or more database files. As mentioned in the DETAIL Tape Processing Overview section, the choice to create DETAIL tape files is independent of, and does not affect the creation of DETAIL files in the unit database.

If you are currently creating one or more cycles of

the files you just activated for DETAIL tape in your unit prefix.MICS.DETAIL timespan, your DAILY update job will continue to do so.

Now that you are creating DETAIL tape cycles for the files, you may want to deactivate, or limit the number of DETAIL cycles written to the unit DETAIL timespan database.

The CA MICS System Modification Guide contains instructions for changing the number of cycles of a file as well as instructions for deactivation of the DETAIL timespan of a file.

If you deactivate the files in the DETAIL timespan, make sure you delete the existing cycles to free up space in the prefix.MICS.DETAIL database. If you reduce the number of cycles, make sure you delete any existing cycles in prefix.MICS.DETAIL beyond the new cycle limit.

The following JCL can be used to delete specific cycles of a file from the DETAIL timespan. Make sure to replace 'x' with the unit identifier:

a) Delete all cycles for file fff (01-nn)

```
//DELETE  EXEC MICSDBx
//SYSIN   DD *
 PROC DATASETS DDNAME=%DDNx(TS=DETAIL);
   DELETE cccfff01 cccfff02 ... cccfffnn ;
   RUN;
```

b) Delete cycles 02, 03, and 04 for file fff:

```
//DELETE  EXEC MICSDBx
//SYSIN   DD *
 PROC DATASETS DDNAME=%DDNx(TS=DETAIL);
   DELETE cccfff02 cccfff03 cccfff04 ;
   RUN;
```

7.3.2.8.8 Deactivating DETAIL Tape Checklist

___ 1. Delete the desired TAPEfff and TAPEfffSMS statements from prefix.MICS.PARMS(cccOPS).

___ 2. If there are no longer any TAPEfff statements in prefix.MICS.PARMS(cccOPS), delete any DETAILTAPEPARM and DETAILTAPESMSPARM statements.

___ 3. Submit prefix.MICS.CNTL(cccPGEN).

___ 4. Edit prefix.MICS.PARMS(JCLGENU) so that it reads:

DAILY

If Incremental Update is active, add this:

INCRccc

___ 5. Submit prefix.MICS.CNTL(JCLGENU) to regenerate the JCL for the DAILY job.

___ 6. (Optional) Delete the index and the cataloged datasets for each TAPEfff statements deleted in step 1.

7.3.2.9 Global and System Level GMT Time Offsets

The raw data time and datetime fields in the SMF type 120 records are recorded in GMT time. The raw data time and datetime fields in the SMF type 103 and the USS generated WEB log files are recorded in local time.

In order to allow you to store all WEB related data in local time, regardless of the data source, two optional statements are provided:

- GLOBAL GMTOFF +/-nn.nn
- GMTOFF orgsysid +/-nn.nn

These statements, if coded, only affect the date and time related data elements in the files created from the SMF type 120 records. The SMF type 120 record subtypes processed by this analyzer are described in section 6.2.1.2.

The following Web Analyzer files created from the SMF type 120 records are affected by use of the GMT time offset statements:

- WEBHPI JVM Heap Interval File
- WEBSAF WAS Server Activity File
- WEBSIF WAS Server Interval File
- WEBMIJ J2EE Method Interval File
- WEBHTP HTTP Session Manager Interval File
- WEBSRV WebContainer Servlet Interval File

Refer to section 5.2 for more information about these files.

Note that the time offset only affects the date and time fields in the CA MICS database records, but does not affect the datetime stamp in the prefix.MICS.CHECKPT.DATA data set. That time stamp represents the time and date that SMF data was written to the SMF buffer and is already in local time.

The GMT time offset statements are described in the following sections:

1. GLOBAL GMTOFF +/-nn.nn Statement
2. GMTOFF orgsysid +/-nn.nn Statement

7.3.2.9.1 GLOBAL GMTOFF +/-nn.nn Statement

This statement is optional and lets you define a global time offset value for every system. This default time offset can be overridden at the original (ORGSYSID) level by coding a GMTOFF orgsysid +/-nn.nn statement.

This statement has the following format:

GLOBAL GMTOFF +/-nn.nn

where:

+/- indicates whether the offset value should be added to (+) or subtracted from (-) the raw data time values. If not coded, a plus sign (+) is assumed.

nn.nn is the adjustment time in hours and fractions of hours that will be added to, or subtracted from, the raw data time values. The range supported is any value from -13.00 through 13.00 hours. The fractional portion is provided to accommodate countries with non-hour boundary timezones.

Note that the fractional part represents fractions of an hour, not minutes. This means that .5 equals 30 minutes and .25 equals 15 minutes.

Examples:

GLOBAL GMTOFF -5 5 hours (18000 seconds) will be subtracted from raw data time values for all systems.

GLOBAL GMTOFF 2.5 2 hours 30 minutes (9000 seconds) will be added to raw data time values for all systems.

7.3.2.9.2 GMTOFF orgsysid +/-nn.nn Statement

This statement is optional and lets you define a time offset value for a specific system. The time offset value defined with this statement affects only the system whose original sysid value is coded on the statement. It overrides, for the system specified, the value specified in the GLOBAL GMTOFF statement.

Note: It is not necessary to have a GLOBAL GMTOFF statement coded in order to use the GMTOFF statement.

This statement has the following format:

GMTOFF orgsysid +/-nn.nn

where:

orgsysid — is the original system identifier (ORGSYSID) found in raw data. It can be up to 4 EBCDIC characters and is input from offset 11 in the SMF type 120 record.

+/- — indicates whether the offset value should be added to (+) or subtracted from (-) the raw data time values. If not coded, a plus sign (+) is assumed.

nn.nn — is the adjustment time in hours and fractions of hours that will be added to, or subtracted from, the raw data time values. The range supported is any value from -13.00 through 13.00 hours. The fractional portion is provided to accommodate countries with non-hour boundary timezones.

Note that the fractional part represents fractions of an hour, not minutes. This means that .5 equals 30 minutes and .25 equals 15 minutes.

Examples:

GMTOFF TSO1 -5 — 5 hours (18000 seconds) will be subtracted from raw data time values for system TSO1.

GMTOFF SYS1 2.5 — 2 hours 30 minutes (9000 seconds) will be added to raw data time values for system SYS1.

7.3.3 Web IP/PORT Associated Parameters (WEBSLIST)

Member WEBSLIST, in the prefix.MICS.PARMS data set, can be used to associate meaningful, user-defined names with each web server. If you do not use the WEBSLIST member, web servers in your database files are represented by their numeric WEB IP Address (WEBIP) and numeric Web Server Port Number (WEBPORT).

WEBSNAME and WEBNAME

You can define both a Web Server Short Name (WEBSNAME) and a Web Server Name (WEBNAME) that are stored as data elements in the database files.

The short name, (WEBSNAME), can be up to 4 characters in length.

The long name, (WEBNAME), can be up to 25 characters in length.

WEBGROUP

Additionally, web servers can be associated with each other using a Web Group Name (WEBGROUP). A Web Group Name can be up to 8 characters in length. Any number of web servers can be assigned to a Web Group Name.

The following table shows how the WEBSLIST values are specified:

IP ADDRESS PORT NUMBER	SHRT NAME	GROUP NAME	WEB SERVER NAME
>> 25 CHARACTERS <<	4 CH	8 CHAR.	>> 25 CHARACTERS <<
111.222.333.444/1080	SALE	B2B	ATLANTIC DIVISIONS
222.333.444.111/2080	HR	B2B	CORPORATE PAYROLL
333.444.111.222/3080	SALE	C2B	PACIFIC DIVISIONS
444.111.222.333/4080	HR	B2B	CORPORATE BENEFITS
444.333.222.111/6080	DEV	C2B	APPLICATION DEVELOP.

```
The following rules apply when coding this member:

o Required:  Ip Address/Port (WEBIP/WEBPORT)   COLS  4 - 28
o Optional:  Short Name      (WEBSNAME)        COLS 30 - 33
o Optional:  Group Name      (WEBGROUP)        COLS 35 - 42
o Optional:  Web Server Name (WEBNAME)         COLS 44 - 68

An entry is valid if the WEBIP/WEBPORT is entered along with
at least one of three optional entries - WEBSNAME, WEBGROUP,
or WEBNAME.

Any values you specify for WEBSNAME, WEBGROUP, or WEBNAME are
used to populate the data elements of those names in the
database files.  Optional values not specified result in
blanks in the corresponding file data elements.

The exception is the long name, Web Server Name (WEBNAME).
If Web Server Name is not specified in the WEBSLIST member,
then the WEBNAME data element is automatically populated with
the concatenation of Web IP Address (WEBIP) and Web Server
Port Number (WEBPORT).
```

7.3.4 INPUTRDR and INPUTWEB PARMS Members

```
The prefix.MICS.PARMS members INPUTRDR and INPUTccc are used
to specify the input data for the Data Integration
Application component step.

To determine whether INPUTccc or INPUTRDR or both are used,
review the DAYSMF, SMFRECORDING, and SMFDRCTR specifications
in prefix.MICS.PARMS(JCLDEF).

                    INPUTRDR            INPUTccc

DAYSMF OFF            NO                  YES
DAYSMF FILE(S) ...    YES                 NO
SMFDIRECTOR           NO                  YES
DAYSMF EXCLUDE ccc    NO - See Note 1     YES

SMFRECORDING ccc      YES                 See Note 2
```

When DAYSMF OFF is specified in JCLDEF, then the prefix.MICS.PARMS(INPUTccc) member is used for each component step.

When DAYSMF FILES PERMANENT or TEMPORARY is specified in JCLDEF, the prefix.MICS.PARMS(INPUTRDR) member is used in DAYSMF.

If there is only one SMF recording component in this unit, DAYSMF will not be generated and the component step with SMF input data will use the prefix.MICS.PARMS(INPUTRDR) member.

NOTES:
If the use of DAYSMF EXCLUDE ccc results in only one component remaining as DAYSMF eligible, DAYSMF will automatically be deactivated. In this case, INPUTRDR will be used as stated above.

Some components accept data from both SMF and non-SMF sources, such as CIC and VCA. In this case, the component may optionally input data from the INPUTRDR member, but will always include the INPUTccc member.

For details on the DAYSMF parameters, see section 2.3.3.2.1, JCL Option Definitions (JCLDEF) in the PIOM.

For details on deactivating DAYSMF, see section 5.10 in the PIOM.

The following sections describe how to specify the input data set JCL definitions.

1 - INPUTRDR PARMS Member JCL Definition
2 - INPUTWEB PARMS Member JCL Definition

7.3.4.1 INPUTRDR PARMS Member JCL Definition

This section describes the information required to define the Analyzer to the Database Space Modeling Facility. Specifically, the user must provide values for the cycle (data retention) definitions. The special considerations for defining the FILE input statements to the Database Space Modeling facility are discussed.

Section 2.3.4 of the PIOM, Database Space Modeling Facility, explains how these values are used in estimating the DASD requirements of the database, and how the user can input these values to the modeling facility.

This section covers the following topics:

1 - Data Retention Specifications (FILE Statements)

Figure 7-3. INPUTRDR JCL Definition Worksheet

7.3.4.2 INPUTWEB PARMS Member JCL Definition

The INPUTWEB member of prefix.MICS.PARMS contains the DD statements to specify the input data for the CA MICS Web Analyzer. The //INPUTSMF DD and, optionally, the //SMFDRCTR DD statements are used by the DAY052 step of the DAILY job.

The INPUTWEB member contains the input SMF data sets when ANY of the following JCLDEF options is used:

- DAYSMF OFF is specified
- SMFDRCTR/SMFDIRECTOR WEB is specified
- DAYSMF EXCLUDE WEB is specified

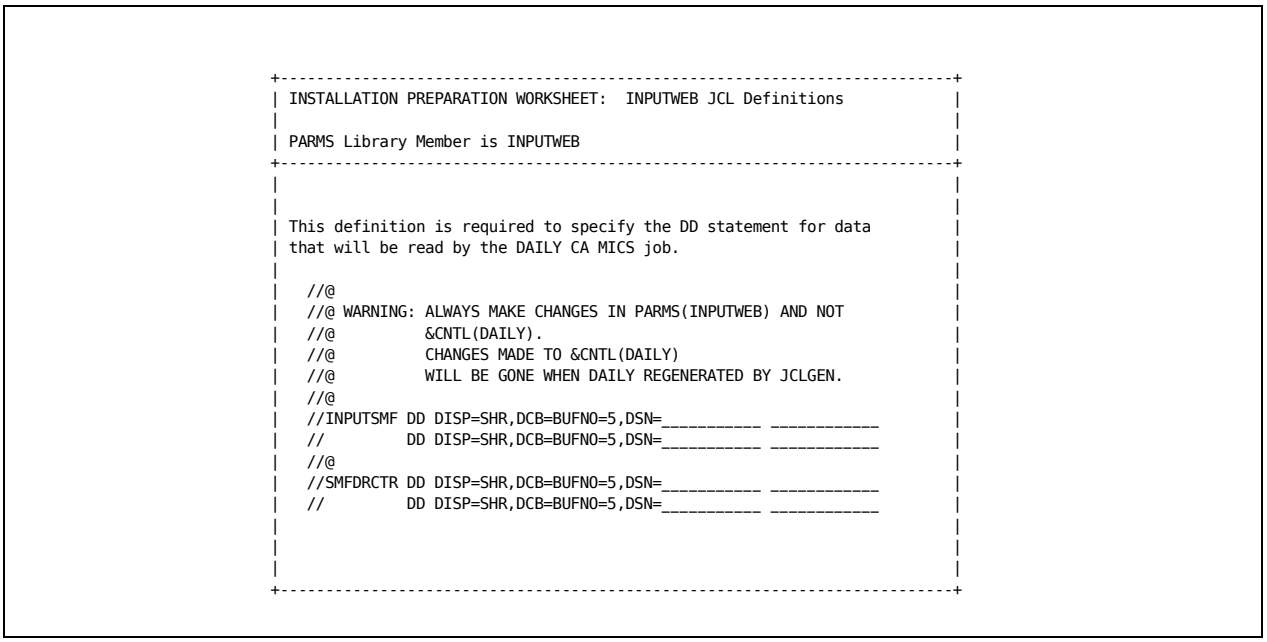
A //INPUTSMF DD statement is required for each SMF data source.

A //SMFDRCTR DD statement is only required when CA SMF Director split indices are used.

A worksheet for preparing the INPUTWEB member is provided below in Figure 7-4.

If you change the contents of the INPUTWEB member, you must regenerate the CA MICS DAILY job using either JCLGENU or JCLGEND in prefix.MICS.CNTL.

NOTE: If your site has CA SMF Director installed, you can use it to extract specific SMF record types and subtypes at SMF dump time. For more details on this product feature see Section 6.3, CA MICS and CA SMF Director Interface, of this guide.



```
+-------------------------------------------------------------------------+
| INSTALLATION PREPARATION WORKSHEET:  INPUTWEB JCL Definitions           |
|                                                                         |
| PARMS Library Member is INPUTWEB                                        |
+-------------------------------------------------------------------------+
|                                                                         |
|                                                                         |
| This definition is required to specify the DD statement for data        |
| that will be read by the DAILY CA MICS job.                             |
|                                                                         |
|   //@                                                                   |
|   //@ WARNING: ALWAYS MAKE CHANGES IN PARMS(INPUTWEB) AND NOT           |
|   //@          &CNTL(DAILY).                                            |
|   //@          CHANGES MADE TO &CNTL(DAILY)                             |
|   //@          WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN.           |
|   //@                                                                   |
|   //INPUTSMF DD DISP=SHR,DCB=BUFNO=5,DSN=___________ ____________       |
|   //         DD DISP=SHR,DCB=BUFNO=5,DSN=___________ ____________       |
|   //@                                                                   |
|   //SMFDRCTR DD DISP=SHR,DCB=BUFNO=5,DSN=___________ ____________       |
|   //         DD DISP=SHR,DCB=BUFNO=5,DSN=___________ ____________       |
|                                                                         |
|                                                                         |
|                                                                         |
+-------------------------------------------------------------------------+
```

Figure 7-4. INPUTWEB JCL Definition Worksheet

7.3.5 Database Space Modeling (DBMODEL)

This section describes the information required to define the Analyzer to the Database Space Modeling Facility. Specifically, the user must provide values for the cycle (data retention) definitions. The special considerations for defining the FILE input statements to the Database Space Modeling facility are discussed.

Section 2.3.4 of the PIOM, Database Space Modeling Facility, explains how these values are used in estimating the DASD requirements of the database, and how the user can input these values to the modeling facility.

This section covers the following topics:

1 - Data Retention Specifications (FILE Statements)

7.3.5.1 Data Retention Specifications (FILE Statements)

This section describes cycle (data retention) definitions required for the CA MICS Web Analyzer, discusses how to determine the Web data cycles required, and provides instructions for completing the FILE statements in the PARMS member DBMODEL. Figure 7-5 is the worksheet to help you collect this information.

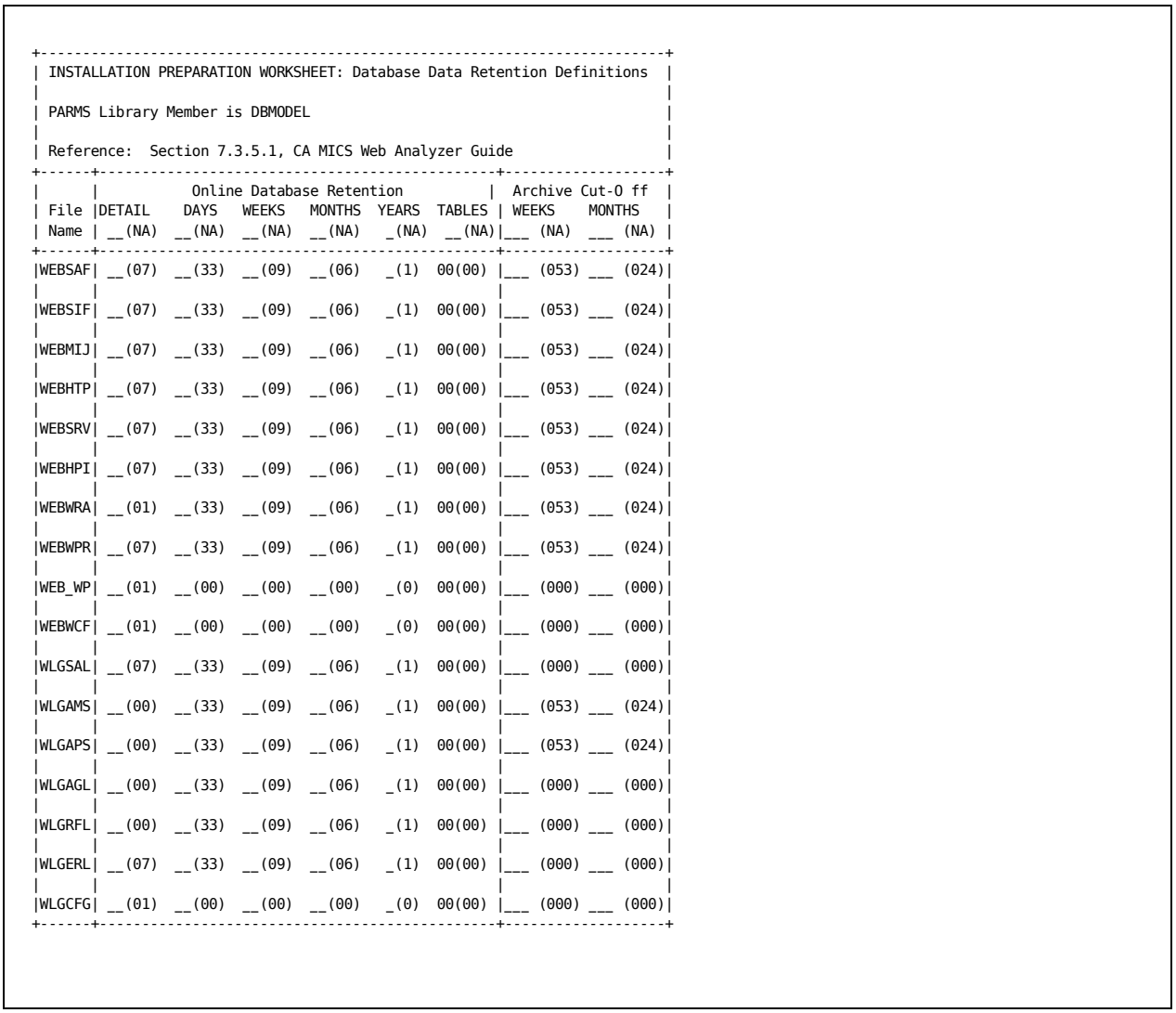
The worksheet is organized by information area. Each file in the area is listed by name. For each file, a line is formatted to allow six definitions in the online database and two in the archive database:

- The online database files quantify the number of cycles of data that is maintained in the DETAIL, DAYS, WEEKS, MONTHS, and YEARS timespans and the TABLES data area.

- The two definitions for the archive database files quantify the number of cycles of data to be retained, up to the cutoff limit defined. The archive definitions have no impact on the size of the database and can be specified whethert the weekly and/or monthly archive history files have actually been activated (see Section 2.3.3, CA MICS JCL Planning and Parameters, of the PIOM).

The worksheet formats provide an underscored area for the user's definition, followed by the recommended value, shown within parentheses. If the underscored area contains a value of 00, the file is not supported for the indicated timespan. To add support, perform database tailoring as described in Section 6.2, Tailoring the Database, of the System Modification Guide (SMG).

When specifying a retention limit, remember that the number may never be zero if the file has been defined to be active in the timespan.



INSTALLATION PREPARATION WORKSHEET: Database Data Retention Definitions

PARMS Library Member is DBMODEL

Reference: Section 7.3.5.1, CA MICS Web Analyzer Guide

File Name	Online Database Retention DETAIL DAYS (NA)	WEEKS (NA)	MONTHS (NA)	YEARS (NA)	TABLES (NA)	Archive Cut-Off WEEKS (NA)	MONTHS (NA)
WEBSAF	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WEBSIF	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WEBMIJ	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WEBHTP	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WEBSRV	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WEBHPI	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WEBWRA	__(01)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WEBWPR	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WEB_WP	__(01)	__(00)	__(00)	__(00)	_(0)	00(00)	___ (000)	___ (000)
WEBWCF	__(01)	__(00)	__(00)	__(00)	_(0)	00(00)	___ (000)	___ (000)
WLGSAL	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (000)	___ (000)
WLGAMS	__(00)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WLGAPS	__(00)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (053)	___ (024)
WLGAGL	__(00)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (000)	___ (000)
WLGRFL	__(00)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (000)	___ (000)
WLGERL	__(07)	__(33)	__(09)	__(06)	_(1)	00(00)	___ (000)	___ (000)
WLGCFG	__(01)	__(00)	__(00)	__(00)	_(0)	00(00)	___ (000)	___ (000)

Figure 7-5. Data Retention Specifications Worksheet (FILE Statement)

Chapter 8: INSTALLATION

After specifying the parameters documented in Chapter 7 of this guide, you can install the CA MICS Web Analyzer using the checklists in Section 3.8 of the PIOM.

Be sure to review Chapters 6 and 7 of this guide before proceeding to the installation checklists in the PIOM. These chapters provide helpful information about the data sources and the input parameters you will need to supply for proper operation of the CA MICS Web Analyzer.

In order to collect WebSphere HTTP Server Logs data, you must install the WebSphere Log Extractor in your UNIX System Services environment.

This section contains the following topics:

8.1 Installing the Web Log Extractor

The Web Log Extractor is delivered in library sharedprefix.MICS.BIN(WLETAR) file. To install this routine, you must run the job sharedprefix.MICS.CNTL(WLEINST). This job inflates the WLETAR file in the defined Unix System Services directory. Three files are created: WLELOAD.EXE, WLELOAD.CONF, and WLELOAD.SCP. See Appendix C of this guide for complete usage and installation instructions.

A sample job to execute the Web Log Extractor on a daily basis is delivered with the CA MICS Web Analyzer. This job is located in prefix.MICS.CNTL(WLERUN) and can be run from an OS/390 system.

Chapter 9: PROCESSING

The processing information in this chapter provides an overview of how data from the CA MICS Web Analyzer becomes an integrated part of the CA MICS database.

This section contains the following topics:

9.1 Processing Overview

The CA MICS Web Analyzer makes use of the standard database update and summarization facilities of CA MICS to maintain its data in the online and offline databases. The SMF data and log data is supplied to CA MICS in the DAY052 step of the DAILY database update job.

The daily update step:

- Reads and formats the raw data
- Maintains data integrity by eliminating duplicate data
- Summarizes activity
- Adds new cycles to the DETAIL and DAYS timespans
- Updates the week-to-date and month-to-date cycles

A more detailed description of DAY052 processing is presented in the next section.

Weekly processing is performed by the WEEK052 and WEEK300 steps. The WEEK052 step splits the week-to-date files into a new week-to-date and a new weekly '01' cycle. The WEEK300 step creates the weekly offline database files.

Monthly processing is performed by the MONTH052 and MONTH300 steps. The MONTH052 step splits the month-to-date file into a new month-to-date and a new monthly '01' cycle. It also appends the data to the year-to-date file. The MONTH300 step creates the monthly offline database files.

The optional yearly processing is performed by the YEAR052 step which splits the year-to-date files into a new year-to-date and a new yearly '01' cycle.

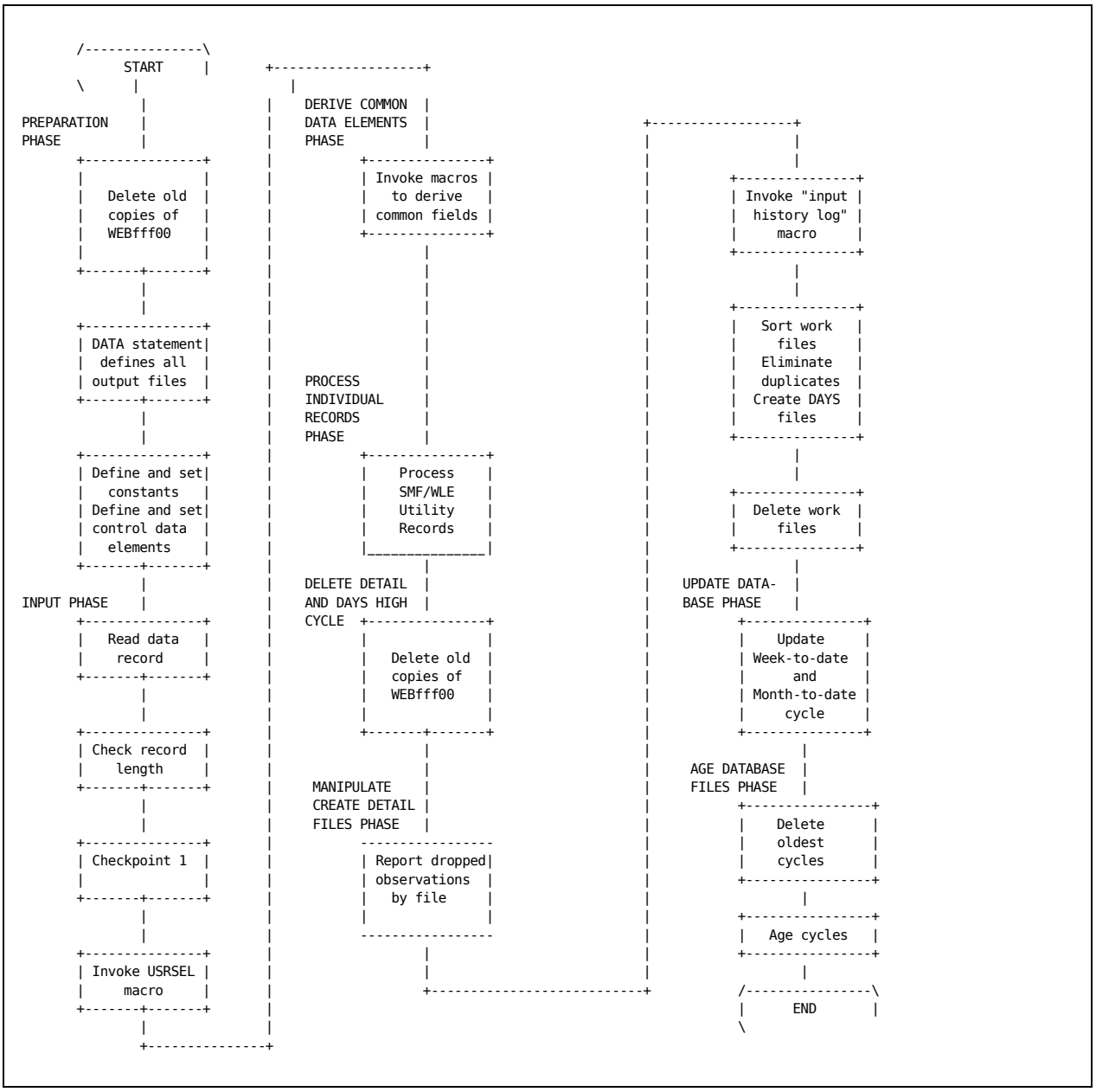


Figure 9-1. Processing Flow Diagram

9.2 Daily Update Processing Flow

The DAY052 step of the DAILY job processes SMF type 103 and type 120 records and WLE utility generated log records.

The SMF type 103 records contain web server configuration and performance information. The SMF records are cut on a predefined interval through the HTTPD.CONF configuration file of IBM's WebSphere Application Server for OS/390 (defaulting to 15 minutes).

The SMF type 120 records contain WebSphere Application Server (WAS) activity information. The server activity is recorded at detail as well as interval level. The interval records are cut at the installation defined interval

Step DAY052 consists of the following phases:

1 - Preparation Phase
2 - Input Phase
3 - Decoding Phase
4 - Data Input Phase
5 - Data Summarization Phase
6 - File Aging

9.2.1 Preparation Phase

During the Preparation Phase the CA MICS Web Analyzer defines its output files with a DATA statement. Data elements, formats and constants are defined and set to their initial values.

9.2.2 Input Phase

During the Input Phase, the CA MICS Web Analyzer reads the SMF and log data. At End-of-File, a count of statistics is printed. The counts will indicate the total records kept for the WEB and WLG Information Areas. The input phase will also give the total short records deleted, total records options deleted, and total records date ranges deleted. In addition, a count of records processed for each type of log file is printed.

9.2.3 Decoding Phase

During the Decoding Phase, the CA MICS Web Analyzer checks the record type , record subtype and transfers control to the appropriate processing code.

Process begins with SMF data and follows through to log data once the end of SMF data is reached.

By default, the Web Information Area (WEB) files are always created but the Web Log Information Area (WLG) files are created only if the WLG option is coded on the OPTION statement in sharedprefix.MICS.GENLIB(WEBGENIN).

9.2.4 Data Input Phase

During the Data Input Phase, the CA MICS Web Analyzer reads the data elements from the various record types and performs computations to scale and accumulate data.

9.2.5 Data Summarization Phase

The DAY052 step creates current cycles for both the DETAIL and DAYS files. It also appends data to both the week-to-date and month-to-date files.

9.2.6 File Aging

CA MICS Web Analyzer files are aged to delete the oldest cycles and rename work files to current cycles. There are no exit points provided in this phase.

Chapter 10: MODIFICATION

This chapter explains the user exit points that are provided with the CA MICS Web Analyzer. You must carefully explore whether it is necessary to develop any user exit routines. To this end, first ensure that the requirement cannot be satisfied through one of the standard options or definitions. This process can be greatly facilitated by discussing the requirement with CA Technical Support.

At times, additional system customizing is needed to tailor CA MICS to meet your site's needs. There are many user exit points provided within CA MICS. Each of these points corresponds to a position in the CA MICS logic where user modification:

- Is likely to occur, based on our experience
- Presents only a slight possibility of compromising CA MICS database or operational integrity
- Is relatively easy to document and understand

Using the CA MICS exit facilities to augment CA MICS processing logic is the safest method of system modification. Generally, we consider that the CA MICS system administrator has four levels or methods by which CA MICS can be augmented, changed, and enhanced:

- CA MICS Standard Option and Parameter Definitions

  The diverse installation and definition options provide most users with adequate provisions for customizing CA MICS.

- Modification through User Exit Points

  There may be site-dependent requirements that are not adequately addressed through the standard options and parameters of CA MICS. In these cases, one or more of the user exit points available to you may be used to code user written routines that would satisfy these extended requirements.

o Extension through Field Developed Applications

  Although most users do not initially consider this a way of modifying CA MICS, the development of field-developed applications may be the most advantageous method of addressing the need to modify a CA MICS product. For example, it may be better to write an application to handle additional SMF data than to attempt to modify the logic of the CA MICS Web Analyzer.

o System Code Modification

  As a last resort, you may be able to satisfy your requirements by implementing a source code change.

Before attempting to activate an exit, you should read and fully understand the information contained in Section 4.3 of the CA MICS System Modification Guide, User Exit Facilities. The following material is intended to supplement, not replace, that discussion.

The design, coding, testing, and implementation of CA MICS user exit routines should be approached with caution as errors may result, causing the corruption of data. You should be meticulous in the definition and validation of the exit routines to ensure that system integrity and performance have not been adversely affected.

The following sections identify the user exit points that are available for CA MICS Web Analyzer processing and describe in detail each of the user exit points:

This section contains the following topics:

10.1 Available User Exits

There are two types of user exits that can be used to modify CA MICS logic: general exits, which apply to all the installed products, and product exits, which apply only to the product for which they are provided.

General exits allow user routines to be used for all products that process the raw SMF data. General exits are briefly described in this guide, and a more detailed description can be found in Chapter 4 of the CA MICS System Modification Guide.

Product exits are classified according to their purpose and include input, work file, output, parameter-related, and accounting exits.

- Input exits are invoked as the raw input records are read. Output exits are invoked just before CA MICS writes an observation to the CA MICS database.

- The CA MICS Web Analyzer provides the general input exit _USRSEL, which may be used to select or exclude data meeting special site-defined criteria, to add data elements, or to modify record fields as they are read. In addition, the general exit, _USRIHL, is provided at the end of input processing to permit examination of the CA MICS Input History Log SAS file so that the user can terminate the DAILY run if critical data is missing.

- An output exit routine is provided for each file and may be used to selectively block the writing of specified records, alter data elements in the records, or produce additional records for the target file or a user-defined file.

- Parameter-related exits are also user exits, but are defined as part of the standard CA MICS installation process. They are normally associated with other parameters or options. For example, the Account Code Derivation Routine is a user exit routine for completing values for account codes that were defined in a separate but related member of prefix.MICS.PARMS, WEBACRT, the Account Code Definition parameter member. These exits are identified in both this chapter and in Chapter 7.

- Accounting exits are described in the CA MICS Accounting and Chargeback User Guide.

The user exits for the CA MICS Web Analyzer are identified below:

- Product Input Exits

  _USRSEL - File Processing Selection
  _USRIHL - Examine Input History Log

- Product Output Exits

  USRSWPR - HTTP Performance File
  USRSSAF - WAS Server Activity File
  USRSSIF - WAS Server Interval File
  USRSMIJ - J2EE Method Interval File
  USRSHTP - HTTP Session Manager Interval File
  USRSSRV - WebContainer Servlet Interval File
  USRSHPI - JVM Heap Interval File
  USRSWCB - Web CPU Usage Breakdown File
  USRSWRA - Web Request Activity File
  USRSWUD - Web User Data File
  USRSWAA - Web Asynch Activity File
  USRSAGL - Agent Log File
  USRSERL - Error Messages Log File
  USRSRFL - Referer Log File
  USRSSAL - Server Access Log File

- Product Parameter-Related Exits

  WEBACRT - Web Account Code Derivation Exit

10.2 Exit Routine Considerations

There are data elements available for use by the user exits that vary according to the processing phase and the particular user exit point. Special data elements that are used in the CA MICS Web Analyzer update process but are not stored in the CA MICS database (and therefore have no corresponding Data Dictionary description) are identified below. The individual user exit descriptions list which of these elements are available for the user exit routine.

ROUTINE - Name of the Invoking Routine

FILEID - Identifier of the File Being Processed

SMFRTYPE - SMF Record Type of the Record Being Processed

COMPT - 8-Character Product and Data Source Identifier

10.3 Input Exits

This section provides a description of the user exits that are invoked during the first phase of the CA MICS Web Analyzer DAILY update processing. The exits are listed alphabetically.

Each exit description includes the name and title, a description of its purpose, when it is invoked, and whether it has an interface to CA MICS Accounting and Chargeback. In addition, it also shows which data elements are available, the special considerations to note, and a sample user exit.

```
+---------------+
| _ U S R S E L |  File Processing Selection Exit
+---------------+
```

DESCRIPTION: This exit allows access to each record processed by the CA MICS Web Analyzer input format routine.

INVOCATION: The exit gains control after a detail input record has been read.

ACCOUNTING INTERFACE: No interface to CA MICS Accounting and Chargeback.

```
USES:  This exit can be used to exclude records from being
processed by the CA MICS Web Analyzer during the daily input
step.  To exclude a record, set SKIP_REC to one.  Refer to
the example later in this section for help on coding.

ELEMENTS AVAILABLE:

  ROUTINE - The name of the routine that invoked this exit.

    To use this exit in the CA MICS Web Analyzer process,
    test for ROUTINE = 'DYWEBFMT'.

  SMFRTYPE - The record type of the input record:

    103 - HTTP Server Performance Statistics
    120 - WebSphere Application Server Statistics

  ORGSYSID - Original SYSID from SMF record.

  ENDTS - End Timestamp

    In all files except the MQBMFA file, ENDTS is set to the
    value in the SMF header that tells the time and date the
    record was written to the SMF buffer, for example
    SM116DTE and SM116TME.

    In the MQBMFA file, ENDTS is set to the interval end date
    and time values found in the message flow accounting
    record fields IMFLENDT and IMFLENTM.

  COMPT - Component ID

    The 8-character value representing CA MICS Web Analyzer
    product ID, either 'WEB SMF ' for SMF data or 'WEB WLG '
    for HTTP Web Server Log data.  It identifies the product
    data source.
```

```
CODING RESTRICTIONS:  Refer to the CA MICS System
  Modification Guide, section 4.3.2.1.

SPECIAL NOTES:

  1. This exit is part of the CA MICS Base component and is
     located in sharedprefix.MICS.SOURCE(#BASEXIT).  However,
     it is recommended that the user modify
     prefix.MICS.USER.SOURCE(#BASEXIT).  A more detailed
     description of this routine is provided in the CA MICS
     System Modification Guide, section 4.3.2.1, Standard
     CA MICS Exits.

  2. Because this exit is used in many places, it is
     necessary to qualify all coding in this exit by
     examining the name of the routine in which the exit was
     invoked.  The program variable ROUTINE exists for this
     purpose.  See the example below.

SAMPLE USER EXIT:

  In this exit, the data logged by WebSphere Web Server CSQ2
  is dropped from further processing.  This data may be
  coming from a test system that does not need to be kept in
  the production database.

  MACRO _USRSEL

  IF ROUTINE = 'DYWEBFMT' THEN DO;
    IF SYSID = 'CSQ2' THEN
       SKIP_REC = 1;
  END;
  %


+---------------+
| _ U S R I H L |  Examine Input History Log Exit
+---------------+

DESCRIPTION:  This exit permits inspection of the DETAIL
timespan checkpoint SAS file immediately following processing
of all input records in the input format routine.

INVOCATION:  This exit is referenced in code that passes the
DETAIL timespan checkpoint SAS file.  The code reads
_ADMX.CKPTDATA as an integrity check on that file, and in the
process allows the user to inspect the file's contents.
```

```
ACCOUNTING INTERFACE:  No interface to CA MICS Accounting and
Chargeback.

USES:  This exit can be used to abort daily update processing
based on an extended checkpoint examination algorithm.  For
example, the daily process will end normally if data from the
SMF or WLE log file is provided.  However, it may be critical
that data from at least one specific system is provided for a
successful update.  This routine can be used to examine all
entries in _ADMX.CKPTDATA to test for the required condition.
If the appropriate data is not found, the CA MICS Web
Analyzer DAILY step could be terminated with an ABORT ABEND
statement.

ELEMENTS AVAILABLE:

  All elements in the ADMIHL File

CODING RESTRICTIONS:  Refer to the CA MICS System
  Modification Guide, section 4.3.2.1.

SPECIAL NOTES:

  1. This exit is part of the CA MICS Base component and is
     located in sharedprefix.MICS.SOURCE(#BASEXIT).  However,
     it is highly recommended that the user modify
     prefix.MICS.USER.SOURCE(#BASEXIT).  A more detailed
     description of this routine is provided in the CA MICS
     System Modification Guide, section 4.3.2.1, Standard
     CA MICS Exits.

  2. Because this exit is used in many places, it is
     necessary to qualify all coding in this exit by
     examining the name of the routine in which the exit was
     invoked.  The program variable ROUTINE exists for this
     purpose.  Refer to the following example.

  3. Note that COMPT has a special meaning to the CA MICS Web
     Analyzer, since it not only identifies the product but
     also the source of data (SMF or log data), as follows:

        'WEB SMF ' - The data is from SMF
        'WEB WLG ' - The data is from the log file.
```

```
SAMPLE USER EXIT:

In this sample exit, the Input History Log is examined to
determine if input data was received for the production SYSID
TST1.  When End of File (EOF) is reached for the DAILY
update run, a check is done to determine if TST1 data was
provided.  If TST1 data was not provided, then further DAILY
processing is aborted and a message is written specifying the
reason for abort.

MACRO _USRIHL
 IF ROUTINE = 'DYWEBFMT' THEN DO;
 RETAIN TST1FLAG 0;
 IF SYSID = 'TST1' THEN
    TST1FLAG = 1;
 IF EOF THEN DO;
   IF TST1FLAG THEN GOTO ENDCHECK;
   PUT 'Data from Production System TST1 missing';
   ABORT ABEND;
 END;
 ENDCHECK:
 END;
%
```

10.4 Output Exits

This section provides an in-depth description of the CA MICS Web Analyzer standard output processing exits that are invoked during phase 2 of the DAILY update processing flow. The exits are organized by information area.

Each exit description includes the user exit name, title, a description of its purpose, when it is invoked, and whether it has an interface to CA MICS Accounting and Chargeback. It also shows what data elements are available and any special considerations to be aware of.

```
+---------+
| USRSWPR |  WebSphere Performance File Exit
+---------+
```

DESCRIPTION: USRSWPR allows access to the data elements used to build the records in the WebSphere Performance File.

INVOCATION: This exit gains control immediately prior to the output of records to the Web Performance File.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add or change data elements, or to delete observations from the WebSphere Performance File. This file is built from the SMF Type 103 Records (subtype 2). The location of the self-defining section of the WebSphere Performance data is found at offset 24 ('18'X).

ELEMENTS AVAILABLE: All elements in the WebSphere Performance File.

CODING RESTRICTIONS: Refer to the CA MICS System Modification Guide, Section 4.3.2.2.

```
+---------+
| USRSSAF |  WAS Server Activity File
+---------+
```

DESCRIPTION: USRSSAF allows access to the data elements used to build the records in the WAS Server Activity File.

INVOCATION: This exit gains control immediately prior to the output of records to the WAS Server Activity File.

ACCOUNTING INTERFACE: The CA MICS Accounting and Chargeback interface follows the invocation of this user exit in a later

```
step while creating the WEBSAF file in the DETAIL database.
If you decide to eliminate data, you may affect the content
of the WEBSAF file and the accounting information for your
system.

USES:  This exit is used to add or change data elements, or
to delete observations from the WAS Server Activity File.
This file is built from the SMF Type 120 Records (subtype 1).
The location of the self-defining section of the WAS Server
Activity File is found at offset 40 ('28'X).

ELEMENTS AVAILABLE:  All elements in the WAS Server Activity
File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSSIF |  WAS Server Interval File
+---------+

DESCRIPTION:  USRSSIF allows access to the data elements used
to build the records in the WAS Server Interval File.

INVOCATION:  This exit gains control immediately prior to the
output of records to the WAS Server Interval File.

ACCOUNTING INTERFACE:  No interface is provided.

USES:  This exit is used to add or change data elements, or
to delete observations from the WAS Server Interval File.
This file is built from the SMF Type 120 Records (subtype 3).
The location of the self-defining section of the WAS Server
Interval File is found at offset 40 ('28'X).

ELEMENTS AVAILABLE:  All elements in the WAS Server Interval
File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSMIJ |  J2EE Method Interval File
+---------+

DESCRIPTION:  USRSMIJ allows access to the data elements used
to build the records in the J2EE Method Interval File.

INVOCATION:  This exit gains control immediately prior to the
```

```
output of records to the J2EE Method Interval File.

ACCOUNTING INTERFACE:  No interface is provided.

USES:  This exit is used to add or change data elements, or
to delete observations from the J2EE Method Interval File.
This file is built from the SMF Type 120 subtype 6 records.
The location of the self-defining section of the J2EE Method
Interval File is found at offset 40 ('28'X).


ELEMENTS AVAILABLE:  All elements in the J2EE Method Interval
File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSHTP |  HTTP Session Manager Interval File
+---------+

DESCRIPTION:  USRSHTP allows access to the data elements used
to build the records in the J2EE Method Interval File.

INVOCATION:  This exit gains control immediately prior to the
output of records to the HTTP Session Manager Interval File.

ACCOUNTING INTERFACE:  No interface is provided.

USES:  This exit is used to add or change data elements, or
to delete observations from the HTTP Session Manager Interval
File.  This file is built from the SMF Type 120 subtype 8
records.  The location of the self-defining section of the
HttpSessionManager section is found at offset 52 ('28'X).

ELEMENTS AVAILABLE:  All elements in the HTTP Session Manager
Interval File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSSRV |  WebContainer Servlet Interval File
+---------+

DESCRIPTION:  USRSSRV allows access to the data elements used
to build the records in the WebContainer Servlet Interval
File.
```

INVOCATION: This exit gains control immediately prior to the output of records to the WebContainer Servlet Interval File.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add or change data elements, or to delete observations from the WebContainer Servlet Interval File. This file is built from the SMF Type 120 subtype 8 records. The location of the self-defining section of the Servlet section is found at offset 264 ('108'X) inside the WebApplication interval section that is found at offset 64 ('40'X).

ELEMENTS AVAILABLE: All elements in the WebContainer Servlet Interval File.

CODING RESTRICTIONS: Refer to the CA MICS System Modification Guide, Section 4.3.2.2.

```
+---------+
| USRSHPI |  JVM Heap Interval File
+---------+
```

DESCRIPTION: USRSHPI allows access to the data elements used to build the records in the JVM Heap Interval File.

INVOCATION: This exit gains control immediately prior to the output of records to the JVM Heap Interval File.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add or change data elements, or to delete observations from the JVM Heap Interval File. This file is built from the SMF Type 120 subtype 3 records. The location of the self-defining section of the JVM heap section is found at offset 8 ('8'X) inside the Server region section that is found at offset 52 ('34'X).

ELEMENTS AVAILABLE: All elements in the WebContainer Servlet Interval File.

CODING RESTRICTIONS: Refer to the CA MICS System Modification Guide, Section 4.3.2.2.

```
+---------+
| USRSWCB |  Web Request Activity File
+---------+
```

DESCRIPTION: USRSWCB allows access to the data elements used

to build the records in the Web CPU Usage Breakdown (WEBWCB) file.

INVOCATION: This exit gains control immediately prior to the output of records to the Web CPU Usage Breakdown File.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add or change data elements, or to delete observations from the Web CPU Usage Breakdown File. This file is built from the SMF Type 120 subtype 9 records.

ELEMENTS AVAILABLE: All elements in the Web CPU Usage Breakdown File.

CODING RESTRICTIONS: Refer to the CA MICS System Modification Guide, Section 4.3.2.2.

```
+---------+
| USRSWRA |  Web Request Activity File
+---------+
```

DESCRIPTION: USRSWRA allows access to the data elements used to build the records in the Web Request Activity (WEBWRA) file.

INVOCATION: This exit gains control immediately prior to the output of records to the Web Request Activity File.

ACCOUNTING INTERFACE: The CA MICS Accounting and Chargeback interface follows the invocation of this user exit in a later step while creating the WEBWRA file in the DAYS database.

USES: This exit is used to add or change data elements, or to delete observations from the Web Request Activity File. This file is built from the SMF Type 120 subtype 9 records.

ELEMENTS AVAILABLE: All elements in the Web Request Activity File.

CODING RESTRICTIONS: Refer to the CA MICS System Modification Guide, Section 4.3.2.2.

```
+---------+
| USRSWUD |  Web User Data File
+---------+
```

DESCRIPTION: USRSWUD allows access to the data elements used to build the records in the Web User Data (WEBWUD) file.

```
INVOCATION:  This exit gains control immediately prior to the
output of records to the Web User Data File.

ACCOUNTING INTERFACE:  No interface is provided.

USES:  This exit is used to add or change data elements, or
to delete observations from the Web User Data File.  This
file is built from the SMF Type 120 subtype 9 records.

ELEMENTS AVAILABLE:  All elements in the Web User Data File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSWAA |  Web Asynch Activity File
+---------+

DESCRIPTION:  USRSWAA allows access to the data elements used
to build the records in the Web Asynch Activity (WEBWAA)
file.

INVOCATION:  This exit gains control immediately prior to the
output of records to the Web Asynch Activity File.

ACCOUNTING INTERFACE:  The CA MICS Accounting and Chargeback
interface follows the invocation of this user exit in a later
step while creating the WEBWAA file in the DAYS database.

USES:  This exit is used to add or change data elements, or
to delete observations from the Web Asynch Activity File.
This file is built from the SMF Type 120 subtype 9 records.

ELEMENTS AVAILABLE:  All elements in the Web Asynch Activity
File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSAGL |  Agent Log File Exit
+---------+

DESCRIPTION:  USRSAGL allows access to the data elements used
to build the records in the Agent Log File.

INVOCATION:  This exit gains control immediately prior to the
output of records to the Agent Log File.
```

```
ACCOUNTING INTERFACE:  No interface is provided.

USES:  This exit is used to add or change data elements, or
to delete observations from the Agent Log File Exit.  This
file is built from the WebSphere Agent Log data.


ELEMENTS AVAILABLE:  All elements in the Agent Log File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSERL |  Error Messages Log File Exit
+---------+

DESCRIPTION:  USRSERL allows access to the data elements used
to build the records in the Error Messages Log File.

INVOCATION:  This exit gains control immediately prior to the
output of records to the Error Messages Log File.

ACCOUNTING INTERFACE:  No interface is provided.

USES:  This exit is used to add or change data elements, or
to delete observations from the Error Messages Log File
Exit.  This file is built from the WebSphere Error Log data.


ELEMENTS AVAILABLE:  All elements in the Error Messages Log
File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSRFL |  Referer Log File Exit
+---------+

DESCRIPTION:  USRSRFL allows access to the data elements used
to build the records in the Referer Log File.

INVOCATION:  This exit gains control immediately prior to the
output of records to the Referer Log File.

ACCOUNTING INTERFACE:  No interface is provided.

USES:  This exit is used to add or change data elements, or
```

```
to delete observations from the Referer Log File Exit.  This
file is built from the WebSphere Referer Log data.

ELEMENTS AVAILABLE:  All elements in the Referer Log File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+---------+
| USRSSAL |  Server Access Log File Exit
+---------+

DESCRIPTION:  USRSSAL allows access to the data elements used
to build the records in the Server Access Log File.

INVOCATION:  This exit gains control immediately prior to the
output of records to the Server Access Log File.

ACCOUNTING INTERFACE:  The CA MICS Accounting and Chargeback
interface follows the invocation of this user exit in a later
step while creating the WLGSAL file in the DETAIL database.
If you decide to eliminate data, you may affect the content
of the WLGSAL file and the accounting information for your
system.

USES:  This exit is used to add or change data elements, or
to delete observations from the Server Access Log File Exit.
This file is built from the WebSphere Server Access Log, FRCA
Access Log, Proxy Access Log, and Proxy (Cache) Log data.

ELEMENTS AVAILABLE:  All elements in the Server Access Log
File.

CODING RESTRICTIONS:  Refer to the CA MICS System
Modification Guide, Section 4.3.2.2.

+----------------+
|  U S R D f f f |  DB2 User-written Cost Derivation Exit
+----------------+

DESCRIPTION:  The USRDfff exit lets you provide the
derivation of the fffCOST data element.  The files available
(fff) are the WEBWRA and WEBWAA files.

INVOCATION:  These exits are invoked in the information area
processing phase immediately before output of the file.

ACCOUNTING INTERFACE:  No interface is provided.
```

```
USES:  The exit allows elements to be passed to the routine
computing the value of fffCOST.

ELEMENTS AVAILABLE:  All elements in the file are available.

+--------------+
| U S R X f f f |  DETAIL Only File Exits
+--------------+

DESCRIPTION:  USRXfff exit is available for the Web Request
Activity (WEBWRA), Web CPU Usage Breakdown (WEBWCB), WEB User
Data (WEBWUD), and the Web Asynch Activity (WEBWAA) files.

- The USRXWRA exit is invoked during the final output of the
  Web Request Activity (WEBWRA) File.

- The USRXWCB exit is invoked during the final output of the
  Web CPU Usage Breakdown (WEBWCB) File.

- The USRXWUD exit is invoked during the final output of the
  Web User Data (WEBWUD) File.

- The USRXWAA exit is invoked during the final output of the
  Web Asynch Activity (WEBWRA) File.

User code in the USRXWRA, USRXWCB, USRXWUD, and USRXWAA exits
only modifies observations that are output to the DETAIL
timespans for the WEBWRA, WEBWCB, WEBWUD, and WEBWAA files.
These modifications are not propagated to the DAYS and higher
timespans.

For example, using the USRXQAA exit to restrict output of
queue level records, where the queue name begins with
"customer," to the DETAIL MQAQAA01 Queue Activity File has no
impact on the observations summarized in the DAYS and higher
timespan MQAQAA files.

INVOCATION:  Exit gains control in the SAS data step that
creates the DETAIL timespan WEBfff00 file in
sharedprefix.MICS.SOURCE(WEB1204).

ACCOUNTING INTERFACE:  No interface to CA MICS Accounting and
Chargeback.

USES:  The USRXfff exits allow subsetting and data element
modification that is restricted to the DETAIL timespan
WEBWRA, WEBWCB, WEBWUD, and WEBWAA files.  Due to data volume
considerations, the WEBWRA, WEBWUD, and WEBWAA files are not
active by default in the DETAIL timespan.  Observations can
```

```
be excluded by setting SKIP_REC=1.

If you choose to create a DETAIL timespan file for the
WEBWRA, WEBWUD, and WEBWAA files on tape rather than on DASD,
use the USRTfff exit.

ELEMENTS AVAILABLE:  All elements in the respective file.

CODING RESTRICTIONS:  Refer to the System Modification Guide,
Section 4.3.2.2.

SPECIAL NOTES:  To implement this exit at the unit level,
code it in prefix.MICS.USER.SOURCE(#WEBEXIT).

SAMPLE USER EXIT - USRXWRA:

DESCRIPTION:  This sample exit lists individual, DETAIL-level
queue segments for all queues associated with the XYZ
application.

/*                                                  */
/* Limit output of Detail Web Request Activity to   */
/* a particular Websphere Application Server        */
/*                                                  */
IF WEBSVRNM NE: 'WSVR42X9' THEN SKIP_REC=1 ;

+----------------+
|  U S R T f f f | DETAIL Tape Data Selection Exit
+----------------+

DESCRIPTION:  The USRTfff exits provide access to DETAIL file
records prior to their output to DETAIL tape data sets.
Refer to Chapter 7 of this guide for information about DETAIL
tape processing and a list of eligible component files.

INVOCATION:  The USRTfff exit is invoked prior to output to
DETAIL tape files.

ACCOUNTING INTERFACE:  No interface is provided.

USES:  Use this exit to limit the amount of data that is
written to DETAIL tape files, or to modify the contents of
data elements written to DETAIL tape files.  Note that
USRTfff exit code only affects the DETAIL tape file content
for a particular file (fff). It has no impact on the normal
unit database content for the same file (fff).

ELEMENTS AVAILABLE:  All elements in the DETAIL timespan of
the file.
```

ACTIVATION: The USRTfff exits are invoked using the indirect exit convention described in Section 4.3.1.2 of the SMG.

All of the USRTfff indirect exits are activated in sharedprefix.MICS.SOURCE(#cccEXIT). In #cccEXIT, you will find the following statement for each file (fff) eligible for DETAIL tape:

```
%LET USRTfff = ;
```

The exit is activated by providing a member name to the right of the equal sign:

```
%LET USRTfff = exitname ;
```

If you code an exitname, you MUST create a member in each unit's prefix.MICS.USER.SOURCE library with that name. Provide valid SAS code in the member. This code will be invoked prior to the output of records to the DETAIL tape file.

Example:

sharedprefix.MICS.SOURCE(#WEBEXIT):

```
%LET USRTWRA = WRATEXIT ;
```

prefix.MICS.USER.SOURCE(WRATEXIT):

```
IF WEBSRVNM NE 'WSVR42X1' THEN SKIP_REC=1 ;
```

In this example, WRATEXIT was chosen as the name for the DETAIL tape exit for the WEBWRA file. Member WRATEXIT was created in the unit level prefix.MICS.USER.SOURCE library for each unit where DETAIL tape was activated for the WEBWRA file.

The exit was coded to limit the records written to the DETAIL tape file to those from a particular Websphere application server, WSRV42X1.

Note that the exit member name must be identical in each unit's prefix.MICS.USER.SOURCE library, but the contents of each member can be different (or identical) for each unit.

CODING RESTRICTIONS: Ensure that you have activated DETAIL tape processing for a file, and executed cccPGEN prior to activating and coding USRTfff exits.

```
 +----------------+
|  U S R U J f f |  Accounting & Chargeback Exits
|  W E B J f f f |
+----------------+
```

DESCRIPTION: These exits provide the linkage into CA MICS Accounting and Chargeback. The files available (fff) are the WEBWRA and WEBWAA files.

INVOCATION: These exits are invoked in the daily summarization #fffDSUM routines, where fff is WRA or WAA as the data is summarized at the DAYS timespan.

ACCOUNTING INTERFACE: These exits provide the interface to Accounting and Chargeback.

USES: The exit allows elements to be passed to Accounting and Chargeback.

ELEMENTS AVAILABLE: All elements in the file are available.

CODING RESTRICTIONS: Refer to Section 4.3.1.1 of the SMG.

Appendix A: MESSAGES

Messages

This appendix lists all messages generated by the CA MICS Web Analyzer. The messages are printed on both the MICSLOG and SAS log data sets. When reviewing the output of a job, first look at the MICSLOG data set because it contains fewer lines of information; this makes it easier to see whether a job or job step completed as expected. If the MICSLOG message does not provide enough information, consult this appendix for more information about the message. If you need more information, consult the SAS log to help debug the problem.

The messages are listed in ascending numerical sequence and include the full text of the message, the type, the reason for the message, appropriate user action, and applicable references to documentation.

The format of a MICSLOG message is:

```
hh:mm:ss.hs cccnnnnns ttttttttttttttttttttttttttttttttttt
```

where

`hh:mm:ss.hs` is the time that the message was issued.

`cccnnnnns` is the message identifier. The message identifier consists of three parts:

ccc	three letter product identifier
nnnnn	the message number
s	the severity identifier, where I is an informational message, W indicates a warning, and E indicates an error

`ttt...ttt` is the message text. If the text spans more than one line, the message identifier is repeated.

The remainder of this appendix shows the MICSLOG messages produced for the CA MICS Web Analyzer. In addition to the message identifier and text, this section presents the reason for the message, suggests actions to take to resolve the problem, and refers you to additional sources of information to help you understand the problem and its resolution.

+----------+
| WEB00100 |
+----------+

TEXT: Starting DELTA PROCESS for: %FILENAME

TYPE: Informational

REASON: Many of the WebSphere data elements read from the raw data are accumulated values since the server started. This message indicates that the delta routines are about to be entered for the file listed. When done, MICSLOG message WEB00199 is displayed.

ACTION: None

REFERENCES: None

+----------+
| WEB00102 |
+----------+

TEXT: Suspend record deleted due to age: SYSID=%SYSID; WEBIP=%WEBIP; ENDTS=%ENDTS; WEBSUSCT=%SUSLIMIT

TYPE: Informational

REASON: Suspend records are deleted from the suspend file for web servers (identified uniquely by WEBIP and WEBPORT) not encountered for the number of days defined by the SUSLIMIT (default 7) statement in prefix.MICS.PARMS(WEBOPS). Suspend records hold the values last encountered for a WEBIP in the DAILY update and are used to initialize the lag variables for delta calculations during the next DAILY run. Web server observations not encountered again for

```
                the default of 7 days are dropped from the
                suspend (WEB_WP) file.

   ACTION:      None

   REFERENCES:  None

+----------+
| WEB00106 |
+----------+

   TEXT:        No suspend record for:  SYSID=%SYSID;
                WEBIP=%WEBIP  ENDTS=%ENDTS WEBPORT=%WEBPORT

   TYPE:        Informational

   REASON:      No suspend record was found to initialize
                lag variables for delta calculations.  The
                WEBIP is probably new.  Delta processing is
                skipped for the first record.

   ACTION:      None

   REFERENCES:  None


+----------+
| WEB00110 |
+----------+

   TEXT:        More than one day exists between current and
                previous suspend records:
                    SYSID=%SYSID; WEBIP=%WEBIP; ENDTS=%XENDTS
                    LAGENDTS=%XLAGTS;
                    FILE00=%FILE00 SUSPEND=%SUSPEND
                    WEBPORT=%WEBPORT

   TYPE:        Warning

   REASON:      The difference between ENDTS on the current
                record and the previous record is greater
                than 24 hours.

   ACTION:      Determine if this gap in data is correct or
                if data was missing from SMF.  Data does not
                exist unless the WEB Server was active (up
                and running) and recording to SMF.  If data
                was missing but is available, you may want
                to restore the unit database and rerun the
```

```
                daily process with all the input data.

   REFERENCES:  None

+----------+
| WEB00111 |
+----------+

   TEXT:        One or more data elements contained a missing
                value:  FILE=%FILE
                This may be normal if the data element is
                inactive in WEBGENIN or unavailable at this
                WEB release.
                Refer to the SASLOG for the specific data
                elements.

   TYPE:        Warning

   REASON:      Either the temporary variable used to hold
                the previous observation's value or the
                current observation contains a missing value.

   ACTION:      This condition may be normal either if the
                variable is not available at the release of
                WebSphere that you are processing, or if the
                data element has been deactivated in
                WEBGENIN.  Determine which it is, and if this
                field is available and you require it,
                activate it.

   REFERENCES:  None

+----------+
| WEB00112 |
+----------+

   TEXT:        During DELTA PROCESSING, at least one data
                element received a negative value.  Data
                element value set to 0.  FILE=%FILE', Refer
                to the SASLOG for the specific data element.

   TYPE:        Warning

   REASON:      The reason depends on the file noted in the
                message text, but it occurred because the
                preceding observation contained a value that
                was greater than the value in the current
                observation, which is unexpected in
                cumulative records.
```

```
   ACTION:      This negative value might occur if there was
                a break in the raw input data and the web
                server was restarted.  In this case, if the
                missing data contained the SMF 103 subtype 1
                configuration record, the processing would
                not have been triggered and the counters
                would have been reset.  Ensure that there
                were no breaks of time period in the raw
                input data.

   REFERENCES:  None

+----------+
| WEB00121 |
+----------+

   TEXT:        SMF type 120 subtype 6 and/or subtype 8
                records not processed. You must be at SAS
                8.02 or higher to process these records.
                This unit is configured to run with SAS
                release %VR .
                Total number of records rejected:
                                Subtype 6 ===>
                                Subtype 8 ===>

   TYPE:        Warning

   REASON:      This message calls attention to the fact that
                the records listed for subtype 6 and/or
                subtype 8 are rejected because the unit is
                configured with the SAS release that does not
                support an informat $UCS2Xw. This format is
                used to read raw data that is in unicode
                format.

   ACTION:      Upgrade your SAS system software in this unit
                to SAS 8.02 or higher.

   REFERENCES:  None.

+----------+
| WEB00125 |
+----------+

   TEXT:        %LINE

   TYPE:        Informational
```

```
   REASON:      To show all the keywords and their values
                that are in effect for the WEBPGEN run.

   ACTION:      Print the contents of the PARM member.

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00126 |
+----------+

   TEXT:        %SRVNAME - The server name (WEBNAME) is set
                to the concatenation of WEBIP/WEBPORT.
                Assign user defined values for WEBNAME,
                WEBSNAME, and WEBGROUP for this server by
                updating: prefix.MICS.PARMS(WEBSLIST).  Then
                submit prefix.MICS.CNTL(WEBPGEN).'

   TYPE:        Informational

   REASON:      This message displays
                prefix.MICS.PARMS(WEBOPS) statements in
                MICSLOG when prefix.MICS.CNTL(WEBPGEN) is
                executed to generate product unit level code.

   ACTION:      None

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00128 |
+----------+

   TEXT:        %TOKEN1 KEYWORD value not coded. Set to
                DEFAULT of 7.

   TYPE:        Informational

   REASON:      The keyword has a missing parameter value.  A
                default value of 7 is automatically selected
                for this parameter. This message is issued if
                the keyword is coded without the parameter
                value.

   ACTION:      Code a value between 1 and 7 and resubmit
                prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.
```

```
+----------+
| WEB00129 |
+----------+

   TEXT:        %TOKEN1 KEYWORD value out of range.  Must be
                from 1 to 7.

   TYPE:        Error

   REASON:      The keyword contains a suspend value that is
                used to determine whether a suspended
                observation is dropped or kept.  This message
                is issued if the value falls outside the
                range of 1 to 7. The job is terminated.

   ACTION:      Code a value between 1 and 7. Resubmit
                prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.


+----------+
| WEB00130 |
+----------+

   TEXT:        Errors found in prefix.MICS.PARMS(WEBOPS).
                Correct all errors noted in MICSLOG.
                Then RESUBMIT prefix.MICS.CNTL(WEBPGEN).

   TYPE:        Error

   REASON:      One or more of the keywords (CFGLIMIT,
                WCFLIMIT, and SUSPEND) contain an out-of-
                range (less than 1 or greater than 7) value
                that caused the job to terminate.

   ACTION:      Code a desired value for all the keywords in
                error and resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00131 |
+----------+

   TEXT:        GMTOFF for orgsysid %GMTORG is already
                defined. Duplicate statement not permitted.

   TYPE:        Error
```

```
   REASON:      The orgsysid for a GMTOFF statement was
                already used on an earlier GMTOFF statement.
                Only one time offset may be specified for a
                particular system.

   ACTION:      Delete or modify the statement as desired and
                resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00132 |
+----------+

   TEXT:        Missing arguments in above line.  GMTOFF
                statement must be followed by two arguments:
                1-an orgsysid value, and 2-a time offset.

   TYPE:        Error

   REASON:      The GMTOFF statement keyword must be followed
                by exactly two arguments: an orgsysid value
                and a time offset.  At least one of these was
                missing from the line displayed immediately
                above the MICSLOG message.

   ACTION:      Delete or modify the statement as desired and
                resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00133 |
+----------+

   TEXT:        GMTOFF / orgsysid ARRAY size exceeded.
                Current ARRAY size: %CKPTCNT
                Array size controlled by CKPTCNT statement in
                prefix.MICS.PARMS(SITE)

   TYPE:        Error

   REASON:      The capacity of the internal array used to
                hold the time offsets for each orgsysid
                listed in GMTOFF statements was exceeded.

   ACTION:      The current array size is listed in the
                message.  It can be altered by changing the
```

```
                value of the CKPTCNT statement in
                prefix.MICS.PARMS(SITE).  Either increase the
                CKPTCNT size, or code fewer GMTOFF statements
                than the array capacity.

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00134 |
+----------+

   TEXT:        GLOBAL statement not followed by recognized
                keyword.  Invalid keyword: %GTOKEN2'

   TYPE:        Error

   REASON:      The GLOBAL statement can only be followed by
                specific keywords, such as GMTOFF.

   ACTION:      Delete or modify the statement by using a
                valid GLOBAL statement keyword.  Then
                resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00135 |
+----------+

   TEXT:        GLOBAL GMTOFF statement must be followed by a
                numeric time offset.

   TYPE:        Error

   REASON:      The argument following the GLOBAL GMTOFF
                statement must be a numeric time offset.
                Either blanks or a non-numeric argument was
                used instead of a numeric time offset.

   ACTION:      Delete or modify the statement by adding a
                numeric time offset value after GLOBAL
                GMTOFF.  Then resubmit
                prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00136 |
+----------+
```

```
   TEXT:        Time Offset value is invalid in above line.
                Code a value in the range of -13 to +13.

   TYPE:        Error

   REASON:      The time offset value coded in the statement
                immediately above the MICSLOG error message
                is out of the allowable range of -13 to +13.

   ACTION:      Delete or modify the statement by adding or
                correcting the keyword's time offset value.
                Then resubmit prefix.MICS.CNTL(WEBPGEN).

+----------+
| WEB00137 |
+----------+

   TEXT:        Undefined orgsysid value: %ORGSYSID
                on the above GMTOFF statement is not defined
                in prefix.MICS.PARMS(SYSID)
                or sharedprefix.MICS.PARMS(CPLXSID).

   TYPE:        Error

   REASON:      The orgsysid value shown in the error message
                was not defined at the unit or complex level.

   ACTION:      Delete the statement or add the orgsysid to
                either prefix.MICS.PARMS(SYSID) or
                sharedprefix.MICS.PARMS(CPLXSID).  Then
                SUBMIT prefix.MICS.PARMS(BASPGEN).  When
                complete, resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00138 |
+----------+

   TEXT:        Only one GLOBAL GMTOFF statement permitted.
                Duplicate statements found.

   TYPE:        Error

   REASON:      Only one GLOBAL GMTOFF statement is
                allowed.

   ACTION:      Delete the duplicate statement and resubmit
```

```
                prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00139 |
+----------+

   TEXT:        GMTOFF statement must be followed by an
                orgsysid value of 4 bytes or less.
                %GTOKEN2 found instead.

   TYPE:        Error

   REASON:      The orgsysid value coded on the GMTOFF
                statement cannot exceed four bytes in length.
                The erroneous orgsysid is listed with the
                message.

   ACTION:      Delete the statement or modify the orgsysid
                value.  Resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00140 |
+----------+

   TEXT:        %LINE
                No value was coded for the required IP
                ADDRESS/PORT number field (COLS 4-28).
                Statement is ignored.

   TYPE:        Warning

   REASON:      IP ADDRESS/PORT NUMBER column in the
                statement is not specified. This is the
                highest level of information required to
                identify unique observations in the data. In
                the absence of this value, all other values
                specified are meaningless and ignored.

   ACTION:      Specify the IP ADDRESS/PORT NUMBER that is
                associated with other specified values and
                resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
```

```
| WEB00141 |
+----------+

   TEXT:        %IP_PORT
                Duplicate IP ADDRESS/PORT values detected.
                Only one entry per IP ADDRESS/PORT value
                allowed.
                Duplicate IP ADDRESS/PORT values must be
                deleted.

   TYPE:        Error

   REASON:      Multiple IP ADDRESS/PORT values detected.

   ACTION:      Delete duplicated IP ADDRESS/PORT values and
                resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00145 |
+----------+

   TEXT:        %LINE
                IP ADDRESS/PORT value coded, but no values
                coded for optional fields.
                Statement is ignored.

   TYPE:        Warning

   REASON:      At least one of the optional values must be
                coded for the IP ADDRESS/PORT entry column to
                have any meaning.  The absence of any values
                in optional fields results in blank values
                for WEBSNAME, WEBGROUP, and WEBNAME elements
                in the database files. These database
                elements will be blank if no IP ADDRESS/PORT
                entry is present in
                prefix.MICS.PARMS(WEBSLIST).

   ACTION:      1. Remove the entry
                2. Define at least one value in one of the
                   columns after IP ADDRESS/PORT column.

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00148 |
+----------+
```

```
   TEXT:        Errors found in prefix.MICS.PARMS(WEBSLIST).
                Correct all errors noted in MICSLOG.
                Then RESUBMIT prefix.MICS.CNTL(WEBPGEN).

   TYPE:        Error

   REASON:      There are errors in
                prefix.MICS.PARMS(WEBSLIST).

   ACTION:      Correct all errors noted in MICSLOG and
                resubmit prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00150 |
+----------+

   TEXT:        %LINE
                No value coded for element(s):  %BLANKFLD
                These element(s) will contain blank values in
                the WLG INFORMATION AREA files.

   TYPE:        Warning

   REASON:      WEBSNAME (Short name), WEBGROUP (group name),
                and/or WEBNAME (Web Server Name) parameters
                were detected with no values specified. The
                message echoes these parameters for your
                information.  If you choose to take no
                action, you will see blank values in
                corresponding fields in the CA MICS Web
                Analyzer files.

   ACTION:      Optional. Enter valid values for the listed
                parameters and resubmit
                prefix.MICS.CNTL(WEBPGEN).

   REFERENCES:  Chapter 7 in this guide.

+----------+
| WEB00199 |
+----------+

   TEXT:        Ending DELTA PROCESS for:  %FILENAME
                LAG RELOADS :  %RELOADS
                SUSPEND RECORDS AGED :  %SUSDELS
                MISSING SUSPEND RECORDS :  %MISSSUSP
```

```
   TYPE:        Informational

   REASON:      Informational processing summary statistics.

   ACTION:      None

   REFERENCES:  None

+----------+
| WEB00210 |
+----------+

   TEXT:        Host name or IP Address 2 (SALHSTI2) contains
                non-blank value displayed below:
                %SALHSTI2
                The element SALHSTI2 is inactive. Please
                review the product guide, Section 5.3.1.3, to
                determine the corrective actions required.

   TYPE:        Warning

   REASON:      SALHSTIP element is active.  While SALHSTI2
                is inactive, the DAILY processing has
                detected a nonblank value in the element. If
                you use the SALHSTIP element for analysis,
                you should consider activating the SALHSTI2
                element.  Please see the discussion in the
                WLGSAL file Usage Consideration section
                5.3.1.3.

   ACTION:      Activate the SALHSTI2 element as required.
                Activating this element is at the user's
                discretion.

   REFERENCES:  See dictionary elements SALHSTIP and
                SALHSTI2.

+----------+
| WEB00211 |
+----------+

   TEXT:        Host name or IP Address 2 (RFLHSTI2) contains
                non-blank value displayed below:
                %RFLHSTI2
                The element, RFLHSTI2 is inactive.

   TYPE:        Warning
```

```
   REASON:      RFLHSTIP element is active.  While RFLHSTI2
                is inactive, the DAILY processing has
                detected a nonblank value in the element. If
                you use the RFLHSTIP element for analysis,
                you should consider activating the RFLHSTI2
                element.  Please see the discussion in the
                WLGRFL file Usage Consideration section
                5.3.3.3.

   ACTION:      Activate the RFLHSTI2 element as required.
                Activating this element is at the user's
                discretion.

   REFERENCES:  See dictionary elements RFLHSTIP and
                RFLHSTI2.

+----------+
| WEB00250 |
+----------+

   TEXT:        ******************************************
                ******************************************
                WLE LOG records were detected but NOWLG
                OPTION is in effect.  All log data discarded.
                See Section 7.1 for more information.
                ******************************************
                ******************************************

   TYPE:        Warning

   REASON:      The INPUTWEB DD contains a data set with
                records, but NOWLG options are in effect for
                the WEB product.

   ACTION:      Edit sharedprefix.MICS.GENLIB(WEBGENIN) and
                change parameter value from NOWLG to WLG on
                the OPTION statement.

   REFERENCES:  Chapter 7 in this guide.


+----------+
| WEB00251 |
+----------+

   TEXT:        +---+----+----+----+----+----+----+----+----+
                A Total of %WRK_NWEB non-WEB related SMF
                records found in input, and discarded.
                +---+----+----+----++---+----+----+----+----+
```

```
   TYPE:        Informational

   REASON:      The INPUTSMF DD refers to a dataset that
                contains SMF records other than type 103s.

   ACTION:      Consider activating DAYSMF step or any other
                utility at your disposal to preprocess SMF
                data to extract SMF type 103 records in a
                dataset that contains type 103 records only.
                This improves DAY052 processing time.

   REFERENCES:  Chapter 7 in this guide.


+----------+
| WEB00252 |
+----------+

   TEXT:        Total number of records processed for each
                LOG file.
                ******************************************
                  SERVER ACCESS LOG        = %SUB1CNT
                  FRCA ACCESS (CACHE) LOG  = %SUB2CNT
                  PROXY ACCESS LOG         = %SUB3CNT
                  PROXY ACCESS (CACHE) LOG = %SUB4CNT
                  AGENT LOG                = %SUB5CNT
                  REFERER LOG              = %SUB6CNT
                  ERROR LOG                = %SUB7CNT
                  CONFIGURATION            = %SUB9CNT
                ******************************************

   TYPE:        Informational

   REASON:      This message provides detail information
                about the input log record count processed in
                DAY052.

   ACTION:      None

   REFERENCES:  Chapter 9 in this guide.

+----------+
| WEB00253 |
+----------+

   TEXT:        OBSERVATIONS skipped by USER EXITS in the
                main data step.
```

```
                    USRSWPR = %WPREXIT

                    USRSSAL = %SALEXIT
                    USRSAGL = %AGLEXIT
                    USRSERL = %ERLEXIT
                    USRSRFL = %RFLEXIT

                    USRSSAF = %SAFEXIT
                    USRSSIF = %SIFEXIT
                    USRSMIJ = %MIJEXIT
                    USRSHTP = %HTPEXIT
                    USRSSRV = %SRVEXIT
                    USRSHPI = %HPIEXIT
                    ------------------------
                    TOTAL    = %TOTEXIT

   TYPE:        Informational

   REASON:      This message is issued to present runtime
                control counts of the number of observations
                that were deleted by each of the indicated
                file exits in the main DATA step.

   ACTION:      None

   REFERENCES:  None

+----------+
| WEB00254 |
+----------+

   TEXT:        Total number of SMF 120 records processed by
                subtype.

                ************************************
                 Subtype 1 Server Activity Records . . . . =
                    %S1SACNT
                 Subtype 3 Server Interval Records . . . . =
                    %S3SICNT
                 Subtype 5 J2EE Container Records  . . . . =
                    %S5JACNT (unsupported)
                 Subtype 6 J2EE Container Interval Records =
                    %S6JICNT
                 Subtype 7 Web Container Activity Records  =
                    %S7JACNT (unsupported)
                 Subtype 8 Web Container Interval Records  =
                    %S8WICNT
                 Subtype 9 Request Activity Records. . . . =
                    %S9JACNT
```

```
                    Subtype 10 Outbound Request Records (v8)  =
                       %S10ICNT (unsupported)
                   # All subtypes unsupported . . . . . . . . =
                       %SUBUNKWN
                   ************************************
```

TYPE: Informational

REASON: This message is issued to present runtime control counts. The numbers represent the number of SMF type 120 records by subtypes read from the INPUTSMF file.

ACTION: None

REFERENCES: None

```
+----------+
| WEB00257 |
+----------+
```

TEXT: The occurrence of unsupported SMF 120 record subtypes in the input should be investigated.

TYPE: Warning

REASON: This message is issued when the count of unsupported or unknown SMF 120 record subtypes was greater than 0.

ACTION: Check the SMF 120 subtype counts in MICS LOG message WEB00254I. Subtypes 5 and 7 are unsupported by design. If you have a non zero count for subtype 10, Outbound Request records, either CA MICS has not added support for that subtype or you do not have the WEB product change applied that supports that subtype. Contact CA MICS Product Support for information regarding support for the subtype 10 record.

REFERENCES: None.

```
+----------+
| WEB00259 |
+----------+
```

TEXT: A total of %MXARRCNT enclave delta array slots were used out of %DELTARY slots

allocated.

TYPE: Informational

REASON: This message is issued to present the maximum number of delta array slots used during the processing of the raw SMF 120 subtype 9 records.

ACTION: If the maximum slots used approaches the total allocated, consider increasing the margin. See Section 7.3.2.4 for information on specifying the DELTA statement in prefix.PARMS(WEBOPS). An update to WEBOPS requires a WEBPGEN to run to activate the parameter.

REFERENCES: See Section 5.2.4.3 regarding the delta processing performed when enclaves are reused.

```
+----------+
| WEB00260 |
+----------+
```

TEXT: CPU time elements set to missing values for %XARRYFUL WEBWRA observations because internal DELTA arrays exceeded the specified array capacity of %DELTARY entries. Update prefix.MICS.PARMS(WEBOPS) DELTA statement and specify a value greater than %DELTARY, then execute prefix.MICS.PARMS(WEBPGEN).

TYPE: Warning

REASON: This message is issued at the end of the main input processing step to indicate that all slots in the delta array were occupied and more web requests needed to be added to the array.

Because the delta of the CPU times for these records was not possible, the following computed CPU time metrics are set to a missing value:

WRASUPTM - zIIP CPU Time
WRASPNTM - Normalized zIIP CPU Time
WRASUCTM - zIIP Eligible CPU Time on CP

```
                    WRAZAPTM - zAAP CPU Time
                    WRAZPNTM - Normalized zAAP CPU Time
                    WRAZACTM - zAAP Eligible CPU Time on CP
                    WRAXCSTM - CP CPU Time Minus zIIP/zAAP
                               Eligible
                    WRACPUTM - CP CPU Time

                  Note: the CPU times read from the raw input
                  record contain their original values.

   ACTION:        Update the prefix.MICS.PARMS(WEBOPS) DELTA
                  statement and specify a value greater than
                  %DELTARY, then execute
                  prefix.MICS.PARMS(WEBPGEN).

   REFERENCES:    See Section 5.2.4.3 regarding the delta
                  processing performed when enclaves are
                  reused.

+----------+
| WEB00261 |
+----------+

   TEXT:          Asynchronous data section not supported.
                  %ASYNCNT records skipped.
                  Contact CA MICS Product Support for
                  information on support for this data.

   TYPE:          Warning

   REASON:        The Asynchronous data section in the SMF 120
                  subtype 9 record at WAS version 8 is not
                  supported with the initial SMF 120 subtype 9
                  support.

   ACTION:        Contact CA MICS Product Support for
                  information on available maintenance to
                  support this data.

   REFERENCES:    None.

+----------+
| WEB02001 |
+----------+

   TEXT:          No data has passed through initial selection.
                  Check file status and selection criteria.
                  Run terminating with USER ABEND 998.
```

```
   TYPE:        Error

   REASON:      The selection criteria used to subset the
                data may have prevented observations from
                passing through, resulting in zero
                observations in the step.

   ACTION:      Change the selection criteria and rerun the
                inquiry.

   REFERENCES:  None

+----------+
| WEB02002 |
+----------+

   TEXT:        This report is only supported through
                versions V5R2 and higher.
                The highest version detected in the data =
                %WEBSRVVR.

   TYPE:        Error

   REASON:      Elements used in this report are only
                supported at Version 5 Release 2 or higher.
                If you are running a report that has two
                different versions of data within it, the
                highest data detected is used for the element
                WEBSRVVR.

   ACTION:      Run this report with only data that is V5R2
                or higher.  If you attempt to use data
                that is less than V5R2, the job will not fail
                as long as there is additional data being
                input from higher versions.

   REFERENCES:  None

+----------+
| WEB02010 |
+----------+

   TEXT:        The lines per page value was set to less than
                30.  This report requires a minimum PAGE SIZE
                of 30.  PAGE SIZE set to 30.

   TYPE:        Informational

   REASON:      The PAGESIZE option specified a value for
```

page size that is less than 30. A minimum value of 30 for the page size is required.

ACTION: Change the PAGESIZE= value to 30 or more to avoid this message.

REFERENCES: You can change the page size by going into MWF, selecting option 2;0, and choosing either foreground or batch execution parameters. Find the page length option and specify the number of lines per printed page for inquiry outputs.

Appendix B: DATA DICTIONARY

The CA MICS Web Analyzer's information areas are supported by the Web component of CA MICS. The following sections describe common data elements supported in these areas:

1 - The Common Data Elements in CA MICS Web Analyzer Files
2 - The WebSphere Information Area
3 - The WebSphere Log Files Information Area

Appendix C: WEB LOG EXTRACTOR (WLE) OVERVIEW

The Web Log Extractor (WLE) utility accesses IBM WebSphere server log files in their UNIX System Services Hierarchical File System (HFS) data sets. The utility reformats and writes selected log file records to a standard z/OS or OS/390 dataset for input processing by the CA MICS Web Analyzer.

Please consult with your WebSphere System Programmer for the directory and file paths to complete many configuration parameters and JCL symbolics that are described throughout this appendix.

This section contains the following topics:

C.1 WLE Utility Requirements

The Web Log Extractor (WLE) utility requires the user to have access to UNIX System Services and for WebSphere server logs to be created using the COMMON format.

The COMMON format is specified with the HTTP server configuration directive:

"LogFormat Common"

If a format other than COMMON is specified, the WLE utility abends.

Not all logs are required or necessarily contain data, but at least one must be present.

C.2 WLE Utility Install Job

The WLEINST and WLEINSTM jobs are generated in sharedprefix.MICS.CNTL after the sharedprefix.MICS.CNTL(JCLGEN0) job ends successfully.

The WLEINST job creates three files in a user-defined HFS directory that are required for the WLE utility extract job, prefix.MICS.CNTL(WLERUN): wleload.exe, wleload.conf, and wleload.scp, which are extracted from the WLETAR tar file.

The WLEINSTM job is similar to the WLEINST job with one exception. WLEINSTM extracts only one file, wleload.exe, from the WLETAR tar file, and it is used to update the WLE utility executable code as necessary.

Once the three files are in your HFS directory they are used to create SMF-like data sets from web server log files. Below is a detail checklist that will help you to configure the necessary JCL.

__ 1. Customize Installation JCL.

```
*********************************************************
*                                                       *
*  Note: This program is case sensitive.  JCL must be   *
*  uppercase and UNIX System Services path names must   *
*  be lowercase.                                        *
*                                                       *
*********************************************************
```

The WLE Utility JCL is found in:

sharedprefix.MICS.CNTL(WLEINST).

On the next page, you will find sample JCL for the WLEINST job. The lines are numbered and then referred to later in the checklist for detailed instructions on both how to tailor the individual lines and what they are used for.

```
 1  //WLEINST PROC HFSPATH='HFS.path.to.WLE'
 2  //*
 3  //STEP1   EXEC PGM=BPXCOPY,REGION=8M,
 4  // PARM='ELEMENT(WLETAR) TYPE(BINARY) MODE(0,6,6,6)'
 5  //SYSUT1   DD DSN=&SPREFIX..&SMICS.BIN(WLETAR),DISP=SHR
 6  //SYSUT2   DD PATH='/tmp/',
 7  //          PATHOPTS=(OCREAT,OWRONLY,OAPPEND),
 8  //          PATHMODE=(SIRWXU,SIRWXG,SIRWXO),
 9  //          PATHDISP=(KEEP,KEEP)
10  //SYSTSPRT DD SYSOUT=*
11  //*
12  //STEP2   EXEC PGM=BPXBATCH,REGION=8M,COND=(0,NE,STEP1),
13  // PARM='sh cd ?HFSPATH/;tar -xvof /tmp/WLETAR'
14  //STDOUT DD PATH='/tmp/wleinst.stdout.dat',
15  //          PATHOPTS=(OCREAT,OWRONLY,OAPPEND),
16  //          PATHMODE=(SIRWXU,SIRWXG,SIRWXO),
17  //          PATHDISP=(KEEP,KEEP)
18  //* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
19  //* Print USS messages and command processing log
20  //* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
21  //*
22  //COPYSTDO EXEC PGM=IKJEFT1A,COND=EVEN,
23  // PARM='OCOPY I(SYSUT1) O(SYSUT2) TEXT PATHOPTS(OVERRIDE)'
24  //SYSTSPRT DD SYSOUT=(*)
25  //SYSUT1   DD PATH='/tmp/wleinst.stdout.dat',
26  //            PATHDISP=(DELETE,DELETE)
27  //SYSUT2   DD SYSOUT=*,DCB=(LRECL=255,RECFM=VB,BLKSIZE=2560)
28  //SYSUT3   DD PATH='/tmp/WLETAR',
29  //          PATHDISP=(DELETE,DELETE)
30  //SYSTSIN  DD DUMMY
31  //*
32  //     PEND
33  //STEP EXEC WLEINST
```

Step one (line 3) executes the BPXCOPY utility program to copy WLETAR (Web Log Extract TAR file) from the partitioned data set sharedprefix.MICS.BIN. The WLETAR file is copied to the default temporary directory, /tmp, in Unix Services (USS). The /tmp is the path name of the directory used for temporary files.

```
__ 2.  The HFSPATH symbolic parameter is assigned the value
       of path name that is defined in JCLDEFC with the
       keyword HFSROOT.

       WLEINST job stores the WLE files in a subdirectory of
       the CA MICS Root Directory (&HFSROOT./wle).

       Do not change the HFSPATH symbolic value in the JCL
       since this member is generated. Any changes made here
       will be lost when regenerated.

       You can override the default HFSPATH symbolic value
       by coding the desired value for WLEDIR symbol in
       sharedprefix.MICS.PARMS(JCLNAMES).


       ***************************************************
       * WARNING!  The &HFSROOT./wle must exist and you    *
       * must have read, write, and execute permissions to *
       * this path name. If the HFSPATH name is not found, *
       * the job completes with condition code zero and    *
       * message EDC5129I, "No such file or directory," in *
       * SYSUT2 DD.                                        *
       ***************************************************

       If the directory path is too long to fit on one line,
       the following example shows how to continue an
       apostrophe-enclosed parameter on another line
       (maximum length 255).

              Leave column 72 blank  ---------------------+
                                                          |
              Enter characters up to column 71   --------+|
                                                         ||
    INST PROC MICSHLQ='SHAREDPREFIX.MICS',                vv
         HFSPATH='/u/users/servers/systems/group/test/platf
         orm/http/logs'
         Y            Y
         |            +--- End with closing apostrophe
         |
         +---  Resume with next character column 16
```

__ 3. Submit WLEINST and ensure that it completes successfully with a condition code of zero.

Step two (line 12) ensures that step one completed with a condition code of zero, and the WLETAR archive file was successfully copied to the specified directory. If step one was successful:

- A change directory command is issued to point to the location where the WLE files are stored.
- A tar command is issued to inflate the tar files.
- Using the WLEINSTM job, the tar command extracts wleload.exe only from the WLETAR file.

C.2.1 Executing the WLE Utility

The WLE utility must execute before the normal CA MICS daily update jobs. It extracts IBM WebSphere log data and creates a z/OS or OS/390 dataset for input processing into the DAY052 step for a CA MICS unit containing the CA MICS Web Analyzer.

In order to collect log file data, you must run the Web Log Extractor. The WLE Utility can be run from batch or UNIX System Services. If you choose to use batch, do not switch to UNIX System Services or your batch job will be overwritten. It is recommended that all jobs and parameter tailoring be performed through batch.

The next sections contain information on the following topics:

1 - Executing WLE Utility in Batch
2 - Tailoring WLESCRPT
3 - WLE Utility Return Codes

C.2.1.1 Executing WLE Utility in Batch

__ 1. Edit prefix.MICS.PARMS(WLECONF).

WLECONF is where you configure necessary parameters. This is where you decide what log files you want to collect data from, space allocations, data set names, etc. See Appendix C.3.1.1 and Appendix C.3.1.2 of this guide for all parameter specifications.

__ 2. The following numbered JCL example and correlating steps help to execute the log extract program (WLELOAD.EXE) from the z/OS or OS/390 environment. After the JCL has been updated:

Submit prefix.MICS.CNTL(WLERUN).

Ensure there are no error messages in the SYSUT2 DD of the WLEERROR step, and the job completes with a return code of zero.

Below is an EXEC step with three parameters that must be updated. This procedure and their descriptions follow.

```
 1 //WLERUN  PROC HFSPATH='HFS.path.to.WLE',
 2 //            HFSHELL='wleload.scp',        /* Script File Name  */
 3 //            WLECONF='wleload.conf',       /* WLE Config File   */
 4 //            WLEREPRT='wleload.rpt',       /* WLE Output Report */
 5 //            WLEERROR='wleload.err'        /*  WLE Error Log    */
 6 //*
 7 //*  Copy the script contents
 8 //*
 9 //OCOPYS1  EXEC  PGM=IKJEFT01
10 //INSCP    DD DISP=SHR,DSN=&SPREFIX..&SMICS.PARMS(WLESCRPT)
11 //OUTSCP   DD PATH='&HFSPATH/&HFSHELL',
12 //        PATHDISP=(KEEP,KEEP)
13 //SYSTSPRT DD SYSOUT=*
14 //SYSTSIN  DD DUMMY
15 //*
```

```
16 //*  Copy the WLE config file
17 //*
18 //OCOPYS2  EXEC  PGM=IKJEFT01
19 //INCONF   DD DISP=SHR,DSN=PREFIX.MICS.PARMS(WLECONF)
20 //OUTCONF  DD PATH='&HFSPATH/&WLECONF',
21 //        PATHDISP=(KEEP,KEEP)
22 //SYSTSPRT DD SYSOUT=*
23 //SYSTSIN  DD DUMMY
24 //*

25 //************************************
26 //*   Execute the script file
27 //************************************
28 //*
29 //RUNSCRPT EXEC PGM=BPXBATCH,REGION=8M,
30 // PARM='sh cd &HFSPATH/;&HFSHELL'
31 //*
32 //************************************
33 //* Copy HFS error.file to MVS dataset
34 //************************************
35 //*
36 //OCOPYS3  EXEC  PGM=IKJEFT01
37 //INERROR  DD PATH='&HFSPATH/&WLEERROR',
38 //        PATHOPTS=(ORDONLY)
39 //OUTERROR DD DSN=&&OUTERROR,DISP=(NEW,PASS),UNIT=SYSDA,
40 //            SPACE=(TRK,(5,5),RLSE),
41 //            DCB=(RECFM=VB,LRECL=133,BLKSIZE=137)
42 //SYSTSPRT DD SYSOUT=*
43 //*
44 //* Copy HFS error.file to MVS dataset
45 //*
46 //WLEREPRT  EXEC PGM=IEBGENER
47 //SYSPRINT DD SYSOUT=*
48 //SYSUT1   DD DISP=(OLD,DELETE),DSN=&&OUTERROR
49 //SYSUT2   DD SYSOUT=*
50 //SYSUT3   DD PATH='?HFSPATH/?WLEERROR',
51 //          PATHDISP=(DELETE,DELETE)
52 //SYSIN DD DUMMY
```

```
53 //************************************
54 //* Copy HFS report.file to MVS dataset
55 //************************************
56 //*
57 //OCOPYS4  EXEC  PGM=IKJEFT01
58 //INREPRT  DD PATH='&HFSPATH/&WLEREPRT',
59 //        PATHOPTS=(ORDONLY)
60 //OUTREPRT DD DSN=&&OUTREPRT,DISP=(NEW,PASS),UNIT=SYSDA,
61 //            SPACE=(TRK,(5,5),RLSE),
62 //            DCB=(RECFM=VB,LRECL=133,BLKSIZE=137)
63 //SYSTSPRT DD SYSOUT=*
64 //*
65 //* Copy HFS report.file to MVS dataset
66 //*
67 //WLEREPRT  EXEC PGM=IEBGENER
68 //SYSPRINT DD SYSOUT=*
69 //SYSUT1   DD DISP=(OLD,DELETE),DSN=&&OUTREPRT
70 //SYSUT2   DD SYSOUT=*
71 //SYSUT3   DD PATH='?HFSPATH/?WLEREPRT',
72 //          PATHDISP=(DELETE,DELETE)
73 //SYSIN DD DUMMY
74 //     PEND
75 //STEP     EXEC WLERUN,
76 //*                                                              */
```

```
77  //*************************************************************
78  //OCOPYS1.SYSTSIN  DD *
79  OCOPY INDD(INSCP) OUTDD(OUTSCP) TEXT CONVERT(YES) PATHOPTS(USE)
80  /*
81  //OCOPYS2.SYSTSIN  DD *
82  OCOPY INDD(INCONF) OUTDD(OUTCONF) TEXT CONVERT(YES) PATHOPTS(USE)
83  /*
84  //OCOPYS3.SYSTSIN  DD *
85  OCOPY INDD(INERROR) OUTDD(OUTERROR) TEXT CONVERT(YES) PATHOPTS(USE)
86  /*
87  //OCOPYS4.SYSTSIN  DD *
88  OCOPY INDD(INREPRT) OUTDD(OUTREPRT) TEXT CONVERT(YES) PATHOPTS(USE)
89  /*

       This job executes the BPXBATCH utility program to run
       a shell script that executes the wleload.exe
       program to process HTTP server log files.  This JCL
       may be submitted directly or can be submitted by a
       scheduling product.
__ .   The HFSPATH symbolic parameter is assigned the value
       of path name that is defined in JCLDEFC with the
       keyword HFSROOT.

       WLERUN job copies wleload.conf and wleload.scp files
       in the subdirectory of the CA MICS Root Directory
       (&HFSROOT./wle).  The wleload.scp is a script file
       that executes wleload.exe program.  The wleload.exe
       program redirects the stderr (wleload.err) and stdout
       (wleload.rpt) output to this directory.

       Do not change the HFSPATH symbolic value in the JCL
       since this member is generated. Any changes made here
       are lost when regenerated.

       You can override the default HFSPATH symbolic value
       by coding the desired value for WLEDIR symbol in
       prefix.MICS.PARMS(JCLNAMES).

       Note:  The &HFSROOT./wle must exist, contain the WLE
       utility executable code, and you must have read,
       write, and execute permissions to this path name or
       the job will fail.
```

```
        The following parameters are optional and can be
        added to the EXEC statement, located at the bottom of
        this JCL job, to override the default file names.

__ 2a.  Change "HFSHELL" to point to the script file that
        must be in the directory, as defined by step 1
        (above).  The default file name is wleload.scp.  If
        you would like to specify a specific date for the
        HFSHELL, you can tailor this member to meet your
        site's needs in:


        prefix.MICS.PARMS(WLESCRPT)

        See Appendix C.2.1.2 in this guide for further
        information about date manipulation and redirection,
        pertaining to the WLESCRPT member.

__ 2b.  Change "WLECONF" to point to the WLE configuration
        file.  This is where you define what log files you
        want to collect data from, the OS data set name,
        space allocations, etc.  See Section C.3.1.1 of this
        guide for all parameter specifications.

        Updates can be made to the configuration in
        prefix.MICS.PARMS(WLECONF).

__ 2c.  Change "WLEREPRT" to match the name of the report
        file set in the wleload.scp file, using redirection.
        To retrieve the report from UNIX System Services when
        running the wleload program from a z/OS or OS/390
        address space, the wleload.scp file must use
        redirection for stdout to create an HFS file that can
        be sent to JES output.

__ 2d.  Change "WLEERROR."  This is where all the non-utility
        related errors are.
```

C.2.1.2 Tailoring WLESCRPT

The WLESCRPT member provides you with the ability to specify date parameters and whether or not output is written to an HFS file. If you would like to specify a specific date for the HFSHELL, you can tailor this member to meet your site's needs in:

prefix.MICS.PARMS(WLESCRPT)

WLESCRPT requires one parameter, the path and name of the configuration file that was created. An example follows:

wleload.exe ./wleload.conf

This command executes the wleload program and directs it to look for the configuration file in the same directory as the wleload program is executing from, with the file name wleload.conf. The user can name the configuration file anything, as long as it conforms to UNIX System Services HFS file naming conventions. In addition, the configuration file does not have to be in the same directory as the executing program. For example, if the user created a configuration file called my.config and placed it in a different directory called /u/users/command, the following command would be used:

wleload.exe /u/users/command/my.config

The HTTPD server creates log files with a suffix in the format of Mmmddyyyy and "rolls" the log files at midnight. For example, on June 6th, 2001 from midnight until 11:59:59 log records are written to selected log files with a suffix of Jun062001. On June 7th, 2001 at midnight, the HTTPD server would stop writing records to the Jun062001 suffixed file. It would open and start writing to a new file with a suffix of Jun072001. Therefore, the wleload extract program should be run shortly after midnight each day. If wleload is executed without a date parameter, it takes the current date, subtracts one day from it, and tries to process log files with a file suffix from the previous day. For normal operations, wleload would be run without any date parameters. If there is a need to process a previous day's log file records, the -d date parameter can be used to specify which log files you want to process. For example, it is June 12th, 2001 and you want to process data for log files with a suffix of Jun052001, to do this follow the example below:

wleload.exe -d Jun052001 ./wleload.conf

The above example would cause wleload to execute and use the wleload.conf configuration file found in the same directory that the wleload program is running from. It would also use a file suffix for the log files to process as Jun052001, instead of Jun112001 (yesterday's date).

Finally, the execution time messages, diagnostic information, and the configuration report are written to stdout. This defaults to the display device. By using UNIX redirection, you can cause the output to be written to an HFS file that can be viewed later. Listed below is an example command to use in order to write output to an HFS file:

```
wleload.exe ./wleload.conf >wleload.rpt
```

When wleload.exe is executed with the redirection symbol for stdout with an >, the output is directed to the HFS file specified following the redirection symbol.

C.2.1.3 WLE Utility Return Codes

The prefix.MICS.CNTL(WLERUN) job designed to run in batch can complete with many different return codes. The description of these return codes is given below:

Return Code	Description
00	Normal Completion
08	Error messages issued for invalid data type. This can be caused by non-numeric values found where numeric values are expected, and so on. OR An empty log file found.
12	Open failed for log file

16 Errors occurred that are caused by discrepancies in script file, wleload.scp, config file, or wleload.conf. The only other time this message is issued is when the number of errors counted during WLE execution exceeds the error limit set by the "ErrorLimit" keyword in prefix.MICS.PARMS(WLECONF).

For a detail description of the WLEnnn error message, see Section C.4 in the CA MICS Web Analyzer Option Guide. Below is a list of some of the common errors:

- Invalid command line arg in wleload.scp
- Open failed for input configuration file pointed to in wleload.scp
- Invalid keyword parameter encountered
- Fopen failed for WLEDATA, MVS data set referred to by this ddname
- Dynalloc failed, error code = 970C, info code = 0, S99ERSN = 42CC.

127 Catastrophic errors

- wleload.exe file name not found in the directory.

C.3 Defining the Configuration Parameters

Configurations pertaining to the Web Log Extractor set up what and where output is written. This chapter describes how to define the parameters that are required for configuring WLE. The CA MICS System Administrator should use this chapter as a detailed reference in conjunction with the PIOM.

Defining Web Log Extractor parameters requires you to gain the necessary understanding of your site and its needs, and translate that understanding into meaningful parameters.

This chapter focuses on considerations that are unique to the Web Log Extractor.

If you have a question at any time during your review of the material presented here, contact CA Technical Support.

C.3.1 Web Log Extractor Configuration File

The Web Log Extractor configuration file contains configuration directives that are used by the wleload.exe program to create OS/390 SMF like records. Various segments exist in which the Global and Server parameters must be specified before processing begins. The following subsections describe the various parameters and other pertinent information that relates to them.

1 - Configuration File Global Information
2 - Configuration File Server Information

C.3.1.1 Configuration File Global Information

When preparing to configure the Web Log Extractor, certain parameters must be defined. This section explains the global parameters and their function.

Note: The following parameters keywords are case-sensitive, and errors can occur if they are not entered correctly. Every attempt is made by the utility program to correct case-sensitive errors before the program aborts.

Note: Only the DataSetName parameter is required. All other parameters are optional, but must be specified if the parameter defaults do not meet your site requirements.

An example of a WLE Configuration File Global Information parameter specification is shown below. A detailed explanation of each parameter follows.

```
DataSetName     SERVER2.OS390.HTTP.LOGRECS
DSmodelDSCB     no
RecType         241
DSprimary       5
DSsecondary     2
DSunit          3390
DSvolser        MICS01
```

```
DSstorClas     no
DSdataClas     no
DSmgmtClas     no
DSdisp         OLD
CompanyName    CA
ConfigReport   yes
ErrorLimit     100
```

o DataSetName - Data Set Name

Required: Yes

Default : None

The data set name, created or written to by the WLE Extract Utility job, must be no more than 44 characters and follow standard MVS data set naming conventions. This is the data set input by the DAY052 step in the CA MICS daily update job. It is recommended that this data set be created as a GDG data set by adding a (+1) to the end of data set name in order to minimize changing of the DISP parameter. If a GDG type data set is indicated, the DSmodelDSCB parameter may be required.

o DSmodelDSCB - Model DSCB name

Required: No

Default : no

The model DSCB name may be required when creating GDG output data sets with the WLE Extract Utility job. It must be no more than 44 characters, and follow standard MVS data set naming conventions. This parameter should reference the model DSCB used for GDG data sets on the system that the WLE utility is being executed on.

- RecType - Record Type / Number

  Required: No

  Default: 241

  The records created from the server log files by the WLE Extract Utility job are "SMF like", and contain the record type or number, in byte 2 of each record. You can specify a numeric value other than 241, as long as it falls between 128 and 241.

- DSprimary - Data Set Primary Space Allocation

  Required: No

  Default: 9

  Primary space allocation, in cylinders, for the output data set name. The default value is 9 cylinders.

- DSsecondary - Data Set Secondary Space Allocation

  Required: No

  Default: 5

  Secondary space allocation, in cylinders, for the output data set name. The default value is 5 cylinders.

- DSunit - Data Set Unit Name

  Required: No

  Default: SYSALLDA

  Unit for the output data set name (e.g., SYSALLDA, SYSDA, 3390, etc.).

- DSvolser - Data Set Volume Serial Number

  Required: No

  Default: None

  Volume Serial number for the output data set name. No default is available.

- DSstorClas - Storage Class

  Required: No

  Default: no

  SMS Storage class name for output data set name. The default value is "no".

- DSdataClas - Data Class

  Required: No

  Default: no

  SMS Data class name for output data set name. The default value is "no".

- DSmgmtClas - Management Class

  Required: No

  Default: no

  SMS Management class name for output data set name. The default value is "no".

- DSdisp - Data Set Disposition

  Required: No

  Default: NEW

  The disposition of the output data set name at open time. Possible values for this parameter are either NEW or OLD.

- CompanyName - Company Name

  Required: No

  Default: CA

  Company name used for web log extract report headings. This field can be up to 44 characters.

- ConfigReport - Configuration Report

  Required: No

  Default: yes

  The ConfigReport parameter allows you to suppress generation of the GLOBAL and SERVER extract reports by the WLE Extract Utility job.

  Possible values for this parameter are yes or no. If ConfigReport no is specified, processing statistics for the selected log files and error messages are produced. If ConfigReport yes is specified, in addition to processing statistics and error messages (generated by ConfigReport no), a formatted report of the configuration file is printed.

- ErrorLimit - Error Limit

  Required: No

  Default: 100

  Number of errors allowed per log file before termination. The WLE Utility job expects the various server log files to be formatted according to industry defined specifications. Occasionally, fields in the log records are not formatted correctly, and must be skipped. This parameter allows for a reasonable amount of malformatted data without rejecting the processing of an entire log file.

C.3.1.2 Configuration File Server Information

Before running the Web Log Extractor certain parameters must be defined. This section describes the server parameters that must be defined and those that are optional.

Note: The following parameters are case sensitive and errors may occur if they are not entered correctly. Every attempt is made by the utility program to correct CASE sensitive errors before the program aborts.

Note: A site may have more than one server. Certain parameters must be specified for each server targeted by the Web Log Extractor.

An example of a Configuration File Server Information parameter specification is shown below. A detailed explanation of each parameter follows.

```
ServerName ABCD
ServerRoot /u/users/webserver/pathname
HostName user_defined_name
PortNo 1234
AccessLog /u/users/mics/humresc/logs/httpd-log
FRCAaccessLog /u/users/mics/humresc/logs/httpd-logfrca
ProxyLog /u/users/mics/humresc/logs/httpd-proxy
FRCAproxyLog /u/users/mics/humresc/logs/httpd-frcaproxy
AgentLog /u/users/mics/humresc/logs/Agent
RefererLog /u/users/mics/humresc/logs/referer-log
ErrorLog /u/users/mics/humresc/logs/httpd-errors
```

o ServerName: Server Name

Required: Yes

Default : None

The name you want to assign to this Server's log file records. This name must be coded in upper case since this is eventually assigned to part of the COMPT field used in checkpoint file during DAY051 processing.

The length can be a maximum of four characters. There is no default name. The ServerName 4-byte value defined here is assigned to WEBSNAME (Web Server Short Name) if one is not assigned in prefix.MICS.PARMS(WEBSLIST).

```
*****************************************************
* WARNING ! WARNING ! WARNING ! WARNING ! WARNING !  *
*                                                    *
* NOTE: The following words must not be used for     *
* ServerName.  They are reserved for CA MICS use     *
* only:                                              *
*                                                    *
*    WAS  WebSphere Application Server               *
*    HTTP HTTP Web Server                            *
*    WLG  HTTP Web Server Log Data Information Area  *
*    WEB  SMF Data Information Area                  *
*    WLE  HTTP Web Server Log Extractor Utility      *
*    SMF  System Management Facility                 *
*                                                    *
* WARNING ! WARNING ! WARNING ! WARNING ! WARNING !  *
*****************************************************
```

o Orgsysid: Original System Identification

Required: Yes. Must contain upper case letters only.

Default : None

The ORGSYSID value corresponds to a default system where the Web Server executes. The actual system where the server executed is not provided in the measurement log data, thus the specified value you provide is required. CA MICS employs the ORGSYSID value to set SYSID (Logical System) a common key sequence element. The value can be a maximum of four characters. There is no default name. The WLE utility will not execute successfully if Orgsysid is left blank. All Orgsysids in the input stream are required in prefix.MICS.PARMS(SYSID).

Additionally, if the system where the WLE utility executes is different from any of the systems whose Web Servers are coded in the prefix.MICS.PARMS(WLECONF) file, then an entry for this system in prefix.MICS.PARMS(SYSID) is required as well.

IMPORTANT CONSIDERATIONS

Since the log records do not identify the executing system, your specification represents a "static" map of the "expected" or "normal" relationship. In some sites, certain servers have a definitive and accurate relationship (server x always runs on TSO2). In other situations the relationship cannot be accurate as the server executes on different systems over time.

A unique value for ORGSYSID can be specified for each server. This will result in each server having a unique SYSID value.

```
You can, however, specify the same ORGSYSID value for
two or more servers.  This allows the log data for
multiple servers to represent a "business entity"
without attempting to define a relationship to an
execution system where execution location is dynamic.

Because SYSID is present in virtually all CA MICS
files, and analysts typically merge files from
multiple CA MICS components to generate reports and
graphs, it is important to avoid misrepresentation of
metrics in cases where the executing SYSID is not
explicitly known.

One method to avoid misrepresentation of metrics is to
map servers to a single ORGSYSID such as "ANYW".  ANYW
then can be mapped to a unique SYSID such as a PWEB
(an imaginary system).


*****************************************************
* WARNING ! WARNING ! WARNING ! WARNING ! WARNING !  *
*                                                   *
* The Orgsysid value is assigned to ORGSYSID, and    *
* then used to lookup logical SYSID.   You must      *
* define each Orgsysid value (including the Orgsysid *
* of the system where the WLE utility executes) in   *
* prefix.MICS.PARMS(SYSID) member.                   *
*                                                   *
* Your DAILY job step DAY052 will abend with U310    *
* error code if the any of the Orgsysid mentioned    *
* above in not defined in prefix.MICS.PARMS(SYSID).  *
*                                                   *
* WARNING ! WARNING ! WARNING ! WARNING ! WARNING !  *
*****************************************************
```

o ServerRoot: Server Root Directory Name

Required: No

Default : No

The name you want to assign to this Server's Root Directory. This can be found in web server's HTTPD.CONF file. If this field is left blank, the CA MICS element WEBSRVRT in files (WLGSAL, WLGAPS, and WLGCFG) also reflects blank values. This information is helpful to determine the path for resources accessed on the web server. The length can be up to 48 characters long. There is no default name.

o HostName - Host Name

Required: Yes

Default : None

A user-defined name assigned to this web server. You should use a meaningful name that can easily identify the web server in reports and CA MICS Web Analyzer WEB Log Files Information Area (WLG) files. The length can be a maximum of 20 characters. There is no default name.

o PortNo - Port Number

Required: Yes

Default : no

The Port Number this server listens on for requests. There is no default value.

NOTE: To turn on the log file processing, enter the path and first node of the filename for the server's log file. The WLELOAD program appends the appropriate file name suffix. This applies for all of the log files listed below.

- AccessLog - Access Log

  Required: Yes

  Default : no

  This parameter specifies whether or not you want to turn on the Access Log file. The default value is NO.

- FRCAaccessLog - FRCA Access Log

  Required: Yes

  Default : no

  This parameter specifies whether or not you want to turn on the FRCA Access Log file. The default value is NO.

- ProxyLog - Proxy Log

  Required: Yes

  Default : no

  This parameter specifies whether or not you want to turn on the Proxy Log File. The default value is NO.

- FRCAproxyLog - FRCA Proxy Log

  Required: Yes

  Default : no

  This parameter specifies whether or not you want to turn on the FRCA Proxy Log File. The default value is NO.

- AgentLog - Agent Log

  Required: Yes

  Default : no

  This parameter specifies whether or not you want to turn on the Agent Log File. The default value is NO.

- RefererLog - Referer Log

  Required: Yes

  Default : no

  This parameter specifies whether or not you want to turn on the Referer Log File. The default value is NO.

- ErrorLog - Error Log

  Required: Yes

  Default : no

  This parameter specifies whether or not you want to turn on the Error Log File. The default value is NO.

C.4 WLE Load Utility Messages

This appendix lists all messages generated by the WLE Server. Some messages are generated during the processing of the control statements, while others are caused by various conditions in the data found during processing. The messages are listed in ascending numerical sequence and include the full text of the message, the type, the reason for the message, appropriate user action, and applicable references to documentation.

The following type codes are used to categorize the messages:

Information	Designates a note that documents a web server option or potentially important feature in the data.
Warning	Designates a condition in either the data or the control statements that does not affect the web server's operation, but that may lead to unexpected results.
Error	Designates that a problem has been encountered with a control statement that will prevent a successful run of the Server. Execution is stopped after all control statements are processed.

The message text often includes references to information that is contained in a control statement or values of permanent or temporary variables created during web server processing. In the descriptions below, a message text may contain a word beginning with a percent sign (%), which indicates that a value will be substituted into the text at execution time. %S signifies that a character string will be substituted and %D signifies that a decimal string will be substituted. For example, message WLE0054E reads as follows:

INVALID DATE IN LOG RECORD, FOUND %S.

If an invalid record was detected in a specific log record where %S='Jun072000' then the message is printed as follows:

INVALID DATE IN LOG RECORD, FOUND JUN072000

This sections lists all messages generated by the CA MICS WLE Load utility.

```
+----------+
| WLE0001E |
+----------+
```

TEXT: INVALID COMMAND LINE ARGS, USAGE: PGNAME.EXE -d date FILENAME

TYPE: Error

REASON: The command line arguments were incorrect. -d date is optional. Filename is required and defines the path/filename of the configuration file.

ACTION: Re-execute the program with the proper command line arguments.

REFERENCES: Sections C.2.1.1 and C.2.1.2 in this guide.

```
+----------+
| WLE0002E |
+----------+
```

TEXT: OPEN FAILED FOR INPUT CONFIGURATION FILE "FILENAME"

TYPE: Error

REASON: The open failed for the configuration file specified as the first command line argument.

ACTION: Check the path and filename specified on the command line argument and ensure the file exists.

REFERENCES: Sections C.2.1.1 and C.2.1.2 in this guide.

```
+----------+
| WLE0003E |
+----------+

   TEXT:        EXECUTION TERMINATED DUE TO PREVIOUS ERROR

   TYPE:        Error

   REASON:      A severe error occurred and execution cannot
                continue.

   ACTION:      Refer to previously printed messages for
                problem determination.

   REFERENCES:  Refer to previously printed messages for
                problem determination.

+----------+
| WLE0004E |
+----------+

   TEXT:        ERROR READING INPUT CONFIGURATION FILE
                "FILE"

   TYPE:        Error

   REASON:      An error occurred while reading the input
                configuration file.

   ACTION:      Check that the input configuration file
                contains data.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0005E |
+----------+

   TEXT:        MALLOC FAILED FOR CONFIGURATION

   TYPE:        Error

   REASON:      An error occurred while attempting to
                allocate storage to contain configuration
                file data.

   ACTION:      Determine why storage is low in USS address
                space and reexecute program.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0006E |
+----------+

   TEXT:        MAXIMUM LENGTH OF KEYWORD TOKEN IS 13,
                FOUND "TOKEN"

   TYPE:        Error

   REASON:      A keyword was found that was longer than the
                maximum allowable keyword length.

   ACTION:      Update the configuration file keyword
                parameter in error and reexecute program.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0007E |
+----------+

   TEXT:        A VALUE MUST BE SPECIFIED WITH A KEYWORD

   TYPE:        Error

   REASON:      A valid keyword was entered with no
                corresponding value.

   ACTION:      Update the configuration file keyword
                parameter to contain a value.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0008E |
+----------+

   TEXT:        THE MAXIMUM LENGTH OF A KEYWORD VALUE IS 128,
                FOUND "VALUE" THAT EXCEEDED THE MAXIMUM
                ALLOWABLE VALUE LENGTH.

   TYPE:        Error

   REASON:      A valid keyword was entered with a value that
                exceeded the maximum allowable value length.

   ACTION:      Update the configuration file keyword
                parameter to contain a value of less than 128
                characters.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0009E |
+----------+

   TEXT:        INVALID KEYWORD ENCOUNTERED, "KEYWORD"

   TYPE:        Error

   REASON:      An unknown keyword was encountered.

   ACTION:      Update the configuration file keyword in
                error to an acceptable value.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0010E |
+----------+

   TEXT:        INVALID KEYWORD PARAMETER ENCOUNTERED,
                "VALUE"

   TYPE:        Error

   REASON:      A valid keyword was encountered with an
                invalid value.

   ACTION:      Update the configuration file keyword
                parameter in error to an acceptable value.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0011E |
+----------+

   TEXT:        PARAMETER FOR THIS KEYWORD MUST BE YES OR NO

   TYPE:        Error

   REASON:      A valid keyword was encountered with a value
                other than YES or NO.

   ACTION:      Update the configuration file keyword
                parameter to a value of YES or NO.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0012E |
+----------+

   TEXT:        RECTYPE VALUE MUST BE BETWEEN 128 AND 255

   TYPE:        Error

   REASON:      A valid keyword was encountered with an out-
                of-bounds value.

   ACTION:      Update the configuration file keyword
                parameter to a value between 128 and 255.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0013E |
+----------+

   TEXT:        THE %s NAME EXCEEDS MAXIMUM OF 44 CHARACTERS

   TYPE:        Error

   REASON:      Either the DataSetName keyword or the
                DSmodelDSCB keyword was specified with a name
                containing more than 44 characters.

   ACTION:      Update the configuration file keyword
                parameter to a value of less than 44
                characters.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0014E |
+----------+

   TEXT:        THE FIRST CHARACTER OF EACH %s NODE MUST BE
                ALPHABETIC

   TYPE:        Error

   REASON:      The first character of DataSetName node or
                DSmodelDSCB name node was not an alphabetic
                character.

   ACTION:      Update the configuration file keyword
                parameter to a valid OS/390 data set name or
                model DSCB.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0015E |
+----------+

   TEXT:        A NODE EXCEEDS THE MAXIMUM LENGTH OF 8

   TYPE:        Error

   REASON:      The DataSetName keyword value was found to
                contain a node with more than 8 characters.

   ACTION:      Update the configuration file keyword
                parameter to a valid OS/390 data set name.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0016E |
+----------+

   TEXT:        TWO CONSECUTIVE NODE SEPARATORS WERE FOUND,
                INVALID

   TYPE:        Error

   REASON:      The DataSetName keyword value or the
                DSmodelDSCB keyword value was found to
                contain two consecutive node separators.

   ACTION:      Update the configuration file keyword
                parameter to a valid OS/390 data set name.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0017E |
+----------+

   TEXT:        A NUMERIC VALUE WAS EXPECTED, FOUND
                NON-NUMERIC DATA

   TYPE:        Error

   REASON:      A value that should contain numeric data was
                found to have non-numeric data.

   ACTION:      If subsequent messages indicate keyword
                processing, correct the keyword parameter in
                the WLE configuration file. If processing log
                file data, try to determine the cause of
                invalid data that is logged in the log file.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0018E |
+----------+

   TEXT:        PARAMETER FOR THIS KEYWORD MUST BE SPECIFIED
                AS NEW OR OLD

   TYPE:        Error

   REASON:      A keyword was encountered that requires a
                value of NEW or OLD, and the value was
                neither.

   ACTION:      Update the configuration file keyword
                parameter to a value of NEW or OLD.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0019E |
+----------+

   TEXT:        COMPANY NAME MAY NOT EXCEED 40 CHARACTERS

   TYPE:        Error

   REASON:      The company keyword had a value that exceeded
                the maximum allowable length.

   ACTION:      Update the configuration file keyword
                parameter to contain value with 40 characters
                or less.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0020E |
+----------+

   TEXT:        LENGTH OF KEYWORD VALUE EXCEEDS MAXIMUM

   TYPE:        Error

   REASON:      A keyword was encountered with a value that
                exceeds the maximum length for that keyword.

   ACTION:      Update the configuration file keyword
                parameter to contain a value with an
                appropriate length.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0022E |
+----------+

   TEXT:        THIS KEYWORD PARAMETER REQUIRES UPPER CASE

   TYPE:        Error

   REASON:      A keyword requiring an uppercase value was
                encountered that contained lowercase data.

   ACTION:      Update the configuration file keyword
                parameter to contain a valid OS/390 unit
                specification.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0023E |
+----------+

   TEXT:        VOLSER PARAMETER MUST BE 6 CHARACTERS IN
                LENGTH

   TYPE:        Error

   REASON:      The VOLSER keyword specified a parameter that
                was greater than 6 characters in length.

   ACTION:      Update the configuration file keyword
                parameter to contain a valid OS/390 VOLSER
                specification.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0024E |
+----------+

   TEXT:        THE FIRST CHARACTER OF VOLSER MUST BE
                ALPHABETIC

   TYPE:        Error

   REASON:      The VOLSER keyword specified a parameter that
                began with a non-alphabetic character.

   ACTION:      Update the configuration file keyword
                parameter to contain a valid OS/390 VOLSER
                specification.

   REFERENCES:  Sections C.2.1.1 or C.2.1.2 in this
                guide.

+----------+
| WLE0025E |
+----------+

   TEXT:        THE FIRST CHARACTER OF %s MUST BE ALPHABETIC

   TYPE:        Error

   REASON:      The DSstorclas, DSdataClas, or DSmgmtClas
                keyword specified a parameter that began with
                a non-alphabetic character.

   ACTION:      Update the configuration file keyword
                parameter to contain a valid OS/390 Storclas,
                DataClas, or MgmtClas specification.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0026E |
+----------+

   TEXT:        DATASETNAME MUST BE SPECIFIED

   TYPE:        Error

   REASON:      The DataSetName keyword was not found.

   ACTION:      The DataSetName keyword must be specified
                with a valid OS/390 data set name.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0027E |
+----------+

   TEXT:        FIRST CHARACTER OF SERVERNAME MUST BE
                ALPHABETIC

   TYPE:        Error

   REASON:      The ServerName keyword was specified with a
                parameter that began with a non-alphabetic
                character.

   ACTION:      Update the configuration file keyword
                parameter to contain a servername that begins
                with an alphabetic character.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0028E |
+----------+

   TEXT:        THIS KEYWORD PARAMETER MUST BE ALPHA-NUMERIC

   TYPE:        Error

   REASON:      A keyword parameter that is required to be
                alpha-numeric is not.

   ACTION:      Update the configuration file to contain a
                value with alpha-numeric characters only.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0029E |
+----------+

   TEXT:        SERVERNAME KEYWORD MUST PRECEDE THIS KEYWORD

   TYPE:        Error

   REASON:      A keyword that must be associated with a
                servername has been encountered, but no
                servername keyword has been encountered.

   ACTION:      Update the configuration file to contain the
                ServerName keyword before this keyword.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0030E |
+----------+

   TEXT:        DUPLICATE KEYWORD WITHIN SEGMENT FOUND

   TYPE:        Error

   REASON:      A duplicate keyword within a Servername
                segment has been encountered and is invalid.

   ACTION:      Update the configuration file so that no two
                keywords in a ServerName segment are the
                same.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0031E |
+----------+

   TEXT:        KEYWORDS HOSTNAME, PORTNO, AND ORGSYSID ARE
                REQUIRED AND NOT FOUND

   TYPE:        Error

   REASON:      The keywords HostName, PortNo, and Orgsysid
                are required entries and none were found.

   ACTION:      Update the configuration file so that each
                ServerName segment contains a HostName,
                PortNo, and Orgsysid specification.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0032E |
+----------+

   TEXT:        AT LEAST ONE LOG FILE MUST BE SELECTED FOR
                PROCESSING.

   TYPE:        Error

   REASON:      A ServerName segment was encountered that had
                no log files specified for processing.

   ACTION:      Update the configuration file so that each
                ServerName segment contains at least one log
                file to process.  If no log file processing
                is required for this ServerName, remove it
                from the configuration file.

   REFERENCES:  Sections C.2.1.1 or C.2.1.2 in this
                guide.

+----------+
| WLE0033E |
+----------+

   TEXT:        AT LEAST ONE SERVERNAME PARAMETER WITH
                SEGMENT PARAMETERS MUST BE SPECIFIED.

   TYPE:        Error

   REASON:      No ServerName keywords and associated
                parameters were found in the configuration
                file.

   ACTION:      Update the configuration file adding a
                Servername keyword and associated segment
                parameters.  If no log file processing is
                required for this ServerName, remove it from
                the configuration file.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0034E |
+----------+

   TEXT:        CLOSE ERROR OCCURRED ON CONFIGURATION FILE.

   TYPE:        Error

   REASON:      An error occurred during close processing for
                the configuration file.

   ACTION:      Reexecute the program.  If the problem
                persists, contact CA Technical Support.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0036E |
+----------+

   TEXT:        MALLOC FAILED FOR RECORD BUFFER.

   TYPE:        Error

   REASON:      An error occurred while trying to obtain
                storage to hold record segment data.

   ACTION:      Determine why storage is low in UNIX System
                Services and reexecute the program.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0037E |
+----------+

   TEXT:        FOPEN FAILED FOR WLEDATA.

   TYPE:        Error

   REASON:      An error occurred while opening the OS/390
                ddname WLEDATA.

   ACTION:      If DSdisp in the configuration file was
                specified as OLD, ensure that the data set
                exists.  If it does not, specify DSdisp as
                new and reexecute program.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0038E |
+----------+

   TEXT:        FCLOSE FAILED FOR WLEDATA.

   TYPE:        Error

   REASON:      An error occurred while closing the OS/390
                ddname WLEDATA.

   ACTION:      Reexecute the program.  If the problem
                persists, contact CA Technical Support.

   REFERENCES:  None
```

```
+----------+
| WLE0039E |
+----------+

   TEXT:        DYNALLOC FAILED, ERROR CODE = XXXX,
                INFO CODE = XXXX, S99ERSN = XXXX.

   TYPE:        Error

   REASON:      An error occurred while attempting to
                DYNALLOC the WLEDATA ddname.

   ACTION:      Look up the returned values for ERROR CODE,
                INFO CODE, and S99ERSN to assist in problem
                determination.  Check for conflicting
                parameters in the configuration file.
                The meaning of the returned values for
                ERROR CODE, INFO CODE and S99ERSN can be
                found in the IBM Manual - MVS Programming:
                Authorized Assembler Services Guide.


   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0040I |
+----------+

   TEXT:        WEB LOG EXTRACT CONFIGURATION PROCESSING
                BEGINS.

   TYPE:        Informational

   REASON:      Indicates that configuration data is about to
                be processed.

   ACTION:      None

   REFERENCES:  None
```

```
+----------+
| WLE0041I |
+----------+

   TEXT:        %D CONFIGURATION SEGMENTS CREATED IN %D
                RECORDS.

   TYPE:        Informational

   REASON:      Indicates the number of configuration
                segments processed and the number of records
                written.

   ACTION:      None

   REFERENCES:  None

+----------+
| WLE0042I |
+----------+

   TEXT:        WEB LOG EXTRACT CONFIGURATION PROCESSING
                COMPLETE.

   TYPE:        Informational

   REASON:      Indicates that configuration segment
                processing completed successfully.

   ACTION:      None

   REFERENCES:  None

+----------+
| WLE0043E |
+----------+

   TEXT:        ERROR WRITING TO WLEDATA, ERRNO = %D.

   TYPE:        Error

   REASON:      An error occurred trying to write to DDname
                WLEDATA.

   ACTION:      Use errno value to determine the cause of the
                failure.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this guide.
```

```
+----------+
| WLE0044I |
+----------+

   TEXT:        %D RECORDS WRITTEN FOR THIS EXECUTION.

   TYPE:        Informational

   REASON:      Indicates the total number of records
                written.

   ACTION:      None

   REFERENCES:  None

+----------+
| WLE0045E |
+----------+

   TEXT:        INVALID DATE ENTERED AS COMMAND LINE ARG,
                FOUND %S.

   TYPE:        Error

   REASON:      The user entered a date as a command line
                argument to process log files other than the
                defaults, and the date was invalid.

   ACTION:      Reexecute WLELOAD supplying a valid date in
                the format MMDDYYYY.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0046E |
+----------+

   TEXT:        ERROR PROCESSING LOG FILE %S.

   TYPE:        Error

   REASON:      An error occurred while processing this log
                file.

   ACTION:      See other correlating error messages for more
                detailed information about the error.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0047W |
+----------+

   TEXT:        OPEN FAILED FOR LOG FILE.

   TYPE:        Error

   REASON:      An error occurred while opening the log file
                indicated in message WLE0051E.

   ACTION:      Determine if the path name specified in the
                configuration file is correct.  If the path
                name is specified correctly, it may indicate
                that no records were produced for this log
                file in the preceding day.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0048E |
+----------+

   TEXT:        ERROR READING LOG FILE.

   TYPE:        Error

   REASON:      An error occurred while reading the log file
                indicated in message WLE0046E.

   ACTION:      Determine that the log file exists and is
                accessible to the program and rerun job.  If
                the problem persists, contact CA Technical
                Support for assistance.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0049I |
+----------+

   TEXT:        WEB LOG EXTRACT LOG FILE PROCESSING BEGINS.

   TYPE:        Informational

   REASON:      Informational message indicating that log
                file processing is starting.

   ACTION:      None

   REFERENCES:  None
```

```
+----------+
| WLE0050E |
+----------+

   TEXT:        MAXIMUM LENGTH OF DOMAIN NAME / TCP/IP
                ADDRESS OR HOST NAME EXCEEDED.

   TYPE:        Error

   REASON:      Client IP address of hostname is longer than
                expected.

   ACTION:      Review log file for possible corruption.  If
                log file appears correct and is in the common
                log format, contact CA Technical Support.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0051I |
+----------+

   TEXT:        PROCESSING LOG FILE %S FOR SERVER %S.

   TYPE:        Informational

   REASON:      Informational message indicating the log file
                being processed and the server that it
                belongs to.

   ACTION:      None

   REFERENCES:  None
```

```
+----------+
| WLE0052E |
+----------+

   TEXT:        INVALID RECORD IN LOG FILE, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0053E |
+----------+

   TEXT:        MAXIMUM LENGTH OF AUTHID WAS EXCEEDED, FOUND
                %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0054E |
+----------+

   TEXT:        INVALID DATE IN LOG RECORD, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0055E |
+----------+

   TEXT:        MISSING TOKEN IN LOG RECORD, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0056E |
+----------+

   TEXT:        INVALID TIME IN LOG RECORD, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0057E |
+----------+

   TEXT:        INVALID REQUEST IN LOG RECORD, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0058E |
+----------+

   TEXT:        INVALID URI IN LOG RECORD, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0059E |
+----------+

   TEXT:        INVALID SERVER IN LOG RECORD, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0060E |
+----------+

   TEXT:        INVALID RELEASE IN LOG RECORD, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0061E |
+----------+

   TEXT:        INVALID STATUS IN LOG RECORD, FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0062E |
+----------+

   TEXT:        INVALID BYTES TRANSFERRED IN LOG RECORD,
                FOUND %S.

   TYPE:        Error

   REASON:      An invalid record was detected in the log
                being processed.

   ACTION:      Check the log records and make sure they are
                in the common format.  Possible missing
                delimiter.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.


+----------+
| WLE0063I |
+----------+

   TEXT:        %D SUBTYPE %D %S SEGMENTS CREATED IN %D
                RECORDS.

   TYPE:        Informational

   REASON:      Indicates how many log records were
                processed, and how many subtype records were
                written for which type of log being
                processed.

   ACTION:      None

   REFERENCES:  None
```

```
+----------+
| WLE0064W |
+----------+

   TEXT:        EMPTY LOG FILE ENCOUNTERED, PROCESSING
                CONTINUES.

   TYPE:        Warning

   REASON:      An empty log file was encountered.

   ACTION:      Determine if data was missed, and rerun job
                if required.

   REFERENCES:  None

+----------+
| WLE0065E |
+----------+

   TEXT:        INVALID AGENT/REFERER DATA IN LOG RECORD,
                FOUND %S.

   TYPE:        Error

   REASON:      An unexpected data format was found in the
                agent referer log records.

   ACTION:      Make sure your server is recording log
                information in the common log format and that
                you are processing the correct file.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.
```

```
+----------+
| WLE0066E |
+----------+

   TEXT:        CLOSE ERROR ON LOG FILE %S.

   TYPE:        Error

   REASON:      An error occurred while closing log file %s.

   ACTION:      Execute the web log extract again, if
                problems persist, contact CA Technical
                Support.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0067I |
+----------+

   TEXT:        WEB LOG EXTRACT EXECUTION COMPLETE.

   TYPE:        Informational

   REASON:      Web log extract processing has completed
                successfully.

   ACTION:      None

   REFERENCES:  None
```

```
+----------+
| WLE0068E |
+----------+

   TEXT:        ERROR LIMIT FOR LOG FILE EXCEEDED.

   TYPE:        Error

   REASON:      Web log extract processing terminates because
                the error threshold for an individual file
                has been reached.

   ACTION:      Check to ensure the log file being processed
                is the correct one.  If so, raise the error
                limit.

   REFERENCES:  Sections C.2.1.1 and C.2.1.2 in this
                guide.

+----------+
| WLE0069W |
+----------+

   TEXT:        %d TOTAL LOG FILE DATA ERROR ENCOUNTERED.

   TYPE:        Warning

   REASON:      Web log extract processing detected nnnn
                errors in log file data.

   ACTION:      If the error threshold was not exceeded,
                execution ends successfully.  If message
                WLE0068E was issued previously, determine if
                error threshold is not high enough or a bad
                log file was processed.

   REFERENCES:  None
```

```
+----------+
| WLE0070W |
+----------+

   TEXT:        %d LOG FILE DATA ERRORS ENCOUNTERED FOR
                PREVIOUS FILE.

   TYPE:        Warning

   REASON:      Web log extract processing detected nnnn
                errors in log file data for the previous
                file.

   ACTION:      Warning message issued when log file data
                errors found without exceeding the error
                threshold.

   REFERENCES:  None

+----------+
| WLE0071E |
+----------+

   TEXT:        LENGTH OF KEYWORD VALUE MUST BE 4.

   TYPE:        Error

   REASON:      The Orgsysid keyword parameter value coded is
                not four bytes long.

   ACTION:      Update the prefix.MICS.PARMS(WLECONF) file
                keyword parameter that is in error to contain
                an Orgsysid value that is four bytes long.

   REFERENCES:  Section C.3.3 in this guide.
```

C.5 WLE Remote Setup

The Web Log Extractor (WLE) utility can be installed and executed on systems where CA MICS is not installed. This may be desirable when you want to process log data from Web servers that cannot be accessed from the system where CA MICS is installed. This is a two-step process.

1 - Installing WLE Utility in a Remote Site
2 - Executing WLE Utility in the Remote Site

C.5.1 Installing WLE Utility in a Remote Site

This procedure is ONLY for installing WLE at a remote site (a site that is not running CA MICS). If you are installing WLE in a CA MICS system, you MUST use the checklist in Section C.2 of this guide.

The primary purpose of installing this utility in a remote site is to provide a way to collect the data generated by servers that are not accessible to the CA MICS system.

Data generated by running WLE utility at the remote site must be transported to the host for processing by CA MICS Web Analyzer.

The WLETAR tar file can be transmitted to another site using a utility such as FTP and uncompressed (inflated), using the tar command at the remote location where the WLETAR is transmitted and stored. The file is found in sharedprefix.MICS.BIN(WLETAR).

The tar command syntax shown below to uncompress WLETAR must be executed in a UNIX System Services shell session:

```
tar -xvof WLETAR
```

There are three files that are extracted from the WLETAR tar file by the above command: wleload.exe, wleload.conf, and wleload.scp.

This completes the WLE utility remote installation.

C.5.2 Executing WLE Utility in the Remote Site

The WLE utility must execute before the normal CA MICS daily update jobs. It extracts IBM HTTP server log data and creates a standard z/OS or OS/390 dataset for input processing into the DAY052 step for a CA MICS unit containing the CA MICS Web Analyzer.

In order to extract log file data, you must run the Web Log Extractor (WLE). This section describes how to run WLE utility in UNIX System Services. If you are executing WLE in a CA MICS system, you MUST use the checklist in Section C.2.1 of this guide.

1 - Tailoring Configuration File
2 - Tailoring WLESCRPT File
3 - Executing WLE Utility in Unix System Services

C.5.2.1 Tailoring Configuration File

Before you can run the script file to generate the desired OS dataset, you must tailor the WLE configuration file, wleload.conf. This is where you define, web servers, their input log files, the output OS data set name and allocation parameters, etc. See Appendix C.3.1.1 and C.3.1.2 of this guide for all parameter specifications.

You can edit wleload.conf file from the UNIX System Services (USS) session prompt:

1. Ensure that the directory where the WLE utility is installed is your current directory.

2. Enter the command below:

   OEDIT wleload.conf

3. Save the file.

C.5.2.2 Tailoring WLESCRPT File

See Appendix Section C.2.1.2.

C.5.2.3 Executing WLE Utility in Unix System Services

You can execute the script file from the Unix System Services (USS) session prompt. Executing the utility is a very simple and quick process once the wleload.conf file is correctly tailored.

Type wleload.scp in the command prompt and press Enter. The script file runs.

The execution of the script file generates two output files:

- wleload.err contains any errors generated during the execution of the wleload.exe program.
- wleload.rpt contains a summary report that informs you about the log files processed, their record count, output dataset name and output record count, and so on.

Note: Data generated by running WLE utility at the remote site must be transported to the host for processing by CA MICS Web Analyzer.