

# CA MICS® Resource Management

## CA ASTEX™ Option Guide

Release 12.9



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

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

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

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

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

The manufacturer of this Documentation is CA.

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

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## 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](mailto: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

---

<b>Chapter 1: THE CA MICS APPLICATION EXTENSION FOR ASTEX</b>	<b>7</b>
1.1 CA MICS Product Interfaces .....	8
1.2 Reports .....	8
1.3 Files .....	9
1.4 Data Sources.....	11
1.4.1 Processing the IDB Directly .....	12
1.4.2 Processing Data from SMF MANx Data Sets .....	13
1.4.3 ASTEX Installation Implications .....	16
1.5 Parameter Specifications.....	18
1.5.1 Complex Level Parameters.....	18
1.5.2 Unit Level Parameters .....	24
1.6 Installation and Operation .....	68
1.6.1 Change the Number of Work Files for DAY097 .....	70
1.6.2 Enable Internal Step Restart .....	73
1.6.3 Implement Incremental Update.....	75
1.7 References.....	77
Messages.....	78



# Chapter 1: THE CA MICS APPLICATION EXTENSION FOR ASTEX

---

The CA MICS Application Extension for ASTEX stores data recorded by the CA ASTEX product into the CA MICS database. ASTEX (short for Automated Storage Expert) is a tool for improving the performance of the storage hierarchy of your MVS/XA and MVS/ESA systems. ASTEX monitors two levels of the storage hierarchy: DASD and cache. ASTEX analyzes the performance of these resources, finds where problems exist, and then maximizes performance by recommending solutions to your I/O response time problems and cache utilization problems. In addition, ASTEX will dynamically optimize the utilization of your cache resources.

The CA MICS Application Extension for ASTEX allows the information recorded by ASTEX to be stored for longer periods of time by using the standard CA MICS database summarization techniques. More importantly, this application extension allows for the merging and joint analysis of detailed ASTEX DASD and cache data with other data in the CA MICS database, such as MVS Batch and Operations Analyzer information.

This application extension provides for standard data conversion by:

- o Converting time units to seconds
- o Decoding hexadecimal fields
- o Calculating derived elements like average response time

It also has interfaces to CA MICS StorageMate and to CA MICS Accounting and Chargeback.

This section contains the following topics:

[1.1 CA MICS Product Interfaces](#) (see page 8)

[1.2 Reports](#) (see page 8)

[1.3 Files](#) (see page 9)

[1.4 Data Sources](#) (see page 11)

[1.5 Parameter Specifications](#) (see page 18)

[1.6 Installation and Operation](#) (see page 68)

[1.7 References](#) (see page 77)

[Messages](#) (see page 78)

## 1.1 CA MICS Product Interfaces

The CA MICS Application Extension for ASTEX is distributed with interfaces to CA MICS StorageMate and to CA MICS Accounting and Chargeback.

### CA MICS StorageMate

-----

The CA MICS StorageMate product provides for integrated storage management reporting by merging information from the CA MICS database to form a comprehensive view of your storage subsystems. It provides for quick and easy access to the ASTEX data in CA MICS. The StorageMate user does not need to know either SAS or the structure of the CA MICS database to get meaningful information on cache and DASD.

### CA MICS Accounting and Chargeback

-----

This application extension supports an interface to CA MICS Accounting and Chargeback that allows for charging on actual cache usage. This is NOT recommended, however, because ASTEX has limits on the number of data sets and jobs it collects information on. This could result in inconsistent results. Refer to Section 1.4.3 of this guide for more information.

Contact the CA MICS Product Support Group if you want to add accounting support for ASTEX data set information.

## 1.2 Reports

This application extension is not distributed with MICS inquiries. As with all CA MICS products, you can create MICS inquiries to report on ASTEX data.

---

## 1.3 Files

The CA MICS Application Extension for ASTEX stores its data in the ASTEX Information Area (AST) in the following files:

- o ASTEX Volume Activity File (ASTAVO)

The ASTAVO file contains information that quantifies cache and DASD I/O activity at the VOLUME level. The source of this information is the ASTEX volume record. The ASTAVO file is summarized by SMF SYSID, SMS Storage Group, VOLSER, and Device Address.

- o ASTEX Storage Group Activity File (ASTASG)

The ASTASG file contains information that quantifies DASD I/O activity at the SMS Storage Group level. The source of this information is the ASTEX volume record. The ASTASG file is summarized by SMF SYSID and SMS Storage Group.

- o ASTEX LCU/Cache Activity File (ASTALC)

The ASTALC file contains information that quantifies cache I/O activity at the Logical Control Unit level. The source of this information is the ASTEX volume record. The ASTALC file is summarized by SMF SYSID and ASTEX Contention Group.

- o ASTEX Data Set Exception File (ASTADS)

The ASTADS file contains information that quantifies cache and DASD I/O activity for individual data sets. The source of this information is the ASTEX data set record. The ASTADS file is summarized by SMF SYSID, SMS Storage Group, VOLSER, and Data Set Name.

- o ASTEX Job Exception File (ASTAJB)

The ASTAJB file contains information that quantifies cache and DASD I/O activity for MVS batch jobs. The source of this information is the ASTEX job record. The ASTADS file is summarized by SMF SYSID, SMS Storage Group, VOLSER, SRM Performance Group, and Jobname.

There are some data elements in the ASTEX Volume Activity file that are comparable to elements in the CA MICS Hardware Device Activity file (HARDVA) in the MVS Hardware and SCP Analyzer. The elements are:

- I/O Events (AVOIOCNT, DVAMEC)
- XA I/O Events (AVOXACNT, DVASSC)
- Average Response Time (AVOAVRES, DVAAVRES)
- Pending Time (AVOPENTM, DVAPEN)
- Average Pending Time (AVOAVPEN, DVAAVPEN)
- Connect Time (AVOCONTM, DVACNN)
- Average Connect Time (AVOAVCNN, DVAAVCNN)
- Disconnect Time (AVODISTM, DVADIS)
- Average Disconnect Time (AVOAVDIS, DVAAVDIS)
- Average IOSQ Time (AVOAVIOS, DVAAVIOS)

A small difference is normal between the two files for the same recording interval for the following reasons:

- o Even if RMF and ASTEX have the same time interval, they will rarely cut collection records at the same instant in time.
- o The values of some of the RMF fields are based on a sampling rate while ASTEX measures every I/O event.

## 1.4 Data Sources

This application extension can process input data in two ways. The preferred method for processing ASTEX information from an ASTEX application running on a local MVS system is to access the ASTEX Interval Database (IDB). This method takes the smallest amount of CPU and DASD overhead. Depending on the version of ASTEX you are running, the IDB may be read directly by CA MICS (ASTEX Version 1.7 and before) or must be converted to an intermediate file by the PMCLAT utility program (ASTEX Version 2.0 and later) and then read by CA MICS. Refer to member PMCLAT in your ASX.CNTL data set if you are using Version 2.0 or later of ASTEX.

ASTEX applications that run on MVS systems remote from your CA MICS Complex can either write their records to the SMF data sets, or you can transmit a copy of the IDB to the MVS system where your CA MICS Complex is located (ASTEX Version 1.7 and before), or you can transmit the PMCLAT output of the IDB to the MVS system where your CA MICS Complex is located (ASTEX Version 2.0 and after).

It is possible to process both SMF and IDB sources in the same update job.

This section contains the following subsections:

- 1 - Processing the IDB Directly
- 2 - Processing Data from SMF MANx Data Sets
- 3 - ASTEX Installation Implications

### 1.4.1 Processing the IDB Directly

The simplest method for processing local ASTEX data into the CA MICS database is to obtain the data from the ASTEX Interval Database (IDB). The ddname that is used in step DAY097 to read IDB files is INPUTIDB, which is defined in member INPUTAST in prefix.MICS.PARMS. For ASTEX Version 1.7 and before, simply code INPUTIDB so it contains the name of the IDB. For ASTEX Version 2.0 and later, the dsname coded for INPUTIDB must represent an intermediate file that was created from the IDB by the ASTEX PMCFLAT utility program. Refer to member PMCFLAT in ASX.CNTL if you need to use PMCFLAT.

To use this method, update INPUTAST and run a JCLGEN for the unit. If you want to process multiple ASTEX IDB files together in the same CA MICS unit, simply concatenate the different IDB files together on the single INPUTIDB DD statement.

Refer to Section 1.5.2.3 of this guide for more detailed information concerning the modification requirements for INPUTAST.

Notes:

- o The ASTEX IDB file or PMCFLAT intermediate file must be addressable from the MVS system on which the CA MICS daily update runs.
- o The duplicate data check in CA MICS checkpoint processing rejects some data because the entire IDB is read by each CA MICS daily update. If your input is from a PMCFLAT intermediate file, you may reduce some of this duplicate data by using the 'DAYS' parameter when running PMCFLAT.

Refer to the ASTEX: Installation and Customization Guide for more information about the ASTEX internal database.

## 1.4.2 Processing Data from SMF MANx Data Sets

In addition to writing its information to the IDB, ASTEX can also write records directly to the SMF data sets. When this method is chosen, CA MICS can access ASTEX information directly from the SMF data.

If this is the method you are using, modify the following CA MICS attributes and parameters:

- o The CA MICS Application Extension for ASTEX needs to be recognized as one of the components that accepts SMF records out of the SMF data set within the CA MICS unit. You accomplish this by modifying your prefix.MICS.PARMS member JCLDEF and adding the "AST" identifier to the SMFRECORDING keyword. Refer to Section 2.3.3.2.1.1 of the CA MICS Planning, Installation, Operation, and Maintenance Guide (PIOM) for more information about the SMFRECORDING keyword.
- o The SMF record types that ASTEX uses need to be defined to the CA MICS complex as being associated with this application extension. You do this by updating the INPUTSOURCE control parameter in the ASTGENIN member of sharedprefix.MICS.GENLIB. The process of defining or updating INPUTSOURCE is fully detailed in Section 1.5.1.1 of this guide.
- o You must modify the INPUTRDR member of prefix.MICS.PARMS to include the name of the data set containing the raw SMF records. This applies not only to the ASTEX Application Extension, but to all CA MICS products that use SMF data. Refer to Section 2.3.3.2.3 of the PIOM for more information about the INPUTRDR member.

This section contains the following subsection:

- 1 - CA MICS and CA SMF Director Interface

### 1.4.2.1 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:

- o Select system identifiers
- o Select or exclude SMF record types and subtypes
- o Use conditional statements for granularity
- o 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 PARM 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)`.

### 1.4.3 ASTEX Installation Implications

How you install ASTEX can significantly impact the amount and meaning of the data stored in CA MICS. We recommend that when installing this application extension, you review how your installation has installed ASTEX.

#### DATA COLLECTION MODE

ASTEX allows you to specify a data collection mode for each volume on your system. The two choices are Detail Mode (DM) or Exception Mode (EM).

If you want detailed job and data set information in ASTEX as well as CA MICS, you should use the data collection mode. If you want the job and data set information in ASTEX and CA MICS to represent true exceptions, you should use the data collection mode with meaningful response time objectives.

#### RESPONSE TIME OBJECTIVE

When ASTEX is using exception mode for data collection, it will only collect data set and job information when the response time objective that you specify is not being met. If you specify a response time objective that is 'too low', then the objective will always be violated and the data set and job data in CA MICS cannot be viewed as true exception data. If you specify a response time objective that 'is too high', then the objective will never be violated and no job or data set information will be available in CA MICS.

#### MANAGED LIST

Whether you run in DM or EM, the maximum number of jobs and data sets kept in the ASTEX managed list is significant. This list contains the jobs and the data sets that ASTEX collects and stores information. If you specify too low a maximum number of elements on the managed list, you may lose significant information. The maximum number of entries on the managed list cannot exceed 100.

The amount of resources ASTEX consumes correlates with the number of jobs and data sets that it manages. You need to determine the optimum balance between data collection requirements and resource usage by ASTEX.

#### VOLUME SELECTION

ASTEX allows you to EXCLUDE volumes from having their I/O events intercepted and analyzed by ASTEX. Volumes that are EXCLUDED from ASTEX will not have any information in the CA MICS database. You could lose significant information if you EXCLUDE the wrong volumes.

#### COLLECTION INTERVAL

ASTEX will write its data set, job, and volume records at the end of each interval. If you have relatively small collection intervals, you will have a finer level of granularity in the data. This means that the data will be more meaningful, but you will need more DASD. If you have relatively large intervals, fluctuations and exceptions will tend to be hidden, but you will minimize your DASD requirements.

You will need to find the optimum balance between data collection requirements and ASTEX's resource usage.

We recommend that the ASTEX interval be the same as your RMF interval, enabling you to correlate information from different data sources, such as SMF, for the same period of time.

#### INTERVAL DATABASE TYPE AND SIZE

Most installations use the ASTEX IDB to collect the volume, job, and data set records. There are two critical considerations concerning the IDB:

- o Whether you define the IDB as a wraparound file
- o How large you allocate the IDB

The parameter IDBWRP indicates to ASTEX that it should continue writing data records to a full IDB by overwriting the oldest records in the IDB. The alternative is to have ASTEX stop recording every time it runs out of space. We recommend that you define the IDB as a wraparound file.

The second critical aspect is the size of the IDB. If you do not allocate an IDB that is large enough to contain all ASTEX data records between your scheduled CA MICS daily update jobs, the IDB may lose valuable information. We recommend that you allocate your ASTEX IDB large enough to contain several days worth of data.

## 1.5 Parameter Specifications

Parameter specifications enable CA MICS sites to easily tailor CA MICS products to their environments. This section contains the complex and unit level parameters provided with this application extension. It contains the following subsections:

- 1 - Complex Level Parameters
- 2 - Unit Level Parameters

### 1.5.1 Complex Level Parameters

Complex level parameters control the processing of the CA MICS Application Extension for ASTEX for all database units within the CA MICS complex. These parameters are stored in the sharedprefix.MICS.GENLIB and sharedprefix.MICS.PARMS data sets. This section contains the following subsections:

- 1 - Specify Data Sources (ASTGENIN)
- 2 - Define ASTEX Account Codes (ASTACCT)
- 3 - Code ASTEX Account Code Derivation Exit (ASTACRT)

### 1.5.1.1 Specify Data Sources (ASTGENIN)

The generation definition statement member for this application extension is `sharedprefix.MICS.GENLIB(ASTGENIN)`. This member defines the ASTEX Information Area (AST), its input sources, its files, and the elements in each file.

File tailoring can be done at the FILE, CYCLES, and NAME/NAMX statement level. Refer to the System Modification Guide, Chapters 4 and 6, for information on file tailoring.

INPUTSOURCE in ASTGENIN defines the input source type and the record types to process. The format of INPUTSOURCE is:

```
INPUTSOURCE SMF|NONSMF recordtype(s)
```

The SMF|NONSMF positional parameter defines to CA MICS whether or not the input records are coming from SYS1.MANx data sets. The recordtype parameter(s) define to CA MICS which SMF record types are ASTEX records. If NONSMF is specified on the INPUTSOURCE statement, the recordtype parameter should be specified as a blank. This application extension can only process up to three different SMF record types.

If your installation processes ASTEX data from SMF SYS1.MANx data sets, you must determine which SMF record types ASTEX is using. Review ASTEX parameters VOLSMFT, DSNSMFT, and JOBSMFT. These parameters have no defaults and must be explicitly coded for ASTEX to write data to SMF. ASTEX allows you to use the same or different record types for VOLSMFT, DSNSMFT, and JOBSMFT.

Example 1: If ASTEX is implemented with VOLSMFT=210, DSNSMFT=210, and JOBSMFT=210, you need to implement support for SMF record type 210. To do this, change INPUTSOURCE from

```
INPUTSOURCE NONSMF
```

to

```
INPUTSOURCE SMF 210
```

Example 2: If you plan to process the 210 SMF records from one ASTEX as well as the IDB directly from another ASTEX, you should change INPUTSOURCE from

```
INPUTSOURCE NONSMF
```

to

```
INPUTSOURCE SMF 210
```

Example 3: If ASTEX is implemented with VOLSMFT=210, DSNSMFT=211, and JOBSMFT=212, you need to implement support for all three SMF types. To do this, change the INPUTSOURCE from

```
INPUTSOURCE NONSMF
```

to

```
INPUTSOURCE SMF 210 211 212
```

We recommend that if you are going to have ASTEX write SMF records directly to SYS1.MANx, you use the same SMF record ID for all three ASTEX records.

Note that you must run an ASTCGEN followed by a unit-level JCLGEN for each unit that contains this application extension for your ASTGENIN changes to take effect.

### 1.5.1.2 Define ASTEX Account Codes (ASTACCT)

This application extension provides up to nine levels of account code for you to associate users and user groups with cache and DASD activity. The account codes are used as summarization keys in the ASTEX Data Set Exception file.

To define the account codes to the application extension, you need to modify member ASTACCT in sharedprefix.MICS.PARMS. ASTACCT defines the number of account codes available in the ASTADS file, the length, and the label associated with each account code. These definitions apply to all CA MICS database units that contain the application extension. For a general discussion of account codes, refer to Section 2.3.1.7 of the CA MICS Planning, Installation, Operation, and Maintenance Guide.

We recommend that you define the same number of account code fields in the application extension that you have defined in the CA MICS DASD Space Analyzer. This will allow integration of VCA and ASTEX data.

#### PREPARATION

Each installation has its own method for associating the work performed through I/O activity with the responsible user or department. Before defining the ASTEX account codes, it is important that you review your installation's accounting and naming standards to:

- o Identify the current organizational coding system (e.g., cost center coding system identifying the division, department, project, and employee) and determine how the account codes are specified. For example, the division and department can be determined by the first three characters of the data set name.
- o Identify if and how the codes are verified to ensure that they correspond to a valid definition. We recommend that account code validation be performed in all cases and that unidentified or invalid account codes be assigned to a special installation overhead account code. This approach provides you with two benefits. First, it groups all invalid codes under the same category and therefore requires less DASD space for storing the ASTEX Information Area files. Second, it enables you to easily determine how much of this overhead activity is taking place.

If you have the DASD Space Analyzer, consider basing your ASTACCT on sharedprefix.MICS.PARMS(VCAACCT). The sample ASTACCT is identical to the sample VCAACCT distributed with the CA MICS DASD Space Analyzer.

DEFINING ACCOUNT CODES

The statement format is:

```
level length 'descriptive title'
```

where:

level = The level of importance of each account code, with level 1 being the most important and the highest level number being the least important. The levels are defined sequentially starting with 1. You can define up to nine levels. This parameter is required.

length = The length of the account code. The length may range from 1 to 30. This parameter is required.

descriptive title = The title that describes the account code. The length of the title is 1 to 40 characters. The title is embedded in single quotes ('). This is a required parameter.

SAMPLE

A sample ASTACCT is as follows. It is identical to the sample provided in VCAACCT.

```
1 3 'DIVISION'  
2 3 'PROJECT'  
3 3 'USER'
```

### 1.5.1.3 Code ASTEX Account Code Derivation Exit (ASTACRT)

After you have defined the number of account codes to be carried in the ASTADS file, your next step is to code the ASTEX Account Code Derivation exit (ASTACRT). This exit is located in sharedprefix.MICS.PARMS.

ASTACRT is invoked for each ASTEX Data Set record encountered. You can use any CA MICS data element that is available in the ASTADS file to determine values for ASTEX account codes. In addition, there are five work fields labeled DSNODE1 through DSNODE5 that can also be used to derive account codes. DSNODE1 contains the first node of the data set name. DSNODE2 contains the second node of the data set name, and so on.

We recommend basing the ADSACTx on these DSNODEx fields.

#### SYNTAX

The ASTACRT exit may contain any SAS statements valid in a DATA step except RETURN and DELETE. Coding RETURN or DELETE statements will cause a U410 abend.

#### SAMPLE

The following sample SAS code is the code that is distributed with this application extension. It is located in data set sharedprefix.MICS.PARMS. This sample is similar to the sample distributed in VCAACRT.

```
IF LENGTH(DSNODE1) LT 8 AND INDEX(DSNODE1, ' ') LE 3
  THEN ASTACT1 = '***';
  ELSE ASTACT1 = SUBSTR(DSNODE1,1,3);
IF LENGTH(DSNODE1) LT 8 AND INDEX(DSNODE1, ' ') LE 6
  THEN ASTACT2 = '***';
  ELSE ASTACT2 = SUBSTR(DSNODE1,4,3);
ASTACT3 = DSNODE2;
```

## 1.5.2 Unit Level Parameters

Unit level parameters control the processing of the CA MICS Application Extension for ASTEX within each database unit. They are stored in library prefix.MICS.PARMS. This section contains the following subsections:

- 1 - ASTEX System Code Generation (ASTPGEN)
- 2 - ASTEX Processing Options (ASTOPS)
- 3 - INPUTRDR and INPUTAST PARMS Members
- 4 - Code the Application Code Derivation Exit (ASTAURT)
- 5 - Database Space Modeling (DBMODEL)

### 1.5.2.1 ASTEX System Code Generation (ASTPGEN)

The ASTEX System Code Generation (ASTPGEN) inputs the unit level parameters and generates SAS code executed by step DAY097. The generated code contains SAS macros that dictate the type of input the application extension processes.

ASTPGEN validates that the options you select in ASTOPS are consistent with the ASTEX Input DD statements you define in INPUTAST.

The JCL for ASTPGEN resides in prefix.MICS.CNTL.

### 1.5.2.2 ASTEX Processing Options (ASTOPS)

This section shows you how to specify the operational statements that control CA MICS Application Extension for Astex 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 ASTOPS member contains the following statements, described in detail below.

```
OPTIONS IDB|NOIDB SMF|NOSMF
```

```
-----
```

The OPTIONS statement has two positional parameters that define which input sources to process.

IDB|NOIDB specifies whether or not the CA MICS DAILY update job is to use an ASTEX IDB file as input. If IDB is specified, the CA MICS DAILY reads information from the file identified by the DD statement INPUTIDB defined in prefix.MICS.PARMS(INPUTAST). For users of ASTEX Version 2.0 and later, the PMCFLAT utility must first be run to create an intermediate file that can be read by the CA MICS DAILY job. This parameter is required and there is no default.

SMF|NOSMF specifies whether or not the CA MICS DAILY update job is to use an SMF data set as input. If SMF is specified, the CA MICS DAILY reads information from the INPUTSMF DD statement. This parameter is required and there is no default.

You must select at least one of the two input options. Coding "OPTIONS NOIDB NOSMF" is not valid. You can read both IDB and SMF files in the same CA MICS DAILY update. This might be the case when running CA MICS on a system with a local ASTEX IDB and accepting ASTEX data for input in SMF format transmitted from a remote processing installation.

The contents of prefix.MICS.PARMS(INPUTAST) are included in the CA MICS DAILY update JCL even if NOIDB is specified on the OPTIONS statement. In this case, be sure that INPUTAST contains comments only.

Refer to Section 1.5.2.3 of this guide for more information about coding INPUTAST.

Example 1: An ASTEX IDB file will be read directly.

```
OPTIONS IDB NOSMF
```

Example 2: The PMCFLAT output of an ASTEX Version 2.0 IDB will be read along with ASTEX SMF records from a remote site.

```
OPTIONS IDB SMF
```

Example 3: ASTEX data that has been written to the SMF data sets will be read.

```
OPTIONS NOIDB SMF
```

```
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 zero (0).  
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:

- o Run cccPGEN
- o 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 AVO AJB ADS ALC ASG

if neither MULTWORK or NOMULT parameters are specified.

The following files are eligible for multiple WORK support:

- AVO Astex Volume Activity File
- AJB Astex Job Exception File
- ADS Astex Data Set Exception File
- ALC Astex LCU/Cache Activity File
- ASG Astex Storage Group Activity File

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(JCLGEN) or by specifying
*       DAILY in prefix.MICS.PARMS(JCLGEN) and
*       executing prefix.MICS.CNTL(JCLGEN).
*
*       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 three 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.
SORT	Sort intermediate work file contents, eliminate duplicate input data, and

prepare for DETAIL cycle creation.

DBUPDATE	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.

#### 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(JCLGEN) or by specifying
*       DAILY in prefix.MICS.PARMS(JCLGEN) and
*       executing prefix.MICS.CNTL(JCLGEN).
*
*       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.

- o 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.
- o 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.

- o 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.

- o 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.

- o 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.

- o 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.

- o 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.

- o 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):

- o Incremental update checkpoint file,  
//IUCKPT DD DSN=prefix.MICS.ccc.IUCKPT,.....
- o Incremental update DETAIL  
//IUDETAIL DD DSN=prefix.MICS.ccc.IUDETAIL,.....
- o 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.
- o 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.
  - o 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.
  - o 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.
  - o 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 IEHPRGM) 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).

- o You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.
- o 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.

- o The INCRDETAIL 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 INCRDETAIL keyword on the continuation line.

- o 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).

- o 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.
- o 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.
- o 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.

## INCRSPLIT

-----

This statement is optional and defaults to this:

INCRSPLIT IGNORE

Specify the following if you want the incremental update job for this product to get input measurement data from the output of the SPLITSMF job. The optional `data_set_allocation_parameters` are used by the SPLITSMF job when creating the measurement data file for this product.

INCRSPLIT USE `data_set_allocation_parameters`

Note: INCRSPLIT is ignored when you specify INCRUPDATE NO.

This option would be used when multiple products in a single unit database are enabled to incremental update. The SPLITSMF job performs the same function for incremental update jobs as the DAILY job DAYSMF step performs for the DAYnnn database update steps.

- o The SPLITSMF job dynamically allocates, catalogs, and populates `prefix.MICS.ccc.IUSPLTDS` data sets for each product in the unit database for which you specified both the INCRUPDATE YES and INCRSPLIT USE parameters. These data sets are then deleted after processing by the appropriate INCRccc job.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRSPLIT keyword on each continuation line.
- o INCRSPLIT 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.

Specify the following or accept the default if you want the incremental update jobs for this product to get their input measurement data from the data sets specified in the INPUTccc (or INPUTSMF) member of `prefix.MICS.PARMS`:

INCRSPLIT IGNORE

When you specify INCRSPLIT IGNORE, this product will NOT participate in SPLITSMF job processing.

Example 1:

```
INCRSPLIT USE  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):

```
INCRSPLIT USE  STORCLAS=MICSTEMP UNIT=SYSDA  
INCRSPLIT      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.

### 1.5.2.3 INPUTRDR and INPUTAST 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 - INPUTAST PARMS Member JCL Definition

### 1.5.2.3.1 INPUTRDR PARMS Member JCL Definition

The INPUTRDR member of prefix.MICS.PARMS defines the DD statements that specify the SMF input data for all products in the units that use SMF data. The INPUTSMF DD statements are used by the DAYSMF step, which reads and splits the SMF data into separate files, one for each CA MICS product.

The INPUTRDR member defines the input SMF data sets when DAYSMF FILES PERMANENT or TEMPORARY is specified in the prefix.MICS.PARMS(JCLDEF) member.

An INPUTSMF DD statement is required for each SMF data source. A worksheet for preparing the INPUTRDR member is provided below. If you change the contents of the INPUTRDR member, you must regenerate the CA MICS DAILY job using either JCLGENU or JCLGEN in prefix.MICS.CNTL.

For example, if you want to read your SMF data from a data set named SMF.DAILY.DATA, you would change the INPUTRDR member to contain the name of the SMF data set as follows:

```
//INPUTSMF DD DISP=SHR,DSN=SMF.DAILY.DATA
```

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 directly.

If DAYSMF has been deactivated by the DAYSMF OFF statement specified in the prefix.MICS.PARMS(JCLDEF) member, then each component step will read its input from their corresponding prefix.MICS.PARMS(INPUTccc) member.

```
+-----+
| INSTALLATION PREPARATION WORKSHEET: SMF Input DD Statements |
| PARS Library Member is INPUTRDR |
+-----+
|
| This definition is required to specify the DD statement for SMF data |
| which will be read by the DAILY CA MICS job. |
|
| //@ |
| //@ WARNING: ALWAYS MAKE CHANGES IN PARS(INPUTRDR) AND NOT |
| //@ &CNTL(DAILY). |
| //@ CHANGES MADE TO &CNTL(DAILY) |
| //@ WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN. |
| //@ |
| //INPUTSMF DD DISP=SHR,DCB=BUFNO=3,DSN=_____ |
| // DD DISP=SHR,DCB=BUFNO=3,DSN=_____ |
|
+-----+
```

Figure

1-1a. INPUTRDR JCL Definition Worksheet

### 1.5.2.3.2 INPUTAST PARMS Member JCL Definition

The INPUTAST member of prefix.MICS.PARMS contains the DD statements to specify the input data for the CA MICS Hardware and SCP Analyzer. The //INPUTIDB DD, the //INPUTSMF DD, and, optionally, the //SMFDRCTR DD statements are used by the DAY097 step of the DAILY job.

The INPUTAST member contains the input SMF/IDB data sets when ANY of the following JCLDEF options is used:

- o DAYSMF OFF is specified
- o SMFDRCTR/SMFDIRECTOR AST is specified
- o DAYSMF EXCLUDE AST is specified

A //INPUTIDB DD statement is required when the input data source is an ASTEX IDB file.

A //INPUTSMF DD statement is required when the input data source is SMF data.

A //SMFDRCTR DD statement is only required when CA SMF Director split indices are used.

A worksheet for preparing the INPUTAST member is provided below in Figure 1-1b.

If you change the contents of the INPUTAST member, you must regenerate the CA MICS DAILY job using either JCLGENU or JCLGEN in prefix.MICS.CNTL.

The INPUTAST member contains the JCL necessary to allocate a local IDB, a transmitted copy of a remote IDB, or the PMCFLAT output from an unloaded IDB (PMCFLAT is required when using an IDB created by ASTEX Version 2.0 or later).

The ASTEX IDB files must use the ddname INPUTIDB. If there are multiple IDB files for input, concatenate them together on the one DD statement. If you are running ASTEX Version 2.0 or later, as well as pre-2.0 versions, IDBs and PMCFLAT files may be concatenated if they have the same block size.

Refer to member PMCFLAT in your ASX.CNTL data set for information about the PMCFLAT utility. Note the DAYS parameter may be used to reduce the amount of data written by PMCFLAT, which should reduce the amount of duplicate data

dropped in the CA MICS DAILY job and improve processing times. Modify your job scheduling system to make sure PMCFLAT runs each day before the CA MICS DAILY job. The contents of prefix.MICS.PARMS(INPUTAST) are included in the CA MICS DAILY update JCL even if NOIDB is specified on the OPTIONS statement. In this case, be sure that INPUTAST contains comments only.

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 1.4.2.1, CA MICS and CA SMF Director Interface, of this guide.

```
+-----+
| INSTALLATION PREPARATION WORKSHEET: INPUTAST JCL Definitions |
| |
| PARMs Library Member is INPUTAST |
+-----+
|
| 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(INPUTAST) AND NOT |
| //@ &CNTL(DAILY). |
| //@ CHANGES MADE TO &CNTL(DAILY) |
| //@ WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN. |
| //@ |
| //INPUTIDB 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 1-1b. INPUTAST JCL Definition Worksheet

### 1.5.2.4 Code the Application Code Derivation Exit (ASTAURT)

The ASTEX Job Exception file (ASTAJB) has an application code element (ASTAPU) in its summarization sequence. This element allows you to classify batch jobs for differentiation at higher timespans where JOBNAME is no longer available. This application element is derived in the ASTAURT exit located in prefix.MICS.PARMS. Because each installation is unique, you must code this exit based on installation specific information.

The first step to coding ASTAURT is to understand the JOBNAME and performance group number standards that are in effect at your installation. We recommend that you use these two fields to set a value for the ASTAPU field.

The classic case is to extract the application identifier out of the JOBNAME. This is possible when the JOBNAME standards are that this application identifier is of a given length and at a given offset in the JOBNAME. For example, numerous shops require that the first three bytes of the JOBNAME identify the application. Therefore, payroll jobs all start with the prefix PAY and inventory jobs all start with INV. In this case, code the ASTAURT exit to extract the application identifier from the first three bytes of the JOBNAME.

Another common methodology is to use the batch job's performance group to indicate whether the job is test or production. This can be achieved when test batch jobs use different performance groups than production batch jobs. In this case, you can code ASTAURT to base the ASTAPU field on the jobs status.

Example 1: Your JOBNAME standards require that the application identifier is always the first three bytes of JOBNAME. You should use this application identifier in ASTAPU.

```
ASTAPU = SUBSTR(JOB,1,3);
```

Example 2: Your installation runs all test jobs in performance group 5 while all production jobs run in performance groups 10 through 15.

```
IF PERFGRP EQ 5 THEN ASTAPU = 'TEST';  
ELSE ASTAPU = 'PROD';
```

Example 3: You want to differentiate between test and production jobs for each application group. Because the ASTAPU field is 12 bytes long, you will use the first three bytes of the ASTAPU field as an application identifier and bytes 4 through 7 as an indicator of whether the job is test or production.

```
ASTAPU = SUBSTR(JOB,1,3);  
IF PERFGRP EQ 5 THEN SUBSTR(ASTAPU,4,4) = 'TEST';  
ELSE SUBSTR(ASTAPU,4,4) = 'PROD';
```

### 1.5.2.5 Database Space Modeling (DBMODEL)

Database space modeling allows you to determine the number of tracks of DASD space required for the CA MICS database. See Section 2.3.4 (and its subsections) of the CA MICS Planning, Installation, Operation, and Maintenance Guide (PIOM) for a complete description of the process.

The FILE statement in prefix.MICS.PARMS(DBMODEL) is required for database space modeling. The default FILE statements are generated from the CYCLES statements in sharedprefix.MICS.GENLIB(ASTGENIN) and copied to prefix.MICS.PARMS(DBMODEL) when a new unit is generated or the product is added to an existing unit.

#### Data Retention Specifications (FILE Statements)

Data retention specifications tell the CA MICS database how many cycles of data to save for each file in each supported timespan, for both online and archive mode.

The default retention period for each file is shown in Figure 1-2 as it will appear in DBMODEL. If '00' shows as the retention value for the timespan, the file is not supported in the timespan. To add support, you must perform database tailoring as described in Chapter 6 of the CA MICS System Modification Guide.

										Archive
										Cut-Off
										-----
File	DETAIL	DAYS	WEEKS	MONTHS	YEARS	TABLES	WEEKS	MONTHS		
										-----
FILE	AST	ASTADS	03	00	00	06	00	00	000	000
FILE	AST	ASTAJB	03	00	00	06	00	00	000	000
FILE	AST	ASTALC	00	10	09	06	00	00	000	000
FILE	AST	ASTASG	00	10	09	06	00	00	000	000
FILE	AST	ASTAV0	01	03	04	06	00	00	052	024

Figure 1-2. Default File Retention Periods

Review the above FILE retention specifications and adjust as required for your site before running a CYCLEGEN.

To actually perform the space modeling, submit the jobs as described in Section 2.3.4.2 of the PIOM.

## 1.6 Installation and Operation

Be sure to review Sections 1.4 and 1.5 of this guide before proceeding to the installation checklists in the CA MICS Planning, Installation, Operation, and Maintenance Guide (PIOM). These sections provide helpful information about the data sources and the input parameters you will need to supply for proper installation of this application extension.

We recommend initially installing this application extension in a TEST unit. Select about one hour of representative measurement data for each ASTEX input source. Analysis of the data in the TEST unit can provide useful information for coding the DATA statements in DBMODEL (Section 1.5.2.5 of this guide). After tests are complete, install this product in your production CA MICS environment.

If you prefer to limit the amount of data set and job data in CA MICS, there are two ways to do so:

- o Decrease the number of data sets and jobs ASTEX manages

To decrease the number of data sets and jobs managed by ASTEX, modify the MLSIZE parameters in ASTEX. ASTEX will work from the top down in deciding which data sets and jobs not to manage, meaning that the busiest data sets and jobs will be managed. Note: this method decreases the data set and job data in both ASTEX and CA MICS.

- o Delete or consolidate volume-specific I/O

To delete or consolidate volume-specific I/O (I/O to the volume VTOC and the volume WVDS), create a USRSADS user exit in prefix.MICS.USER.SOURCE using the guidance provided in Sections 4.3.1 (coding restrictions) and 4.3.1.2 (coding indirect exits) in the CA MICS System Modification Guide. Contact the CA MICS Product Support Group if you need assistance in implementing this exit.

If you plan to install this product in an existing database unit, follow the checklist in Section 3.8.2 of the PIOM.

If you plan to install this product in a new database unit, follow the checklist in Section 3.8.3 of the PIOM.

The CA MICS Application Extension for ASTEX executes as step 097 in operational jobs. There will be a step in the DAILY, WEEKLY, MONTHLY, and YEARLY jobs for this product.

This section contains the following subsections:

- 1 - Change the Number of Work Files for DAY097
- 2 - Enable Internal Step Restart
- 3 - Implement Incremental Update

### 1.6.1 Change the Number of Work Files for DAY097

To change the number of work files used in CA MICS Astex Application Extension processing in Step DAY097, follow the checklist provided below for each unit.

See Section 1.5.2.2 of this guide for a detailed description of the statement syntax.

```
*****  
*                                                                 *  
*           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.

## 1.6.2 Enable Internal Step Restart

To enable the internal step restart in CA MICS Astex Application Extension, follow the checklist provided below.

See Section 1.5.2.2 of this guide for a detailed description of the statement syntax.

```
*****
*                                     *
*           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.

### 1.6.3 Implement Incremental Update

To implement incremental update in CA MICS Astex Application Extension, follow the checklist provided below.

See Section 1.5.2.2 of this guide for a detailed description of the statement syntax.

```
*****
*                                     *
*          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).
  - o 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.
  - o 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.

## 1.7 References

The following manuals provide more information about CA-ASTEX and the CA MICS products mentioned in this chapter.

- o CA-ASTEX: An Introduction
- o CA-ASTEX: Installation and Customization Guide
- o CA-ASTEX: Operations and Facilities Guide
- o CA-ASTEX: Storage Manager User Guide
- o CA-ASTEX: DASD Manager User Guide
- o CA-ASTEX: Cache Manager User Guide
- o CA MICS Planning, Installation, Operation, and Maintenance Guide
- o CA MICS System Modification Guide
- o CA MICS StorageMate Guide
- o CA MICS Accounting and Chargeback Guide
- o CA MICS Hardware and SCP Analyzer Guide
- o CA MICS Batch and Operations Analyzer Guide

## Messages

Messages generated by the CA MICS Application Extension for ASTEX are listed below. 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 reason for the message, user action where appropriate, and applicable references to documentation.

The following type codes are used to categorize the messages:

Information	Designates a note that documents an application extension option or potentially important feature in the data.
Warning	Designates a condition in either the data or control statements that does not affect the application extension's operation but may lead to unexpected results.
Error	Designates a problem has been encountered with a control statement or data that will prevent a successful run of this application extension.

### Messages:

-----

AST00001I	This is an informational message indicating the data set and member being processed in ASTPGEN.
AST00016E	This is an error message indicating that the statement found in ASTOPS is invalid. Remove the invalid statement and rerun ASTPGEN.
AST00020E	This is an error message indicating that ASTPGEN encountered a second OPTIONS statement in ASTOPS. Remove the invalid statement and rerun ASTPGEN.
AST00021E	This is an error message indicating that the first positional parameter on the ASTOPS OPTIONS statement is invalid. Valid choices are IDB or NOIDB. Fix the parameter and rerun ASTPGEN.
AST00023E	This is an error message indicating that the third positional parameter on the ASTOPS OPTIONS

- statement is invalid. Valid choices are SMF and NOSMF. Fix the parameter and rerun ASTPGEN.
- AST00025E This is an error message indicating that ASTPGEN encountered an OPTIONS statement that did not specify any input source. At least one input source is required. Fix the OPTIONS statement and rerun ASTPGEN.
- AST00030E This is an error message indicating that ASTPGEN encountered two INPUTIDB DD statements in member INPUTAST. Delete the repeated statement and rerun ASTPGEN.
- AST00032E This is an error message indicating that ASTPGEN encountered an unknown DD statement in INPUTAST. Delete the invalid DD statement and rerun ASTPGEN.
- AST00035E This is an error message indicating that ASTPGEN found a mismatch in ASTOPS, which had IDB coded and INPUTAST that did not have a INPUTIDB DD statement. Fix the mismatch and rerun ASTPGEN.
- AST00036E This is an error message indicating that ASTPGEN found a mismatch in ASTOPS, which had NOIDB coded and INPUTAST that had a INPUTIDB DD statement. Fix the mismatch and rerun ASTPGEN.
- AST00040I This is an informational message shows each line of input from the member being processed.
- AST00045I This is an informational message indicating that ASTPGEN generated member \$ASTMSTR and ended normally.
- AST00066W This is a warning message indicating that the record type is not supported and will not be processed. Processing continues for valid record types.
- AST00067W This is a warning message indicating that only the first ten "not supported record type" messages will be printed. The "option deleted" count will indicate how many records were not processed. Contact CA MICS Product Support for information on support for this version of ASTEX data.

- AST00070W This is a warning message indicating the data is from an unsupported version of ASTEX. The job continues processing but new fields will be ignored. The supported version and the version found in the data are printed in the message. Call the CA MICS Product Support Group for more information on supported versions of ASTEX.
- AST00072W This is a warning message indicating that ASTEX records that are shorter than expected will be ignored. These records will not be processed but the job continues to process valid records. You should investigate the source of these records.
- AST00076I This is an informational message indicating records were dropped by the user exit specified.
- AST00104I This is an informational message indicating that an invalid SMF record type was encountered in the ASTEX daily format routine. See Section 1.5.1.1 of this guide for more information.