CA Common Services for z/OS

Service Desk Integration Guide

Release 14.1.00

CA Product References

This document references some of the following CA products:

- CA 1® Tape Management
- CA 7® Workload Automation
- CA 11™ Workload Automation Restart and Tracking
- CA ACF2™
- CA Allocate™ DASD Space and Placement
- CA Audit
- CA Automation Point
- CA Balancing
- CA Bundl®
- CA Database Analyzer™ for DB2 for z/OS
- CA Datacom®/AD
- CA Data Compressor™ for DB2 for z/OS
- CA DB2
- CA Deliver™
- CA Disk™ Backup and Restore
- CA Dispatch™
- CA Earl™
- CA Endevor® Software Change Manager
- CA Fast Check® for DB2 for z/OS
- CA Fast Index® for DB2 for z/OS
- CA Fast Load for DB2 for z/OS
- CA Fast Recover® for DB2 for z/OS
- CA Fast Unload® for DB2 for z/OS
- CA IDMS™
- CA IDMB™/DB
- CA Insight™ Database Performance Monitor for DB2 for z/OS
- CA Index Expert™ for DB2 for z/OS
- CA JARS®
- CA JARS® Resource Accounting

- CA Jobtrac™ Job Management
- CA Log Analyzer™ for DB2 for z/OS
- CA Mainframe Software Manager™ (CA MSM)
- CA Merge/Modify™ for DB2 for z/OS
- CA MIA Tape Sharing
- CA MIC Message Sharing
- CA MICS® Resource Management
- CA MII Data Sharing
- CA MIM™ Resource Sharing
- CA NetMaster® File Transfer Management
- CA NetMaster® Network Automation
- CA NetMaster® Network Management for SNA
- CA NetMaster® Network Management for TCP/IP
- CA NetMaster® Network Operations for TCP/IP
- CA NetSpy™ Network Performance
- CA Network and Systems Management
- CA NSM System Status Manager
- CA OPS/MVS® Event Management and Automation
- CA Partition Expert™ for DB2 for z/OS
- CA Plan Analyzer® for DB2 for z/OS
- CA Quick Copy for DB2 for z/OS
- CA Rapid Reorg® for DB2 for z/OS
- CA RC/Extract™ for DB2 for z/OS
- CA RC/Migrator™ for DB2 for z/OS
- CA RC/Query® for DB2 for z/OS
- CA RC/Secure™ for DB2 for z/OS
- CA RC/Update™ for DB2 for z/OS
- CA Recovery Analyzer™ for DB2 for z/OS
- CA Roscoe®
- CA Scheduler® Job Management
- CA SYSVIEW® Performance Management
- CA Service Desk (Service Desk)
- CA Spool™ Enterprise Print Management

- CA SQL Ease® for DB2 for z/OS
- CA SYSVIEW® Performance Management
- CA TCPAccess™ Communications Server for z/OS
- CA TLMS Tape Management
- CA Top Secret®
- CA TPX™ Session Management for z/OS
- CA Value Pack for DB2
- CA Vantage™ Storage Resource Manager
- CA View®
- CA XCOM™
- CA Workload Control Center

Contact CA

Before contacting CA Support, do the following:

- Verify that TCP/IP is active and functional. Issue the following z/OS console command:

  ```
  DISPLAY TCPIP
  ```

- Verify that CAICCI is active and functional. Issue the following z/OS console command:

  ```
  DISPLAY A,name
  ```

  name

  Identifies the name of the CAICCI started task.

- Verify that the desired CA Service Desk server is accessible. Logon to TSO on the system where CAICCI is active and issue the following TSO command:

  ```
  PING hostname
  ```

  hostname

  Identifies the host name assigned to the server where CA Service Desk is running.

- Verify that the CAISDI/soap server address space is active. Issue the following z/OS console command:

  ```
  DISPLAY A,name
  ```

  name

  Identifies the name of the CAISDI/soap started task.

- Verify that CA Service Desk is started and functional on the desired server.
- Review the SYSLOG for any messages that might provide an indication of a problem condition.
- For problems with CA products that use CAISDI/elmds, make sure the CAISDI/elmds address space is active. Issue the following z/OS console command:

  `DISPLAY A,`*`cdyfapi`*

  cdyfapi

  Identifies the name of the CAISDI/elmds started task (or job).
- For problems with CA products that use the standalone CAISDI/med address space (not elmds), make sure the standalone CAISDI/med is active. Issue the following z/OS console command:

  `DISPLAY A,`*`medname`*

  medname

  Identifies the name of the standalone CAISDI/med started task.
- For problems with CA products that use the standalone CAISDI/els address space (not elmds), make sure the standalone CAISDI/els interface has been properly initialized. Issue the following z/OS console command:

  `S CASDIELS,CMD=ELSLIST`

  This produces a report of all products currently defined to the standalone CAISDI/els interface. This report shows the event status of each product's events. Each product begins on a new page with the product and its current status being displayed in the report headings. Make sure the product in question appears in the report and that “Status: Enabled” is displayed for that product.

Contact CA Support

For your convenience, CA provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA 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 product documentation, you can send a message to techpubs@ca.com.

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

Contents

Chapter 1: Introduction

This guide describes the CA Service Desk Integration (CAISDI), which enables CA mainframe z/OS products to automatically generate requests and incident tickets in CA Service Desk. CAISDI is a CA product that runs on Windows and UNIX systems. This chapter provides an overview of mainframe integration and outlines the CA products that support this interface.

This section contains the following topics:

Integrating CA Mainframe Products

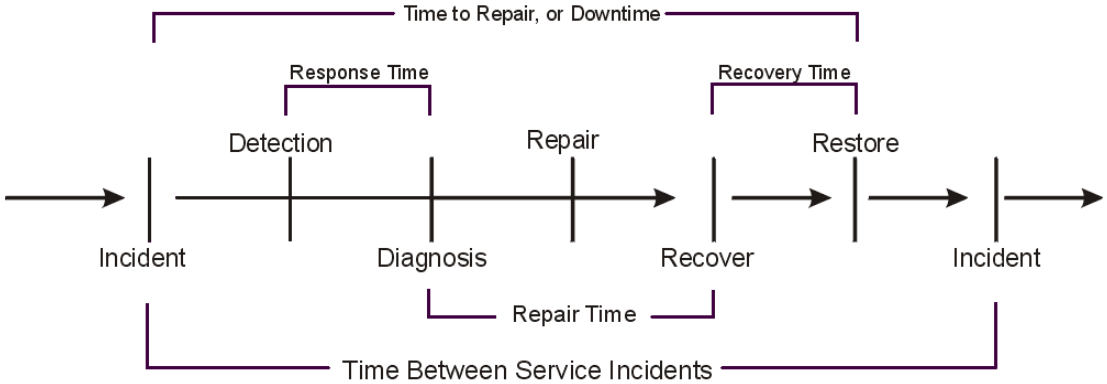
CAISDI uses Web services technology to provide an open and flexible mechanism for the CA mainframe management products to simply and efficiently generate tickets in CA Service Desk. The automatic detection and escalation of mainframe problems through CA Service Desk empowers IT to take a proactive, business driven, or “Service Aware” approach to incident escalation and service management.

Reducing the Extended Incident Lifecycle

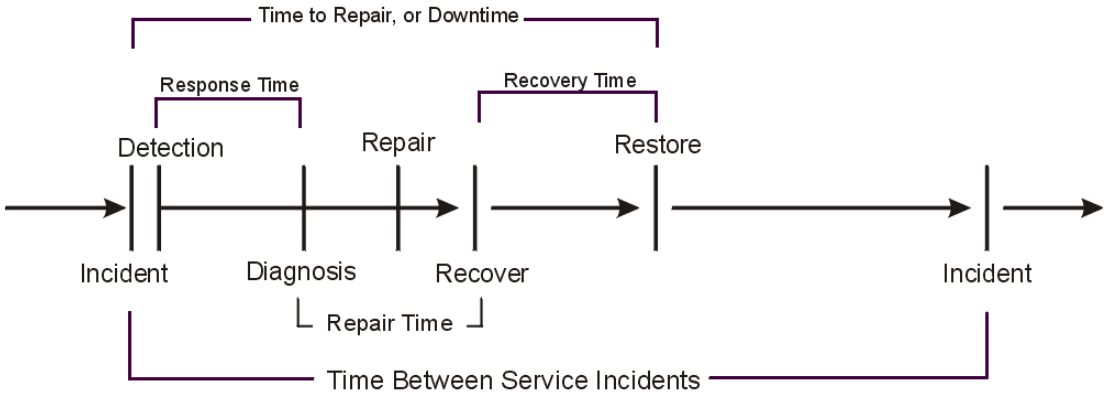
Timely notification and escalation of mainframe exception conditions is vital to meeting IT service commitments in support of business objectives. Automated incident escalation increases IT efficiency through reduced lag time between problem detection and resolution. By compressing the incident detection and resolution stages, the length of IT service outages is reduced to minimize business impact. The resulting increase in IT service availability helps deliver quantifiable business advantages.

The following diagrams depict how automated incident detection reduces the elapsed time to repair a service incident and reduces the Extended Incident Lifecycle.

Extended Incident Lifecycle



Compressing the Extended Incident Lifecycle



Open and Flexible Interface

CAISDI uses Web services to deliver an open and flexible method for enterprise-wide incident management. CAISDI is delivered as part of CA Common Services as a value-added capability that is available to all CA mainframe management products for zero cost.

Sample Incident Scenario

The following implementation scenario describes how CAISDI delivers an open and flexible method for enterprise-wide incident management in the banking industry. Banking was selected for this example as it uses a heterogeneous IT environment that has strict service level availability requirements and well defined problem escalation procedures.

The fictitious United Banana Growers Bank of the Bahamas (UBGBB) will be used in this example. The UBGBB runs several IBM z/OS mainframe systems that run IMS, CICS, and DB2 to drive their core banking application such as Auto Teller Machine transactions and Bank Branch Customer processing. Their communications network is predominantly TCP/IP based and they run a plethora of 'WinTel' servers that are located both within the Central IT operation glasshouse and across the Wide Area Network in remote Bank Branches.

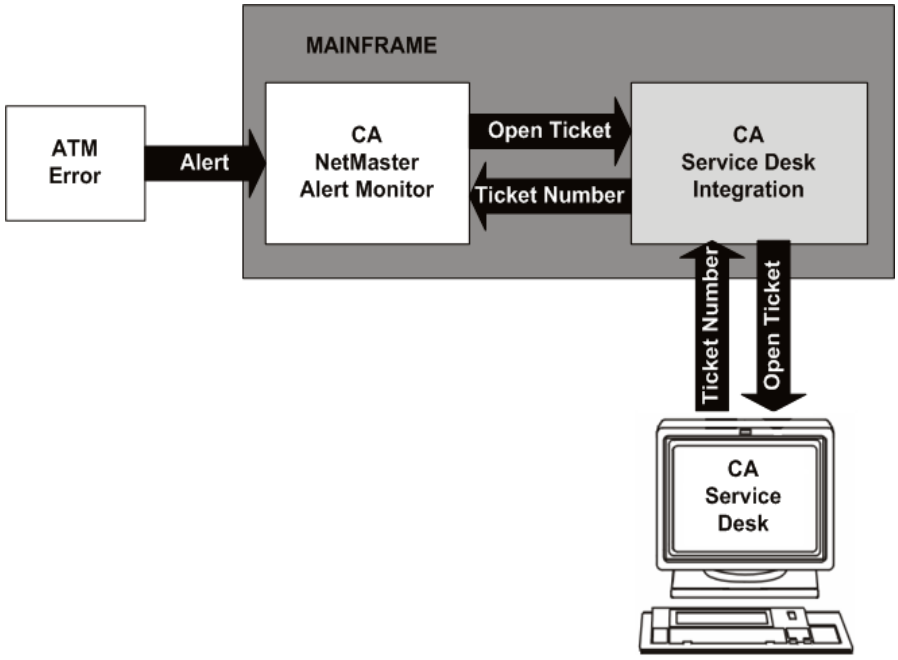
The UBGBB uses CA mainframe management products to help guarantee the mainframe application, systems, and network service reliability. Additionally they use various CA NSM products to manage their distributed servers, network and other devices. They have standardized on CA Service Desk to manage all exceptions impacts in their IT environment.

Severity 1 alert

The CA NetMaster Network Management products, proactively monitoring for failures in any of the bank branch network nodes, determines that a Nassau bank branch ATM has stopped communicating with IMS on the mainframe. When CA NetMaster Network Management identifies that the Nassau branch ATM node is no longer contactable, the node is flagged as failed and an alert is generated in the CA NetMaster Alert Monitor to warn the mainframe operations team of the problem. The CA NetMaster Alert Monitor has been configured so that the alert notification process for this specific alert automatically and simultaneously opens a priority 1 request record in CA Service Desk. The opened request contains the details of the resource name and the time the network node failed, and provides a summary of the identified failure.

This priority 1 request record appears in CA Service Desk and, based on the resource name and location, it is automatically assigned or escalated to the local Nassau ATM server failure response team. CA Service Desk also returns the assigned request number to CA NetMaster Network Management, where it is added to the alert record to allow correlation between the alert and request record. CA Service Desk also notifies the mainframe operations team that the request has been opened and assigned to the Nassau branch ATM server failure response team. A Service Impact is generated to notify IT management that a service outage has occurred due to an ATM server hardware failure, and that the ATM failure response team is taking the necessary recovery action.

The following graphic illustrates the mainframe-to-CA Service Desk incident lifecycle from creating an alert, to opening a request, to receiving a request number.



Alert-based Request Creation

In the scenario, automatic notification and escalation to CA Service Desk occurred before the Nassau bank branch staff noticed the ATM had a problem. This proactive management resulted in significantly reducing the mean time to identify and recover from the failure.

Manual Request Creation

In addition to raising an impact based on a specific alert instance, CA NetMaster Network Management also supports the manual creation of tickets in CA Service Desk. This requires a user to enter the TT command against an alert, which results in a request being opened containing the specific alert details.

Alert Auto Forward Request Creation

You can configure the CA NetMaster Alert Monitor to automatically forward tickets in CA Service Desk based on an alert forwarding filter. This filter defines the alert selection criteria upon which to open a request of a specific type in CA Service Desk. This auto forwarding alert filter contains a Boolean expression that determines which alerts are to be sent by the filter. For example, you could define filter criteria to generate a request for all severity 1 alerts for a specific resource type or matching resource name.

Note: For details on creating CA NetMaster Network Management tickets, see the CA NetMaster Network Management documentation**.**

CA Mainframe Products That Support the Interface

The following z/OS mainframe-based CA products, r11 or later, can use CAISDI to open CA Service Desk tickets:

- CA IDMS
- CA 1 Tape Management
- CA Allocate DASD Space and Placement
- CA Disk Backup and Restore
- CA TLMS Tape Management
- CA Vantage Storage Resource Manager
- CA Value Pack for DB2
- CA 7 Workload Automation
- CA Deliver
- CA Dispatch
- CA JARS Resource Accounting
- CA Jobtrac Job Management
- CA MIM Resource Sharing
- CA OPS/MVS Event Management and Automation
- CA Spool Enterprise Print Management
- CA SYSVIEW Performance Management
- CA View
- CA Data Compressor for DB2 for z/OS
- CA Database analyzer for DB2 for z/OS
- CA Fast Index for DB2 for z/OS
- CA Fast Load for DB2 for z/OS
- CA Fast Recover for DB2 for z/OS
- CA Fast Unload for DB2 for z/OS
- CA Index Expert for DB2 for z/OS
- Ca Log Analyzer for DB2 for z/OS
- CA Merge/Modify for DB2 for z/OS
- CA NetMaster Network Management for TCP/IP
- CA NetMaster Network Management for SNA

- CA NetMaster Network Automation
- CA NetSpy Network Performance
- CA NetMaster Network Operations for TCP/IP
- CA NetMaster File Transfer Management
- CA MICS Resource Management
- CA Partition Expert for DB2 for z/OS
- CA Plan Analyzer for DB2 for z/OS
- CA Quick Copy for DB2 for z/OS
- CA Rapid Reorg for DB2 for z/OS
- CA RC/Extract for DB2 for z/OS
- CA RC/Migrator for DB2 for z/OS
- CA RC/Query for DB2 for z/OS
- CA RC/Secure for DB2 for z/OS
- CA RC/Update for DB2 for z/OS
- CA Recovery Analyzer for DB2 for z/OS
- CA SQL Ease for DB2 for z/OS

Additional z/OS-based CA products may integrate with CA Service Desk in the future.

CAISDI is a zero-cost CA common component service. Customers who wish to use the product can do so without any licensing concerns or costs.

The following sections describe product-specific information provided to CAISDI for the products listed above. This information appears in the content of the CA Service Desk tickets that are opened.

CA Product

The following CA product supports CA Service Desk Integration.

CA IDMS Product

Sites running CA Service Desk can have CA IDMS automatically open CA Service Desk tickets when a CA IDMS Central Version region abends. This provides your organization with an immediate recorded notification of the identified problem so that it can be quickly addressed.

CA Products

Sites running CA Service Desk can have any of the following five key CA z/OS storage products, 11.5 or later, automatically open CA Service Desk tickets to track product-related events that must be addressed for optimal processing:

- CA 1 Tape Management
- CA Allocate DASD Space and Placement
- CA Disk Backup and Restore
- CA TLMS Tape Management
- CA Vantage Storage Resource Manager

A priority code will be used to indicate the criticality of events created. This functionality provides true enterprise-class integration by recording events from various mainframe systems in a centralized CA Service Desk. Each of the products identified above can create CA Service Desk tickets for selected events such as resource shortages or possible component failures. For example, CA 1 Tape Management can create an event when the minimum number of Data Set Name Block (DSNB) records in the Tape Management Catalog (TMC) falls below a user defined threshold. A specific technician can be automatically assigned to follow up on this problem.

The CA products use the CA Service Desk Integration/event library support (CAISDI/els) component of CA Common Services. This component provides for the externalization and localization of event text. Event members are delivered with the CA products listed above and can be easily customized. In addition, this component allows you to:

- Conveniently identify the technician to be assigned to each product's tickets by defining the names in a parameter file on z/OS.
- Optionally define a unique CA Service Desk asset name for each product to be associated with each request opened by that product.

CA Workload Control Center Products

Sites running CA Service Desk can have any of the following job management products, r11 or later, automatically open CA Service Desk tickets to track product-related events that must be addressed for optimal processing:

- CA 7 Workload Automation
- CA Jobtrac Job Management
- CA Scheduler Job Management

The CA Workload Control Center products provide a comprehensive and flexible enterprise management solution for all IT scheduling environments. This includes both cross-platform job management cooperative implementations as well as the mainframe-centric business models.

As a major component of IT business process management, the z/OS Job Management engines control critical workload assets that require accurate and timely execution and coordination with external processes. Notification of exception conditions and events that occur during processing of these flows is vital to meeting your business objectives.

Through integration with the CA Service Desk, the z/OS Job Management engines provide a true “Service Aware” paradigm for workload management. The automatic generation of problem tickets for job and application health events reduces costs and increases productivity by expediting the notification and resolution of complex issues and eliminates the need for manual data entry when problems occur. Through the use of Web services technology, the CA Service Desk enables a true self-management methodology for Enterprise Job Management.

CA Deliver

Sites running CA Service Desk can have CA Deliver automatically open CA Service Desk tickets for unexpected product abends. This provides your organization with an immediately recorded notification of the identified problem so that it can be quickly addressed allowing for minimal interruption in CA Deliver system operations.

CA Dispatch

Sites running CA Service Desk can have CA Dispatch automatically open CA Service Desk tickets for unexpected product abends with CADDSPL. This provides your organization with an immediately recorded notification of the identified problem so that it can be quickly addressed before causing more serious problems with CA Dispatch that could impact your system operations and the archiving of your critical reports.

CA JARS Resource Accounting

Sites running CA Service Desk can have CA JARS Resource Accounting automatically open CA Service Desk tickets when there is an unrecoverable error condition during operation of the CA JARS Report Writer.

CA Service Desk tickets are generated when the Report Writer abends with any User Completion Code, or with any System Completion Code that does not end with '22' (Operator Cancel or Automated System Termination) and does not end with '37' (Disk resource shortages).

CA MIM Resource Sharing

Sites running CA Service Desk can have CA MIM Resource Sharing automatically open CA Service Desk tickets for unexpected product abends. This provides your organization with an immediately recorded notification of the identified problem so that it can be quickly addressed before causing more serious problems with CA MIM Resource Sharing that could impact the efficient sharing of your organization's DASD, tape, and console resources.

CA OPS/MVS Event Management and Automation

Sites running CA Service Desk can have CA OPS/MVS Event Management and Automation automatically open CA Service Desk tickets for a variety of problems detected internally by CA OPS/MVS Event Management and Automation. This provides your organization with an immediately recorded notification of the identified problem so that it can be quickly addressed before causing more serious problems with CA OPS/MVS Event Management and Automation or your system operations. The following types of problems are logged to CA Service Desk:

- Product abends
- Shortages of Process Blocks, which are necessary for automation
- Failure to respond to internal MSF (Multi-System Facility) ping requests
- OSF (Operator Server Facility) TSO server transactions that exceed their elapsed time or output line limits
- AOF (Automated Operations Facility) rules that fail to complete due to errors

CA Spool Enterprise Print Management

Sites running CA Service Desk can have CA Spool Enterprise Print Management automatically open CA Service Desk tickets to track product-related events and unexpected abends that must be addressed for optimal processing. You can designate which events you want to track by setting the value for a new SDIREQ parameter added to the message table in CA Spool Enterprise Print Management. This provides your organization with an immediately recorded notification of the identified problem so that it can be quickly addressed allowing for minimal interruption in CA Spool Enterprise Print Management system operations.

CA SYSVIEW Performance Management

Sites running CA Service Desk can have CA SYSVIEW Performance Management automatically open CA Service Desk tickets when CA SYSVIEW Performance Management detects an internal problem that would cause a dump to be created. This provides your organization with an immediately recorded notification of the identified problem so that it can be quickly addressed before causing more serious problems with CA SYSVIEW Performance Management or your system operations.

CA View

Sites running CA Service Desk can have CA View automatically open CA Service Desk tickets for unexpected product abends. This provides your organization with an immediately recorded notification of the identified problem so that it can be quickly addressed allowing for minimal interruption in CA View system operations.

CA DB2 Tools Products

Sites running CA Service Desk can have any of the following CA DB2 Tools products automatically open CA Service Desk tickets when one of their batch or online programs abends:

- CA Database Administration Solutions for DB2 for z/OS
  - CA Fast Index for DB2 for z/OS
  - CA Fast Load for DB2 for z/OS
  - CA Fast Unload for DB2 for z/OS
  - CA Partition Expert for DB2 for z/OS
  - CA RC/Extract for DB2 for z/OS
  - CA RC/Migrator for DB2 for z/OS
  - CA RC/Query for DB2 for z/OS
  - CA RC/Secure for DB2 for z/OS
  - CA RC/Update for DB2 for z/OS
- CA Backup & Recovery Solutions for DB2 for z/OS
  - CA Fast Check for DB2 for z/OS
  - CA Fast Recover for DB2 for z/OS
  - Ca Log Analyzer for DB2 for z/OS

- CA Merge/Modify for DB2 for z/OS
- CA Quick Copy for DB2 for z/OS
- CA Recovery Analyzer for DB2 for z/OS

- CA Database Performance Management Solutions for DB2 for z/OS
  - CA Database analyzer for DB2 for z/OS
  - CA Data Compressor for DB2 for z/OS
  - CA Index Expert for DB2 for z/OS
  - CA Plan Analyzer for DB2 for z/OS
  - CA Rapid Reorg for DB2 for z/OS
  - CA SQL Ease for DB2 for z/OS
- CA Value Pack for DB2 for z/OS

  The CA DB2 Tools products use the CA Service Desk Integration/event library support (CAISDI/els) component of CA Common Services. This component provides for the externalization and localization of event text. Event members are delivered with the DB2 Tools products listed previously and can be easily customized. In addition, this component allows you to:
  - Conveniently identify the technician to be assigned to each product's tickets by defining the names in a parameter file on z/OS.
  - Optionally define a unique CA Service Desk asset name for each product to be associated with each request opened by that product.

CA NetMaster Network Management Products

Sites running CA Service Desk can have any of the following CA NetMaster Network Management products, r11 or later, automatically open CA Service Desk tickets to track product-related events that must be addressed for optimal processing:

- CA NetMaster Network Management for TCP/IP
- CA NetMaster Network Operations for TCP/IP
- CA NetMaster Network Management for SNA

- CA NetMaster Network Automation
- CA NetMaster File Transfer Management
- CA NetSpy Network Performance

CA NetMaster Network Management products send details of selected alerts to CA Service Desk to create CA Service Desk tickets. This provides your organization with an immediately recorded notification of the identified problem so that it can be quickly addressed before causing more serious problems with CA NetMaster Network Management products or your system operations.

CA MICS Resource Management

Sites running CA Service Desk can have CA MICS Resource Management automatically open CA Service Desk tickets when there is a CA MICS Resource Management-generated abend in CA MICS Resource Management operational jobs. These types of abends include:

- CA MICS Resource Management checkpoint errors
- Processing data from unknown systems
- Insufficient input data
- Improper user exit usage

CA Service Desk tickets are generated when the Report Writer abends with any User Completion Code, or with any System Completion Code that does not end with '22' (Operator Cancel or Automated System Termination) and does not end with '37' (Disk resource shortages).

CAISDI Component Overview

The CAISDI interface is a set of services that open CA Service Desk tickets from the z/OS environment. The tickets can be opened directly by CA products or they can be opened on their behalf, depending upon the requirements of each specific product using the interface.

Components

CAISDI consists of the following components in FMID CDYFE10:

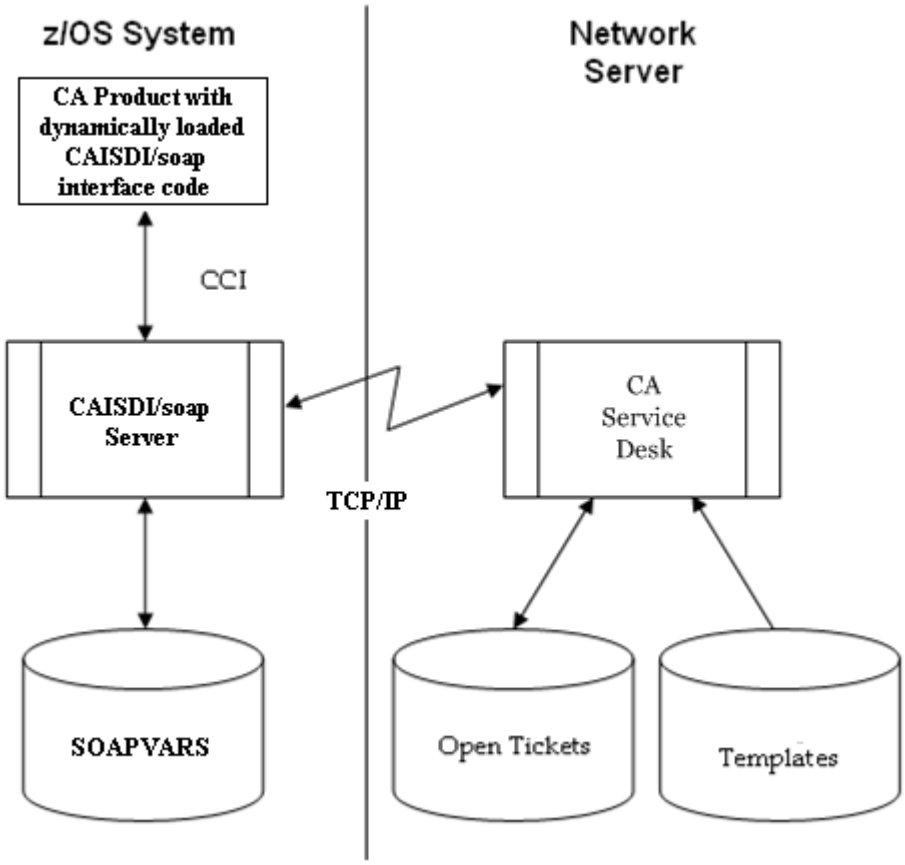
Name	Description
CAISDI/soap	The z/OS SOAP (Simple Object Access Protocol) server/client communicates with the CA Service Desk. It manages the communication over TCP/IP to the CA Service Desk. It also provides the basic mechanisms that allow CA products to open CA Service Desk tickets. This component is required for all CA Service Desk integration. It is a client with respect to Service Desk and a server with respect to mainframe applications.
CAISDI/med	The Mainframe Event Director component monitors the z/OS environment and opens CA Service Desk tickets on behalf of CA products and other system components. It provides a way to open CA Service Desk tickets when the CA product is unable to do so, such as the case of an abend. It also opens solicited tickets for the products that use this interface component.
CAISDI/els	The Event Library Support component provides a mechanism for CA products to open CA Service Desk tickets for events they detect directly. Supported events are defined in an event library that contains the customizable text and symbolic parameters used in the CA Service Desk request. The CA z/OS product family requires installation of this component.
CAISDI/elmds	This combined med and els component provides the functionality of both CAISDI/med and CAISDI/els. CAISDI/elmds is loaded into a permanent address space and adds the ability to update and close tickets. It also allows med and els to initialize even if the soap server is down..

For product details, see the table in Required Components by CA Products (see page 31) in this guide.

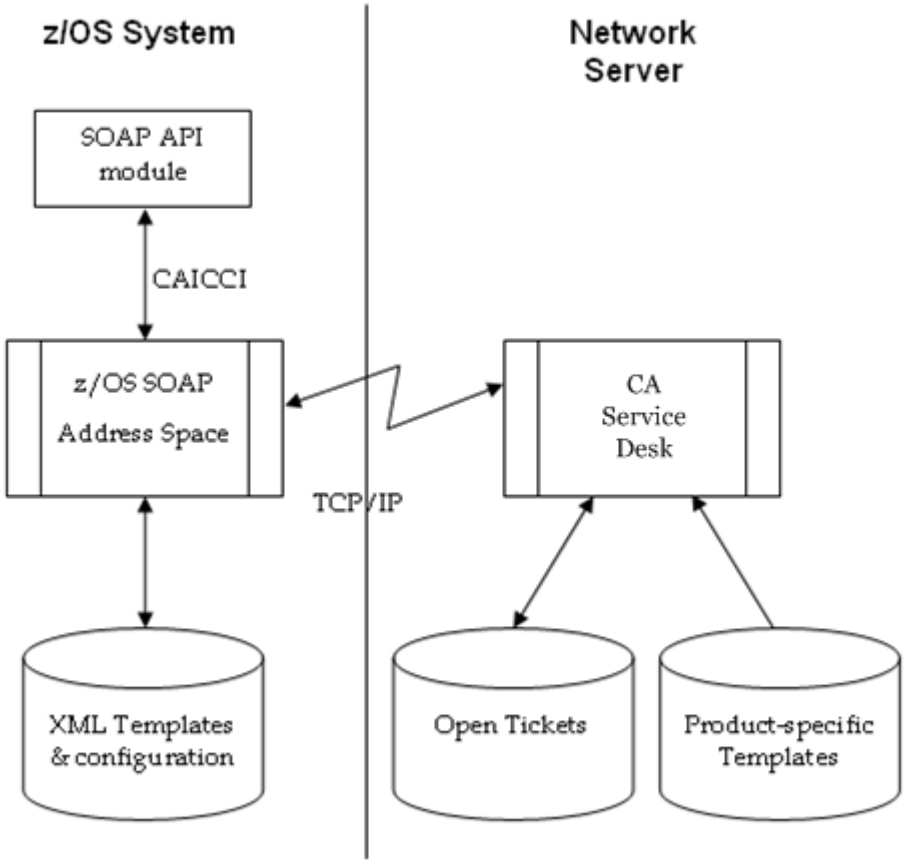
About CAISDI/soap

The CAISDI/soap server component executes in a z/OS address space and provides interface routines to allow applications to communicate with it using the CA Technologies CAICCI. The CAISDI/soap server address space acts as a server to mainframe applications. The CAISDI/soap server address space acts as a SOAP client to Service Desk and communicates using TCP/IP to the target CA Service Desk server using SOAP over HTTP 1.1. This component provides the basic mechanisms to open CA Service Desk tickets. Normally, these tickets are request tickets, but if the CA Service Desk is configured to be ITIL compliant, the tickets will be incident tickets.

With Version 14.0, the CAISDI/soap component is enhanced to support any release of CA Service Desk using r11 or later WSDL and to make use of the IBM XML Parser. The new procedure is CASOAPE.



For r12.0 or earlier (using proc CASOAP)



About CAISDI/elmds

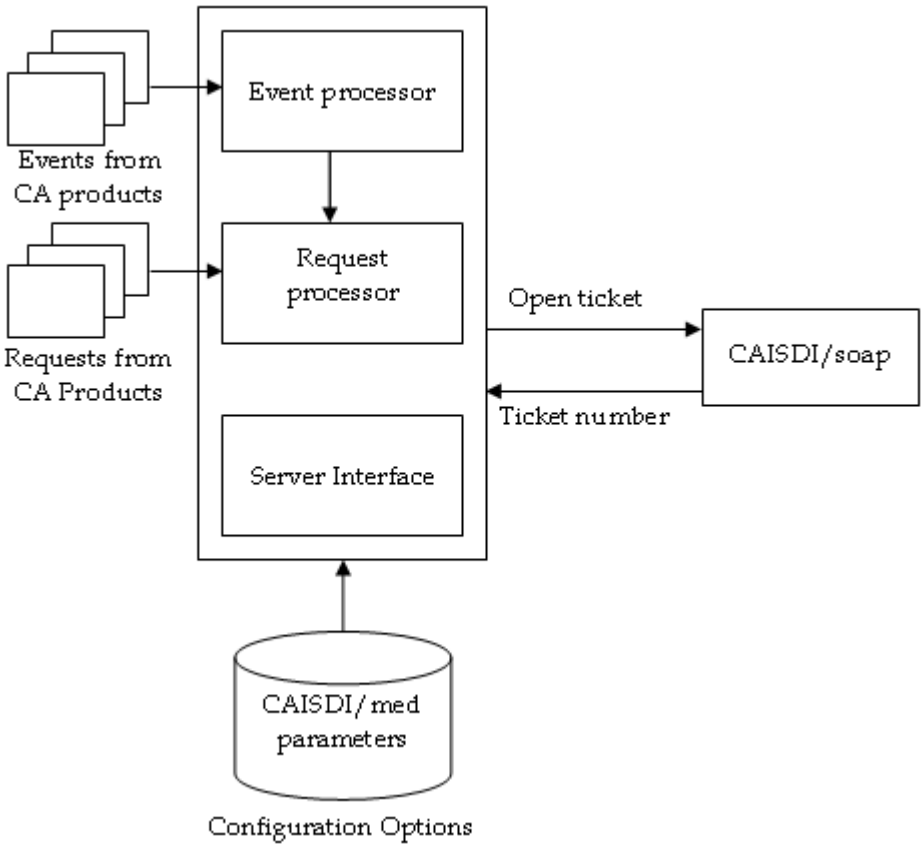
The CAISDI/elmds component combines the functionality of CAISDI/med and CAISDI/els.

Beginning with CA Commons Services for z/OS Release 14.1, CAISDI/elmds is the primary component replacing CAISDI/med and CAISDI/els. CAISDI/med and CAISDI/els are included with Release 14.1 only in case a problem develops with the use of CAISDI/elmds.

The CAISDI/elmds facility executes in a permanent address space on a z/OS operating system.

About CAISDI/med and the med Portion of CAISDI/elmds

The following diagram shows the various components involved in generating a CA Service Desk ticket from a given instance of a z/OS system event involving a recognized CA product. The explanation follows the flow from left to right on the diagram.



The CAISDI/med facility executes as an address space on a z/OS operating system. Through customizable parameters, CAISDI/med is able to monitor the z/OS system for selected events, both solicited and unsolicited, that are generated by other z/OS-based CA products. Events such as product abends from CA products defined to CAISDI/med are captured and used to automatically open a CA Service Desk ticket on the defined CA Service Desk server. Events recognized by CAISDI/med that warrant a CA Service Desk ticket are presented directly to CAISDI/med by the defined z/OS-based CA product through an internal call mechanism.

CAISDI/med relies on two components to provide a connection between the z/OS operating system and the CA Service Desk server:

- CAICCI
- CAISDI/soap

Both of these components are provided on the CA Common Services for z/OS product tape. TCP/IP is used to communicate between the CAISDI/soap component and the CA Service Desk server; therefore, TCP/IP is considered a prerequisite as well.

As previously mentioned, CAISDI/med provides a means to determine which z/OS-based CA product events are to be captured. Similarly, CA Service Desk tickets generated by CAISDI/med can be directed to one or many CA Service Desk servers. CA Service Desk tickets can be customized to assign appropriate attributes to a given ticket so that it can be categorized, assigned, and prioritized as needed to ensure effective event tracking and awareness.

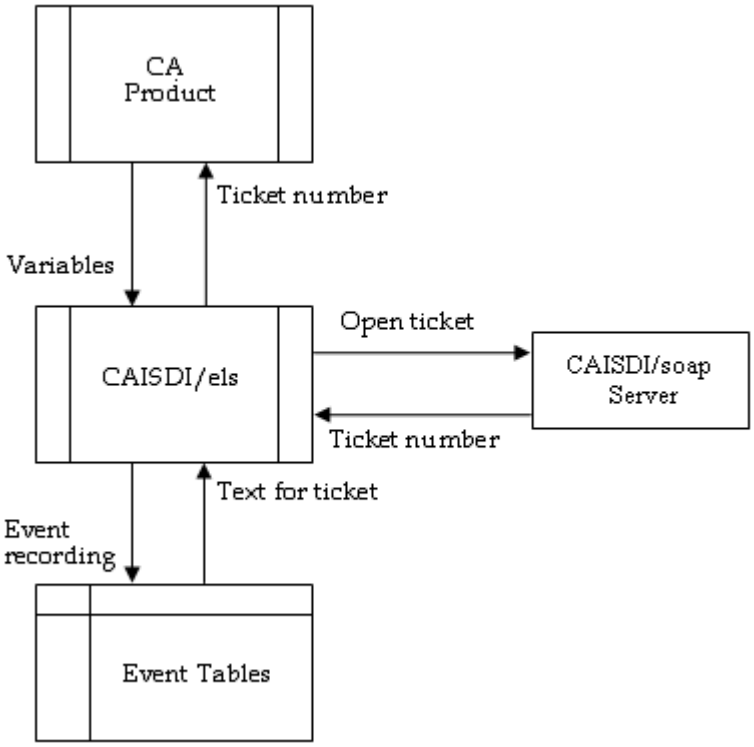
Once the CAISDI/med address space is initialized, it establishes a means of examining selected z/OS system events that occur for recognized CA products. These system events are captured in an encapsulated fashion and presented to the event manager task in the CAISDI/med address space. Using configuration options supplied through CAISDI/med initialization statements, the event manager task formats a request for a CA Service Desk ticket, and passes the request to the request manager task within the CAISDI/med address space.

In addition to asynchronous system events, CAISDI/med provides a means for CA products to directly communicate requests that pertain to events of interest to individual CA products. Again, using configuration options, the request manager constructs a request for a CA Service Desk ticket and passes the request to the appropriate server interface task within the CAISDI/med address space.

A given instance of a CAISDI/med service interface task is attached for each unique instance of a CA Service Desk server that is defined to CAISDI/med through the configuration options. In other words, there is a CAISDI/med server task for each unique CA Service Desk URL address that is supplied by way of CAISDI/med initialization statements. A ticket is formulated and then passed to the CAISDI/soap address space.

About CAISDI/els and the els Portion of CAISDI/elmds

CAISDI/els provides a mechanism to open CA Service Desk tickets with a high degree of control over the textual content of the ticket's summary and description fields. The supported events are defined in the event library, CAI.CAIEVENT. The event members are provided by each of the products that use the CAISDI/els component. The text for each event may be provided in several languages. An initialization module reads all of the events, selects the language version, and loads the event definitions into Extended CSA for reference by CAISDI/els when a CA Service Desk ticket is opened. The CA product needing to open a ticket passes a list of variables to CAISDI/els. These variables describe the specifics about the event. CAISDI/els acquires the text for the event and resolves symbolic references using the variables supplied by the calling program. It then opens the CA Service Desk ticket through the CAISDI/soap component, using the completed text.



The CAISDI/els component also provides for reporting on the CA Service Desk tickets opened by CAISDI that are still in open status.

Required Components by CA Product

The components required for each CA mainframe product to integrate with CA Service Desk depend upon the architecture of that CA mainframe product. Use the following table to determine which components are required for a given product:

CA Product	CAISDI/med	CAISDI/els	CAISDI/soap
CA IDMS	X		X
CA 1 Tape Management		X	X
CA Allocate DASD Space and Placement		X	X
CA Disk Backup and Restore		X	X
CA TLMS Tape Management		X	X
CA Vantage Storage Resource Manager		X	X
CA Value Pack for DB2		X	X
CA 7 Workload Automation		X	X
CA Deliver	X		X
CA Dispatch	X		X
CA JARS Resource Accounting			X
CA Jobtrac Job Management		X	X
CA MIM Resource Sharing	X		X
CA OPS/MVS Event Management and Automation	X		X
CA Spool Enterprise Print Management	X		X
CA SYSVIEW Performance Management	X		X
CA View	X		X
CA Data Compressor for DB2 for z/OS		X	X
CA Database analyzer for DB2 for z/OS		X	X

CA Product	CAISDI/med	CAISDI/els	CAISDI/soap
CA Fast Check for DB2 for z/OS		X	X
CA Fast Index for DB2 for z/OS		X	X
CA Fast Load for DB2 for z/OS		X	X
CA Fast Recover for DB2 for z/OS		X	X
CA Fast Unload for DB2 for z/OS		X	X
CA Index Expert for DB2 for z/OS		X	X
Ca Log Analyzer for DB2 for z/OS		X	X
CA Merge/Modify for DB2 for z/OS		X	X
CA NetMaster Network Management for TCP/IP			X
CA NetMaster Network Management for SNA			X
CA NetMaster Network Automation			X
CA NetMaster			X
CA NetMaster Network Operations for TCP/IP			X
CA NetMaster File Transfer Management			X
CA NetSpy Network Performance			X
CA MICS Resource Management			X
CA Partition Expert for DB2 for z/OS		X	X
CA Plan Analyzer for DB2 for z/OS		X	X

CA Product	CAISDI/med	CAISDI/els	CAISDI/soap
CA Quick Copy for DB2 for z/OS		X	X
CA Rapid Reorg for DB2 for z/OS		X	X
CA RC/Extract for DB2 for z/OS		X	X
CA RC/Migrator for DB2 for z/OS		X	X
CA RC/Query for DB2 for z/OS		X	X
CA RC/Secure for DB2 for z/OS		X	X
CA RC/Update for DB2 for z/OS		X	X
CA Recovery Analyzer for DB2 for z/OS		X	X
CA SQL Ease for DB2 for z/OS		X	X

Security Considerations

Access to the CA Service Desk requires a user ID and password. The user ID and password can be stored in the CAISDI/soap server Environment Variables file (DDNAME ENVVAR) or it may be provided by the other CAISDI components or it may be provided by another CA program that is using CAISDI. The libraries or control data sets holding the user ID and password must be secured. Additionally, you should use the Secured Socket Layer (SSL) to keep the ID and password encrypted as it is relayed through the network.

For CAISDI/soap server prior to Version 14.0, the user ID and password are stored in its configuration file allocated to DDNAME CONFIG.

ITIL Considerations

The CA Service Desk ITIL (Information Technology Infrastructure Library) configuration option supports additional data objects not used in the standard out-of-the-box CA Service Desk product. In ITIL configurations, what are referred to in this guide as *tickets* are actually *incident tickets*; in non-ITIL configurations they are *request tickets*. CA Service Desk displays Incident and Request queues for ITIL configurations, while for non-ITIL configurations, only the Request queue displays. Also, in ITIL configurations, the term “Asset” is replaced with the term “Configuration Item”. If you are running Service Desk r12.6 or later, you must load the ITIL version of the data files using PDM_LOAD. See Loading Data Files (see page 38). Service Desk r12.6 or later only runs in ITIL-mode so the ITIL version data files must be loaded. For more information about the ITIL interface, see the CA Service Desk documentation, specifically the *ITIL User Guide*.

WSDL Level Considerations

CA Service Desk can be configured with either the r6 level WSDL, the r11 level WSDL, or both. Which WSDL levels to run depends on which CA products you are running that are configured to open Service Desk tickets, and at what levels for some products. All products that use CAISDI/els or CAISDI/med require CA Service Desk to run with the r11 level WSDL. Products that do not use CAISDI/els or CAISDI/med (only use CAISDI/Soap) need the r11 level WSDL if the product has been upgraded to a level that supports the r11 level WSDL, otherwise the r6 level WSDL must be installed. To determine which products do not require either CAISDI/els or CAISDI/med, see Required Components by CA Product (see page 31).

For example, the CA NetMaster suite of products are all products that require the r6 level WSDL on the CA Service Desk server platform if the CA NetMaster product is running at a level that does not support the r11 level WSDL. The CA NetMaster documentation states whether the version of CA NetMaster does or does not support the CA Service Desk r11 level WSDL.

On the mainframe side CA Common Services for z/OS delivers two sample procs to run the CAISDI/Soap address space. The CASOAP proc is for communicating with a CA Service Desk platform running the r6 level WSDL and its associated URL. The CASOAPE proc is for communicating with a CA Service Desk platform running the r11 level WSDL and its associated URL.

Chapter 2: How to Implement CAISDI

This chapter briefly describes the requirements and recommendations for installing and running CAISDI.

Migration to CAISDI/elmds

Beginning with CA Common Services for z/OS Release 14.1, the CAISDI/elmds procedure (CDYFAPI) replaces CAISDI/med (CASDIMED) and CAISDI/els (CASDIELS). If problems arise during the transition, you can run any combination of elmds with med and els. See CAISDI/elmds Backout Procedure (see page 113).

Consider the following migration issues before starting CAISDI:

- The elmds configuration file (CDYFCNFG) is processed before any other files. If you run either CAISDI/med (CASDIMED) or CAISDI/els (CASDIELS) (not both) as specified by the product suite above, you must specify NO_ELS or NO_MED appropriately in the CAISDI/elmds configuration file.

  The current med procedure CASDIMED allocates DDNAME MEDPARMS for its initialization statements. The current els procedure CASDIELS allocates SYSIN to process its primary initialization statements which can then specify additional data sets for input. The elmds procedure CDYFAPI has DDNAME MEDPARMS for initializing the med support and DDNAME ELSSTART for the primary initialization statements for els support.

  For the existing med parameters referred to as MEDPARMS, copy the existing MEDPARMS used by CASDIMED to a new library and allocate it to DDNAME MEDPARMS in the CDYFAPI procedure.
- On the MED INIT statement, the DEBUGMSG and REUSE parameters are deprecated. A message is issued but it does not cause termination.
- On the MED SERVER statement, the range for the TIMEOUT parameter has been changed from 10 to 3600 seconds to 20 to 180 seconds. The default remains at 30 seconds.
- For the existing els primary initialization statements, all products must copy their DEFAULT and DEFINE statements to a common file allocated to ELSSTART in CDYFAPI. All products are initialized once at the start of CDYFAPI. Unlike the existing els interface, it is not possible to remove or define a product after elmds has been started.
- The EVENTLIB data sets and members can remain as they are currently defined except that PRODNAME on the SETUP statement is limited to 100 characters.
- &PARM is no longer supported for els. Update any configuration files appropriately.

- On the ELS DEFAULTS and DEFINE statement, parameters USD_ID and USD_PW are deprecated. A message is issued but it does not cause termination. Use the URL_SYMBOL if necessary and see the CAISDI/soap Server documentation regarding URL_SYMBOL and associated user IDs and passwords. The configuration data set is no longer needed to hold Service Desk user IDs and passwords.
- The elmds interface provides an optional initialization statement (SYNTAX_CHECK_ONLY in CDYFCNFG) for elmds to verify syntax in the various configuration files and then terminate. With this parameter the elmds address space can be started to verify syntax changes, even if there is another instance active.
- The elmds log (CDYFLOG) contains messages and potentially trace information. CDYFLOG is spun automatically based on parameters in CDYFCNFG, or through operator commands.
- Messages for els and med are now written to CDYFLOG and not issued as a WTO in the application address space. New message IDs and text are used.
- If CASDIMED is run using the same MEDNAME as specified in the elmds MEDPARMS, then elmds must be started with either a different MEDNAME or with NO_MED in CDYFCNFG.
- If either the CASDIELS or CASDIMED procedure must be run, then elmds must first be started with NO_ELS or NO_MED (respectively) in CDYFCNFG.

  If both NO_MED and NO_ELS are specified in CDYFCNFG, CDYFAPI can be terminated after starting CASDIMED and CASDIELS.
- The string &SYSEDATE is a system variable that can appear in the els eventlib SUMMARY or DESCRIPTION sections. The variable is substituted with the current date in the form *dd/mmm/yyyy*. The support in this product is for English only. If another language is required, either use a user variable or open an enhancement request (DAR) requesting a change.

System Requirements

The following table summarizes the requirements for installing and running CAISDI:

Type	Description
Operating System	The minimum z/OS operating system is 1.10.
Virtual Storage	CA recommends using a region size of 4 MB or larger for each component.

Type	Description
Common Storage Area	CAISDI/med uses approximately 24 KB of the extended common service area (ECSA) for loading common storage resident services and intercepts. Certain types of service calls use approximately 100 bytes of ECSA while the service call is outstanding; however, CAISDI makes extensive use of cross-memory services to avoid the need for common storage areas. CAISDI/els uses varying amounts of ECSA depending on which products use this component. The CAISDI/els Interface Controller reports the ECSA usage separately for each product defined to it.
IBM System Modification Program Extended (SMP/E)	All CAISDI components are installed using IBM SMP/E protocols and services. Periodic CAISDI maintenance updates are delivered in SMP/E-ready format as well.
APF-authorized Load Library Data Set	CAISDI requires an APF-authorized load library data set. This data set must reside on a shared DASD volume if it is to be accessed by multiple z/OS systems.
Link List	All CAISDI components must reside in the system link list. This gives all CA product common access to the interface modules.
Parameter Data Set	The CAI.CAW0OPTN and CAI.CAW0OPTV data sets contain the CAISDI initialization statements for all CAISDI components. If these are to be shared by multiple z/OS systems, they must reside on a shared DASD volume.
Event Library	The CAI.CAIEVENT data set is the repository for all the CAISDI/els event definitions. The event members are furnished by the CA products that use CAISDI/els.
CA Common Services Components	CAISDI uses other CA Common Services, such as CAICCI. These components must be installed and properly configured.

Pre-installation Requirements

CAISDI requires that the following are installed and properly configured:

- TCP/IP on your z/OS system - For information on installing and configuring TCP/IP, see the documentation for the TCP/IP product you have.
- CA Service Desk on a server that can be accessed through the network from your z/OS system - The CA Service Desk is a CA Unix- or PC-based product that automatically streamlines identifying, tracking, and resolving end-user issues and problems

  For details on installation and configuration of the CA Service Desk, see .Configure Unicenter Service Desk (see page 38)
- CAICCI on your z/OS system - For information on installing and configuring the CAICCI service, see the CA Common Services for z/OS *Installation Guide*.

  The various CAISDI components use CAICCI to communicate with each other. Using CAICCI allows you to run the CAISDI/soap address space on one system and access that one address space from all other z/OS systems interconnected by CAICCI.
- The CAISDI components configured and installed on your z/OS system. See Configure CAISDI/elmds (see page 55).
- At least one CA product that uses the CAISDI interface. For details on configuring a particular CA mainframe-based product to work with CAISDI, see Configuring CA Products for CAISDI (see page 75) in this guide.

Configure CA Service Desk

This section describes how to configure CA Service Desk to work with CAISDI.

Loading Data Files

CA Service Desk is shipped with a number of product-specific data files. Each data file contains a collection of unloaded records that define or alter a product specific service desk template, a product specific pseudo contact name, and various other data required by CA Service Desk to support the integration with a specific CA product. These unloaded records must be loaded by the CA Service Desk administrator into the CA Service Desk database using the PDM_LOAD utility. This utility must be run from a command prompt on the system where the CA Service Desk is installed.

Once you open a command prompt on the CA Service Desk server, you must orient to the directory containing the data files. Begin by orienting on the CA Service Desk program directory. That directory is referred to generically as $NX_ROOT. The default $NX_ROOT is:

```
c:\Program Files\CA\Service Desk
```

It may be different in your environment. The data files are located in the directory $NX_ROOT\data\integrations, so if you used the default settings when you installed CA Service Desk, the data files are located in this directory:

```
c:\Program Files\CA\Service Desk\data\integrations
```

Orient to this directory and refer to the following table. For each CA product you plan to enable within CAISDI, you must run the PDM_LOAD utility specifying the associated CA Service Desk primary data file name. For example, if you want to load the data elements for CA 1 Tape Management, use the following command:

```
pdm_load -f integCA1.dat
```

This would load the CA 1 Tape Management ticket template, the CA 1 Tape Management pseudo contact definition, and other data required by CA Service Desk to support the integration with CA 1 Tape Management.

If the CA Service Desk server is configured to be ITIL compliant, you must run the PDM_LOAD utility again, specifying the associated ITIL update file name. This alters the definitions that were loaded from the primary file so they are also ITIL compliant. Using the CA 1 Tape Management example, you would issue this second command to apply the ITIL compliance updates:

```
pdm_load -f itil_integCA1.dat
```

Among other things, this would convert the request ticket template into an incident ticket for the ITIL environment.

The following table lists the data files for each product that integrates with CA NSM Service:

Product Name	Primary Data File	ITIL Update File
CA IDMS	integIDMS.dat	itil_integIDMS.dat
CA 1 Tape Management	integCA1.dat	itil_integCA1.dat
CA Allocate DASD Space and Placement	integAllocate.dat	itil_integAllocate.dat
CA Disk Backup and Restore	integDisk.dat	itil_integDisk.dat
CA TLMS Tape Management	integTLMS.dat	itil_integTLMS.dat
CA Vantage Storage Resource Manager	integVantage.dat	itil_integVantage.dat
CA Value Pack for DB2	integDB2Tools.dat	itil_integDB2Tools.dat
Generic MVS for CSS	integMVS.dat	itil_integMVS.dat
CA 7 Workload Automation	integCA7.dat	itil_integCA7.dat

Product Name	Primary Data File	ITIL Update File
CA Deliver	integDeliver.dat	itil_integDeliver.dat
CA Dispatch	integDispatch.dat	itil_integDispatch.dat
CA JARS Resource Accounting	integJARS.dat integJARSMVS.dat	itil_integJARS.dat itil_integJARSMVS.dat
CA Jobtrac Job Management	integJobtrac.dat	itil_integJobtrac.dat
CA MIM Resource Sharing	integMIM.dat	itil_integMIM.dat
CA OPS/MVS Event Management and Automation	integOPSMVS.dat	itil_integOPSMVS.dat
CA Spool Enterprise Print Management	integSpool.dat	itil_integSpool.dat
CA SYSVIEW Performance Management	integSysview.dat	itil_integSysview.dat
CA View	integView.dat	itil_integView.dat
CA Data Compressor for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Database analyzer for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Fast Check for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Fast Index for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Fast Load for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Fast Recover for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Fast Unload for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Index Expert for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
Ca Log Analyzer for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Merge/Modify for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA NetMaster Network Management for TCP/IP	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster Network Management for SNA	integNetMaster.dat	itil_integNetMaster.dat

Product Name	Primary Data File	ITIL Update File
CA NetMaster Network Automation	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster Network Operations for TCP/IP	integNetMaster.dat	itil_integNetMaster.dat
CA NetMaster File Transfer Management	integNetMaster.dat	itil_integNetMaster.dat
CA NetSpy Network Performance	integNetMaster.dat	itil_integNetMaster.dat
CA MICS Resource Management	integNeuMICS.dat	itil_integNeuMICS.dat
CA Partition Expert for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Plan Analyzer for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Quick Copy for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Rapid Reorg for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA RC/Extract for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA RC/Migrator for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA RC/Query for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA RC/Secure for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA RC/Update for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Recovery Analyzer for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat
CA Scheduler	integScheduler.dat	itil_integScheduler.dat
CA SQL Ease for DB2 for z/OS	integDB2Tools.dat	itil_integDB2Tools.dat

HTML Rendering

Some products are capable of generating CA Service Desk Request content in HTML format. (See the configuration requirements for each product supporting CAISDI in the following chapter.) CA Service Desk can render embedded HTML directives in the request Description and Summary fields. However, this capability is not activated when the CA Service Desk is initially installed. To render the HTML in the ticket properly, your CA Service Desk server must support HTML rendering.

Your local CA Service Desk administrator must perform this task. Discuss the impact in your environment of making this change to the CA Service Desk. The CA Service Desk reports that display the summary and description text do not render the HTML. In those reports, the HTML will appear in its raw source form. This may be undesirable in your enterprise.

To render HTML, the detail_cr.htmpl form must be customized. The administrator can use the CA Service Desk Web Screen Painter to select this form, modify the KEEPTAGS setting for the description and summary fields, and publish the form. Consider these important properties of the summary and description text fields:

KEEPLINKS	If set to “Yes”, render only HTML links (the anchor Action: tag) as hyperlinks. This is the default for the summary and description fields.
KEEPTAGS	If set to “Yes”, render all HTML tags. Consult with your CA Service Desk Administrator before implementing this feature.

Note: The KEEPLINKS property is ignored if KEEPTAGS is set to “Yes”.

For detailed instructions on setting these parameters, see the CA Service Desk r11 *Modification Guide*.

Configure CAISDI/soap

After installing CAISDI/soap using the procedures in the *Installation Guide*, follow this process to ensure you complete all the customization tasks for the CAISDI/soap component:

1. Customize and copy the CAISDI/soap JCL PROC (see page 43)
2. Copy the XML template data set (see page 45)
3. Obtain CA Service Desk URLs (see page 45)
4. Obtain a CA Service Desk Logon ID (see page 46)
5. Set up USS authorization (see page 46)
6. Set up SSL (see page 47)
7. Load the default CA Service Desk Template (see page 47)
8. Define the startup variables for CAISDI/soap (see page 48)

Step 1: Customize and Copy the CAISDI/soap JCL PROC

A sample started task JCL procedure for the Version 14.0 CAISDI/soap server, is located in member CASOAPE of the CAI.CAW0PROC data set:

```
//CASOAPE  PROC CAW0PLD='CAI.CAW0PLD',
//         SSLLOAD='SYS1.SIEALNKE',
//         SYSTCPD='TCPIP.DATA',
//         SOAPVARS='YOUR.ENVVAR.LIBRARY(SOAPVARS)'
//CASOAPE  EXEC PGM=CAS0SOAP,
// PARM='ENVAR("_CEE_ENVFILE=DD:ENVVAR")/'
//*
//STEPLIB  DD DISP=SHR,DSN=&CAW0PLD
//         DD DISP=SHR,DSN=&SSLLOAD
//*
//SYSTCPD  DD DISP=SHR,DSN=&SYSTCPD
//ENVVAR   DD DISP=SHR,DSN=&SOAPVARS
//LOGFILE  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//*
```

To customize and stage the CAISDI/soap server JCL PROC CASOAPE

1. Copy the CASOAPE member of CAI.CAW0PROC to any JCL PROCLIB that is automatically searched as part of z/OS START command processing.
2. Supply the name of the SSL load lib, if using SSL.
3. Customize the //ENVVAR DD statement by specifying the CAISDI/soap server environment variables file name. The CAI.CAW0OPTV library contains sample member SOAPVARS.

For the CAISDI/soap server before Version 14.0, a sample started task JCL procedure is contained in member CASOAP of the CAI.CAW0PROC data set:

```
//CASOAP   PROC CAIPLD='CAI.CAW0PLD',
//         SSLLOAD='GSK.SGSKLOAD',     SYSTEMS PRIOR TO Z/OS 1.6
//         SYSTCPD='TCPIP.DATA',
//         TEMPLATE='YOUR.TEMPLATE.LIBRARY',
//         CONFIG='YOUR.CONFIG.LIBRARY',
//         CONFIGM=SDCONFIG
//CASOAP   EXEC PGM=CASDSOAP
//*
//STEPLIB  DD DISP=SHR,DSN=&CAIPLD
//         DD DISP=SHR,DSN=&SSLLOAD
//*
//SYSTCPD  DD DISP=SHR,DSN=&SYSTCPD
//TEMPLATE DD DISP=SHR,DSN=&TEMPLATE
//LOGFILE  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//CONFIG   DD DISP=SHR,DSN=&CONFIG(&CONFIGM)
        //*
```

To customize and stage the CAISDI/soap server JCL PROC CASOAP

1. Copy the CASOAP member of CAI.CAW0PROC to any JCL PROCLIB that is automatically searched as part of z/OS START command processing.
2. Supply the name of the SSL load lib, if using SSL.
3. Customize the //TEMPLATE DD statement by specifying the CAISDI/soap server template library that is to contain the SDLOGIN and SDLOGOUT templates, which will be copied in the next step.
4. Customize the //CONFIG DD statement by specifying the CAISDI/soap server configuration file name. The CAI.CAW0OPTN library contains member SDCONFIG, which will be copied to a shared PARMLIB data set in a subsequent step.

Step 2: Copy XML Template Data Set (for proc CASOAP only)

Make sure the XML template members of CAI.CAW0OPTN, SDLOGIN and SDLOGOUT, are copied to a shared PARMLIB data set.

Step 3: Obtain CA Service Desk URLs

Contact your local CA Service Desk administrator to determine the URLs for the CA Service Desk server. This information will be used in configuring CAISDI/soap server in Step 7. You need two URLs:

- CAISDI access to create tickets and query results
- Web browser access so you can view the tickets that have been created

The CAISDI/soap server communicates with CA Service Desk through its Web Services. Unless the server running CA Service Desk has been configured otherwise, your URL must include a TCP/IP port specification of “8080”. For example:

http://your.local.com:8080/axis/services/USD_...

To view the tickets that have been opened, you would use a URL like the following in your web browser:

http://your.local.com:8080/CAisd/pdmweb.exe

If you are coding literal IP addresses in a URL in an IPv6 environment, do not code an IPv4 address as an IPv4-mapped IPv6 address as the code will create the IPv4-mapped IPv6 address if necessary.

If coding a literal IPv6 address in a URL, code the address as a literal ipv6-url as defined in RFC2732; that is, you should enclose the literal IPv6 address in square brackets and any optional port designation outside the brackets. For example:

```
[3ffe:2a00:100:7031::1]:8080
```

Where possible, specify host names rather than using literal IP addresses.

If you plan to use Secured Sockets (SSL), then tell your local CA Service Desk administrator that you need a URL for HTTPS communications. Unless the server running CA Service Desk has been configured otherwise, your URL must include a TCP/IP port specification or “8443”.

For example, if you are using HTTPS to access your CA Service Desk, you might use the following URL for CAISDI:

https://your.local.com:8443/axis/services/USD_...

To view the tickets that have been opened, you would use a URL like the following in your web browser:

https://your.local.com:8443/CAisd/pdmweb.exe

Step 3 is complete when you have the URL of the target CA Service Desk server.

Step 4: Obtain a CA Service Desk Logon ID

Contact your local CA Service Desk administrator and request a logon ID and password to be used by CAISDI. All communication to the CA Service Desk is performed via a session. The session must have a logon ID to begin. The logon ID must have sufficient authority to perform the following functions:

- Open CA Service Desk tickets
- Create new CA Service Desk contacts - CAISDI/els can dynamically create CA Service Desk Contacts in the CA Service Desk data base if they do not already exist. Many of the products that use this interface are prepackaged with “pseudo contacts.” They would have names like “System_CA1_User” so the tickets opened by this interface are readily recognizable.
- Create new assets - CAISDI/els can dynamically create CA Service Desk assets in the asset class of “Software” if they do not already exist. Many of the products that use this interface are prepackaged with software asset names. They would have names like “CA 1 Tape Management”.
- Create asset class - CAISDI/els can dynamically create the asset class named “Software” for the purpose of dynamically creating new assets.

Step 5: Set Up UNIX System Services Authorization

For the Version 14.0 CAISDI/soap server, basic socket programming is used. Environmental variables will be set by LE. For SSL, if the database is on UNIX files, then the CAISDI/soap server must have access to those files.

For the CAISDI/soap server before Version 14.0, the CAISDI/soap server address space must have specific access privileges granted by the System Authorization Facility, such as RACF, CA Top Secret, or CA ACF2. In order to access TCP/IP functions, it must have READ access to the BPX.SERVER resource, or it must run with UID=0.

Step 6: Set Up SSL

If you do not plan to use the Secured Socket Layer (SSL) for secured communications, then you may skip this step.

If you plan to use SSL for secured communications (HTTPS) between the mainframe CAISDI soap server and the CA Service Desk Web Server, you must obtain an SSL certificate and install it on the CA Service Desk Web Server. (Follow the appropriate procedures for your particular Web Server software.) An exported copy of the certificate must then be accessible to the CAISDI/soap server in either of two methods:

- Have the certificate stored in an SAF keyring file that is controlled by your security system (CA Top Secret, CA ACF2, or IBM's RACF). The certificate is given a name that can be used by CAISDI/soap server to retrieve the certificate. For information on installing a certificate in a SAF keyring, see the appropriate SAF documentation.
- Have the certificate stored in an HFS key database. If you choose this method, you will also need a "stash" file to contain the password to the key database. For information on installing a certificate in an HFS key database and stash file, see the IBM z/OS Manual *System Secure Sockets Layer Programming*.

The user ID associated with the CAISDI/soap server address space must have access to either the HFS files or the SAF keyring. The configuration file must be set up to identify either the HFS key database (and stash file) or the SAF keyring (Step 7 that follows).

Step 6 is complete when the SSL certificate is installed and HFS keyring database or a SAF keyring authorization correctly configured.

Step 7: Load the Default CA Service Desk Template

For products that directly call CAISDI/soap server and that do not manage their own CA Service Desk templates, you must load the predefined template named "CA-MVS,Low Priority." For more information, see Configuring CA Service Desk (see page 38).

Step 8: Define the Startup Variables for CAISDI/soap

For proc CASOAPE, you must define the Environmental Variables in the file allocated to ENVVAR. A sample member, SOAPVARS, is provided in data set CAI.CAW0OPTV. These are read and processed by LE. See the *z/OS XL C/C++ Programming Guide*. LE sets the Environmental Variables before the CAISDI/soap server starts. For description of the variables, see the CASOAPE Environmental Variables (see page 52).

For the CAISDI/soap server before Version 14.0 using proc CASOAP, you must define the startup variables for the CAISDI/soap server in a configuration file. A sample configuration file is provided in the SMP/E data set CAI.CAW0OPTN, member SDCONFIG, which you must customize and copy to a shared system PARMLIB.

The following rules apply to the parameter specification in the configuration file:

- The formats of the parameters are **parameter** and **parameter=value** as indicated in the table below.
- Columns 1 through 71 are examined for input.
- A “#” or “*” in column 1 designates that line is a comment.
- If a given value must be continued onto the next line, use a “\” character as the last character of data on that line. Columns 1 through 71 of the next line are logically appended to the value in the continued line at the point where the “\” is detected and that “\” character will be removed.
- If a parameter's value is to end in a “\” character, then code “\\” to signify a single “\” and no continuation.

The following table describes the CAISDI/soap server startup parameters when using proc CASOAP.

Important! Many of these fields are case sensitive. Set your text editor so that it keeps the input values in mixed case.

Parameter	Description

SD_URL	Sets the URL for SOAP requests to the Web Services directory of the CA Service Desk server. Set this to the URL gathered in Step 3. Specify SD_URL=*url* where *url* is the full URL. For example:
	`SD_URL=http://server.local.com:8080\`
	`/axis/services/USD_WebServiceSoap?wsdl`
	Note the continuation character (\) at the end of line one on this two-line input.
	When coding the SD_URL parameter in the SDCONFIG member of PARMLIB, specify a host name if possible. If coding a literal IP address in a URL in an IPv6 environment, do not code an IPv4 address as an IPv4-mapped IPv6 address. Code an IPv4 address as such and the code will create the IPv4-mapped IPv6 address.
	If coding a literal IPv6 address for a URL for an IPv6 environment, the address should be enclosed in square brackets with any optional port designations placed outside the brackets. For example:
	`[3FFE:2A00:100:7031::1]:8080`

SD_USER	The CA Service Desk User ID to be used in all CAISDI connections to the CA Service Desk. Set this parameter to the logon ID gathered in Step 4 above. Specify as SD_USER=*userid* where *userid* is the user ID.
SD_PASS	The password associated with SD_USER. Set this parameter to the logon password gathered in Step 4 above. Specify as *SD_PASS=password* where *password* is the password.

Parameter	Description
SSL_SAF	For SSL only. Specifies the keyring name from Step 6 above if your chose method 1. Specify as *SSL_SAF=ringname* where *ringname* is the name of the ring to which the certificates are connected. This parameter is mutually exclusive with SSH_HFS. If both SSL_SAF and SSL_HFS are specified, CAISDI/soap initialization will fail.
SSL_HFS	For SSL only. Specifies the HFS file name of the keyring file from Step 6 above if you chose method 2. Specify as SSL_HFS=*filename* where *filename* is the HFS file containing the certificates. This parameter is mutually exclusive with SSH_SAF. If both SSL_HFS and SSL_SAF are specified, CAISDI/soap initialization will fail.
SSL_STASH	For SSL only when SSL_HFS has been specified. Specifies the HFS file name of the password file for the keyring file from Step 6 above if you chose method 2. Specify as SSL_STASH=*filename* where *filename* is the HFS file containing the password to the keyring file specified in SSL_HFS.
TZ	Sets the time zone that is to be used for the message logging in the CAISDI/soap server address space. Specify this as *TZ=timezone* where *timezone* is a valid C++ time zone notation. See the IBM z/OS C/C++ Run-Time Library Reference for more details. Some examples are: EST5EDT for Eastern Time Zone, CST6CDT for Central Time Zone, MST7MDT for Mountain Time Zone, and PST8PDT for Pacific Time Zone.
WORK_DIR	The UNIX System Services working directory for writing “debug” statements by the script process. This is required only in DEBUG mode and only if you need the debugging output of the script process. In DEBUG mode without this variable set, you get detailed debugging information and packet tracing written to the LOGFILE DD. Specify as WORK_DIR=*hfsdir*, where *hfsdir* is the HFS working directory where the script debugging lines will be written. The filename specified at open is "./soap_script.log".

Parameter	Description
LOGIN	Sets the registration value associated with the CA Service Desk Login method. This field *must* contain the following value: http://www.ca.com/UnicenterServicePlus/Service Desk/Login Specify as LOGIN=*regvalue* where *regvalue* is the CA Service Desk registration value for the Login Web Services method.
LOGOUT	Sets the registration value associated with the CA Service Desk Logout method. This field *must* contain the following value: http://www.ca.com/UnicenterServicePlus/Service Desk/Logout Specify as LOGOUT=*regvalue* where *regvalue* is the CA Service Desk registration value for the Logout Web Services method.
CREATE	The registration value associated with the CA Service Desk CreateRequest method. This field *must* contain the following value: http://www.ca.com/UnicenterServicePlus/Service Desk/CreateRequest Specify as CREATE=*regvalue* where *regvalue* is the CA Service Desk registration value for the CreateRequest Web Services method.
DEBUG	Keyword used to enable diagnostic logging in CAISDI/soap. Specify as DEBUG.
NODEBUG	Keyword used to disable diagnostic logging in CAISDI/soap. Specify as NODEBUG.
TIMEOUT_CONNECT	Specifies the maximum time in seconds to wait for a connection to a web server. The default is 60 seconds. Specify a number between 1 and 3600. **Note:** Setting a value too low may prevent a connection from completing that would otherwise complete.

Parameter	Description
TIMEOUT_IO	Specifies the maximum time in seconds for a socket to wait for data from a Web Server or to be able to send it data. The default is 20 seconds. **Note:** Setting a value too low may cause the connection to close prematurely. Specify a number between 1 and 3600.
URL_symbol	A product may require a web server other than the default web server as specified by SD_URL. In this case the product provides the value for symbol. The value of URL_symbol is the URL that identifies the web server. For example, if a product associates symbol WS2 with an alternate web server, then define variable "URL_WS2" and assign the value the URL of the alternate web server.
USER_symbol	If URL_symbol is defined and the associated user ID is different from SD_USER, use USER_symbol to specify the associated user ID. For example, if the symbol is WS2, define variable "USER_WS2" and assign the value the associated user ID.
PASS_symbol	If URL_symbol is defined and the password of the associated user ID is different from SD_PASS, use PASS_symbol to specify the password. For example, if the symbol is WS2, define variable "PASS_WS2" and assign the value the password.

Note: For security purposes, the CAISDI/soap server configuration data set should be secured to prevent unauthorized access to the CA Service Desk user ID and password.

For a sample configuration file, see member SDCONFIG in the CAI.CAW0OPTN data set.

CASOAPE Environmental Variables

The following are the Environmental Variables for CASOAPE.

SD_URL

(Required) URL that identifies the Service Desk Service. Obtain the URL from the CA Service Desk administrator. See Step 3 (see page 45).

SD_WSDL_LEVEL

(Optional) Defines the WSDL level corresponding to the URL identified by SD_URL. The CAISDI/soap server normally determines the WSDL level from the URL, provided the absolute path in the URL is the standard CA Service Desk path.

Important! This should only be specified at the direction of CA support.

SD_USER

(Required) The user ID associated with the Service Desk identified in SD_URL. See Step 4 (see page 46).

SD_PASS

(Required) The password associated with the user identified by SD_USER.

URL_symbol

(Optional) A product may require a web server other than the default web server as specified by SD_URL. In this case the product provides the value for symbol. The value of URL_symbol is the URL that identifies the web server. For example, if a product associates symbol WS2 with an alternate web server, then define variable "URL_WS2" and assign the value the URL of the alternate web server.

WSDL_LEVEL_symbol

(Optional) Defines the WSDL level corresponding to the URL identified by URL_symbol. The CAISDI/soap server normally determines the WSDL level from the URL, provided the absolute path in the URL is the standard CA Service Desk path.

Important! This should only be specified at the direction of CA support.

USER_symbol

If URL_symbol is defined and the associated user ID is different from SD_USER, use USER_symbol to specify the associated user ID. For example, if the symbol is WS2, define variable "USER_WS2" and assign the value the associated user ID.

PASS_symbol

If URL_symbol is defined and the password of the associated user ID is different from SD_PASS, use PASS_symbol to specify the password. For example, if the symbol is WS2, define variable "PASS_WS2" and assign the value the password.

SSL_SAF

For SSL only. Specifies the keyring name from Step 6 if your chose method 1. Specify as SSL_SAF=ringname where ringname is the name of the ring to which the certificates are connected.

This parameter is mutually exclusive with SSH_HFS. If both SSL_SAF and SSL_HFS are specified, initialization will fail.

SSL_HFS

For SSL only. Specifies the HFS file name of the keyring file from Step 6 if you chose method 2. Specify as SSL_HFS=filename where filename is the HFS file containing the certificates.

This parameter is mutually exclusive with SSH_SAF. If both SSL_HFS and SSL_SAF are specified, initialization will fail.

SSL_STASH

For SSL only when SSL_HFS has been specified. Specifies the HFS file name of the password file for the keyring file from Step 6 if you chose method 2. Specify as SSL_STASH=filename where filename is the HFS file containing the password to the keyring file specified in SSL_HFS.

TZ

(Optional) Sets the time zone that is to be used for the message logging in the CAISDI/soap server address space. Specify this as TZ=timezone where timezone is a valid C/C++ time zone notation. See the IBM z/OS C/C++ Run-Time Library Reference for more details. Some examples are: EST5EDT for Eastern Time Zone; CST6CDT for Central Time Zone, MST7MDT for MountainTime Zone, and PST8PDT for Pacific Time Zone. This is not directly inspected by the CAISDI/soap server; rather it is used by the C/C++ functions.

TIMEOUT_CONNECT

(Optional) Specifies the maximum time in seconds to wait for a connection to a web server. The default is 60 seconds. Specify a number between 1 and 3600.

Note: Setting a value too low may prevent a connection from completing that would otherwise complete.

TIMEOUT_IO

Specifies the maximum time in seconds for a socket to wait for data from a Web Server or to be able to send it data. The default is 20 seconds. Specify a number between 1 and 3600.

Note: Setting a value too low may cause the connection to close prematurely.

DEBUG

N|Y. (Optional) The default is DEBUG=N. Specify DEBUG=Y for diagnostic information to be logged to the LOGFILE file.

TRACE

N|Y. (Optional) The default is TRACE=N. Specify TRACE=Y for internal trace information to be logged to the LOGFILE file. Only set at the direction of CA support.

CCI_DUMP_BUFFER

N|Y. (Optional) The default is CCI_DUMP_BUFFER=N. Specify CCI_DUMP_BUFFER=Y for CCI buffers to be dumped to the LOGFILE file. Only set at the direction of CA support.

TRACE_XML_PARSE

N|Y. (Optional) The default is TRACE_XML_PARSE=N. Specify TRACE_XML_PARSE=Y for trace information related to XML parsing to be logged to the LOGFILE file. Only set at the direction of CA support.

Note: For security purposes, the data set containing the Environmental Variables should be secured to prevent unauthorized access to any CA Service Desk user ID and password.

Configure CAISDI/elmds

When CAISDI/elmds is run, it initializes the environment which includes the med and els interfaces. After installing CAISDI/elmds using the procedures in the *Installation Guide*, follow this process to complete all customization tasks for both the CAISDI/els and CAISDI/med functions.

The CAISDI/elmds messages are distributed in CAW0OPTV member CDYFMSGS. The text of those messages can be changed (see specific instruction in the file). Each message has a WTO or NOWTO designation. The WTO indication causes messages to be issued as a WTO (see specific instructions in the file). If this file is changed, make a backup copy available as CAISDI/elmds does not start without a valid message file.

Note: Beginning with CA Commons Services for z/OS Release 14.1, CAISDI/elmds is the primary component replacing standalone CAISDI/med and standalone CAISDI/ els. If circumstances require you to run standalone CAISDI/els or standalone CAISDI/med components, see CAISDI/elmds Backout Procedure (see page 113).

1. Configure the CAISDI/elmds parameters (see page 55)
2. Define a Product Security User ID (see page 58)
3. Customize the CA Service Desk Parameters (see page 59)
4. Customize the CAISDI/med Product Parameters (see page 60)
5. Set Up CAISDI/els Startup Parameters (see page 62)
6. Configure the CAISDI/els Interface for Each CA Product (see page 65)
7. Consider the CAISDI/els Event Trigger Utility (see page 70)

Configure the CAISDI/elmds Parameters

The following CDYFCNFG statements can be specified, but none are required:

DUMPS ALL|SYSUDUMP|SVCDUMP ON|OFF

Indicates which dumps are taken. If not specified the default is DUMPS ALL ON. (Operator DUMPS command can change the setting.)

NO_ELS

Indicates do not initialize ELS support (comment out DDNAME ELSSTART).

NO_MED

Indicates do not initialize MED support (comment out DDNAME MEDPARMS).

Note: If NO_MED is not specified, copy the existing CAISDI/med MEDPARMS to a new library and allocate it to CAISDI/elmds.

SPIN nnnn

Controls whether the CDYFLOG SYSOUT data set can be spun or not. If nnnn is 0, the file is not automatically spun. If nnnn is a value from 1000 through 1000000 (no commas), the file is automatically spun after nnnn lines of output. The operator SPIN command can be used to reset the values or force the file to be spun. The operator SPINCLASS command can be used to reset the class for the next output file. You can specify SPIN 0 to allow use of the operator SPIN and SPINCLASS commands without an automatic spin.

If the statement is not present, then the output data set is not eligible to be spun and the operator SPIN and SPINCLASS commands are not available.

SYNTAX_CHECK_ONLY

Indicates CAISDI/elmds only validates syntax for the configuration files. After syntax validation, CAISDI/elmds terminates. With this parameter a CAISDI/elmds address space can be started, even if there is another instance active, to validate syntax changes.

TRACE ON|OFF

Indicates if tracing is active (ON) or inactive (OFF). (Operator TRACE command can change the setting.)

The following are operator commands using the MVS MODIFY command:

DUMPS ALL|SYSUDUMP|SVCDUMP ON|OFF

Sets which dumps are taken.

ELSDISPLAY

Displays the status of the ELS products as defined by the DEFINE statements in ELSSTART.

ELSDISABLE product|ALL

Dynamically disables new activity to Service Desk for the ELS product defined in the ELSSTART file as “PRODUCT product”. If “ALL” specified then all products are disabled. (Similar to the CAISDI/els DISABLE command.)

ELSENABLE product|ALL

Dynamically resumes activity for the ELS product defined in the ELSSTART file as "PRODUCT product". If "ALL" specified then all ELS products are resumed. (Similar to the CAISDI/els ENABLE command.)

MEDDISPLAY

Displays the status of the MED servers as defined by the "SERVER server" statements in MEDPARMS.

LOGREC DISPLAY|ENABLE|DISABLE

Displays the status of the LOGREC PC interface. Dynamically enable or disable the interface. If disabled, the LOGREC PC exits into the CAISDI/elmds address space but it returns immediately without processing any events.

Note: The messages are now written to CDYFLOG and not issued as a WTO in the application address space.

Note: There are new message IDs and text.

MEDDISABLE server|ALL

Dynamically disables new activity to Service Desk for the MED server defined in MEDPARMS as "SERVER server". If ALL is specified, then all MED servers are disabled. (Similar to the CAISDI/med STOP command.)

MEDENABLE server|ALL

Dynamically resumes activity for the MED server defined in MEDPARMS as "SERVER server". If "ALL" specified then all MED servers are resumed. (Similar to the CAISDI/med START command.)

QUERY

Displays the status of TRACE, DUMPS, SPIN, SPINCLASS and TIMEOUT.

SPIN nnnn|NOW

Controls SPIN action of the output. The SPIN configuration statement must first be specified in CDYFCNFG. If SPIN NOW is specified, the output data set is spun immediately and the output line count reset to 0. If nnnn is 0, the file is not automatically spun. If nnnn is a value from 1000 through 1000000 (no commas), the file is automatically spun after nnnn lines of output.

SPINCLASS c

Sets the class of the output data set when it is next allocated. Spin implies the file is closed, unallocated, and then a new SYSOUT data set allocated. The default is that it uses the class from the SYSOUT (which can be "*") in the JCL. If specified the "c"must be A-Z or 0-9. (Requires that the "SPIN" configuration statement was specified in CDYFCNFG.)

STOP

Indicates that CAISDI/elmds terminates. You can use either “F jobname, STOP” or “P jobname” to terminate the job.

SVCDUMP

Performs an immediate SVC dump regardless of the DUMPS setting.

TIMEOUT nnnn

Sets or cancels the global override timeout value in seconds used on each #SOAPSDI macro. A value of zero indicates any previous override cannot be canceled. If not zero, the valid range is 20 through 900 seconds.

Define a Product Security User ID

If your installation requires z/OS started tasks to have valid security system user IDs, contact the system security administrator at your installation to have a user ID defined for the CAISDI/elmds started task. If you are using the CASDIMED procedure, obtain a user ID for it also.

The CASDI/elmds user ID needs security access to the various configuration files allocated to CDYFCNFG, MEDPARMS and ELSSTART. If you are using a STEPLIB statement, the user ID also has to have access to the APF-authorized load library.

If you are using the CASDIMED procedure, the CAISDI/med user ID needs security access to the following:

- The CAISDI/med parameter data set referenced by the //MEDPARMS DD statement in the CAISDI/med JCL PROC
- The APF-authorized load library referenced by the //STEPLIB DD statement.

The local security product possibly protects data set resources such as these. CAISDI/med initialization fails unless the CAISDI/med started task has been granted proper access to the data sets it needs.

Customize the CA Service Desk Parameters

Follow these steps to customize CA Service Desk parameters:

1. So that CAISDI can successfully generate a ticket, CAISDI must define certain fields in a request. Because the CA Service Desk administrator typically builds and manages these field specifications, that person must obtain such information.
2. Before CA Service Desk can create tickets generated by CAISDI/elmds (supporting the CAISDI/med and CAISDI/els functionality), the CA Service Desk administrator must predefine the contacts referenced by the AFFECTED_END_USER, REPORTED_BY, and ASSIGNEE fields in a request. Consult the CA Service Desk administrator for appropriate contact names to define these fields.
3. To categorize and direct CA Service Desk tickets effectively, field data is sometimes required such as AFFECTED_END_USER, REPORTED_BY, and ASSIGNEE. In that case, the CA Service Desk administrator can predefine other fields in a CA Service Desk template. CAISDI/elmds can reference these fields when it generates a ticket using a template. Unique templates are predefined for each individual CA z/OS-based product that generates CA Service Desk tickets through CAISDI/elmds. The contacts and ticket templates must be replicated on every CA Service Desk server that receives tickets from CAISDI/elmds. The appropriate template name to use for each CA product is documented in the chapter “Configuring CA Products for the CAISDI” in this guide.
4. Learn the contact names as applicable that are specified for AFFECTED_END_USER, REPORTED_BY, and ASSIGNEE. If a CA Service Desk template is used for opening tickets, learn the name of the template as defined in the CA Service Desk server. Use this information in CAISDI/elmds PRODUCT initialization statements.

This step is complete when:

- Customized fields and templates have been defined in the CA Service Desk server for the CA products that generate tickets.
- The specified information has been obtained from the CA Service Desk administrator.

Customize the CAISDI/med Product Parameters

The CAISDI/med parameter member, MEDPARMS, is installed into CAI.CAW0OPTN during med or elmds installation. Copy this member into a system PARMLIB data set, so that customizations are not lost if future maintenance updates overwrite this member. CAISDI/med initialization statements found in the MEDPARMS member are read and processed during CAISDI/elmds address space initialization. Customize these statements so that CAISDI/med can operate in your unique environment.

You can customize the actual CAISDI/med parameter data set and member that is read during CAISDI/elmds address space initialization. Define them in the MEDINPUT symbolic for the //MEDPARMS DD statement found in the CAISDI/elmds (CDYFAPI) JCL PROC.

The initialization statements found in the MEDPARMS member, establish which z/OS-based CA products can communicate CA Service Desk tickets to the CA Service Desk product through CAISDI/med. Parameter specifications indicating the CA Service Desk server URL address, ticket attributes, z/OS-based product names, and others must be established using CAISDI/med initialization statements.

This section contains the following topics:

Configuration Examples

For a better understanding of general CAISDI/med statement and command syntax, see CAISDI/med Control Statements (see page 114). The following sections contain examples designed to illustrate a realistic configuration.

INIT Statement

The sample INIT statement defines CAISDI/med operational parameters. The INIT statement provides default values that identify the CAISDI/med address space and control the reusability of common storage in the event of a CAISDI/med restart.

```
INIT     REUSE=YES,        /* REUSE INTERCEPTS DURING RESTARTS      */
         MEDNAME=SDIA       /* UNIQUELY IDENTIFY THIS CAISDI/med   */
```

Important! You should not alter these values unless directed by CA Support.

Important! If CASDIMED must be run using the same MEDNAME as specified in the CDYFAPI MEDPARMS, then CDYFAPI must be started with either a different MEDNAME or with NO_MED in CDYFCNFG.

EXCLUDE Statement

The EXCLUDE statement is used to define events that CAISDI/med should ignore for the purpose of CA Service Desk ticket recording.

```
EXCLUDE  ABEND,                  /* Always exclude these ABENDs    */+
         CODES=((X'047'),        /* APF authorization              */+
                (X'13E'),(X'33E'),/* DETACH with and without STAE   */+
                (X'322'))        /* CPU time exceeded              */
```

In this example, CAISDI/med is directed to ignore abend events for the selected abend codes, regardless of any CA product-specific definitions.

SERVER Statement

The SERVER statement identifies the connection between the CAISDI/med address space and a named CA Service Desk server through an optionally named CAISDI/soap address space.

```
SERVER   USDSRV01,           /* Name for this server connection          */+
         HOSTNAME=host_name, /* SOAP Client Service CAICCI host name     */+
         TIMEOUT=30,         /* SOAP Client Service TIMEOUT in seconds   */+
         URL_SYMBOL=symbol   /* Service Desk server URL address          */
```

The SERVER name parameter is purely arbitrary; however, the installation should supply a meaningful name.

The HOSTNAME parameter should be used to supply the CAICCI name of the CAISDI/soap address space, if the CAISDI/soap is not running on the local z/OS image.

The TIMEOUT parameter controls the amount of time the CAISDI/soap server waits before returning a time-out condition to the CAISDI/med address space. The default value of 30 seconds should be adequate unless network or server traffic is particularly high between the CAISDI/elmds address space and the CA Service Desk server.

The optional URL_SYMBOL identifies a “symbol” that is passed to the CAISDI/soap server. The CAISDI/soap server will then determine the value of its Environmental Variable URL_symbol and use that as the URL of the CA Service Desk server. For example, if “URL_SYMBOL=WS2” then Environmental Variable “URL_WS2” must be defined to the CAISDI/soap server with a value which is the URL for the CA Service Desk server. See the CASOAPE Environmental Variables (see page 52).

Note: You need one SERVER statement for each CA Service Desk server with which your CAISDI/elmds address space should communicate.

PRODUCT Statement

The PRODUCT statement is used to establish initialization values that do the following:

- Define the representation of a CA product.
- Provide specific values for all CA Service Desk tickets generated on behalf of the defined CA product.

EVENT Statement

The EVENT statement is used to define to CAISDI/med the abend events for a given CA product that should be recorded as CA Service Desk tickets.

Interfacing with CA Mainframe Products

For details on configuring individual products to interface with CA Service Desk, see Configuring CA Products for CAISDI (see page 75) or the individual product documentation.

Product parameter configuration for CAISDI/med is complete when you have determined your site requirements and have modified the CAISDI/med initialization statements to meet those environmental factors.

Set Up CAISDI/els Startup Parameters

The CAISDI/els parameter library contains several members designed to control the CAISDI/els component. The ELSSTART, ELSSTOP, ELSREMOV, ELSLIST, and ELSREQS members are installed into CAI.CAW0OPTN. You may wish to copy these members to a shared system PARMLIB. The ELSSTART member contains the control statements necessary to establish the CAISDI/els interface. This member must be customized to your environment. Consider the contents of the ELSSTART member as it is distributed:

```
DEFAULTS USD_AUTODEF=yes          Dynamically add contacts, assets
```

The DEFAULTS control statement sets the default values for several DEFINE control parameters. The DEFINE control statements that follow will inherit the values from the DEFAULTS control statement. Any parameter specified on the DEFAULTS control statement may also be coded on a DEFINE control statement, but if you plan to specify the same parameter on all of your DEFINE commands, the DEFAULTS control statement allows you to code these parameters once, rather than repeating the specification on multiple DEFINE control statements.

In configuring any parameter in the ELSSTART member, if the parameter value is too long to fit on one line, it may be broken onto multiple lines. Using the "+" sign and continuation indicator. For example:

```
DEFAULTS  USD_ID='Service'+
          'Desk'
```

The trailing "+" immediately following the quote mark is the "continued string" indicator. Either single or double quotes may be used, but for a continuation, the continued string must begin with the delimiter character immediately preceding the "+" character. You may continue a string value on as many lines as necessary.

These control parameters may be specified in upper/lower/mixed case, but many of the **values** are case sensitive. Be sure the editor you are using does not force your control statements into upper case. The parameters to be configured are:

USD_AUTODEF

Indicates whether CAISDI/els should automatically define contact names and asset names. Each CA product that uses this interface ships its event definitions with pseudo contact names and asset names. For example, CA 1 Tape Management is packaged with a pseudo asset named "CA 1 Tape Management" and pseudo contact ID of "System_CA1_User". Associating these pseudo entities with the CA Service Desk tickets opened by CAISDI/els will simplify searching and tracking of those tickets. If USD_AUTODEF=YES has been specified, the contact IDs and asset names will be automatically defined in your CA Service Desk system if they are not already present.

If USD_AUTODEF=NO is specified, CAISDI/els cannot establish the interface for any product whose definitions include references to contacts or assets that do not already exist. The interface will not be enabled for those products.

Default: USD_AUTODEF=YES

URL_SYMBOL

Optional. Used to identify the CA Service Desk server on which you want the tickets opened. If omitted, the SD_URL value from the CAISDI/soap address space's Environmental Variables is used. You may select an alternate CA Service Desk server by assigning a “symbol” to URL_SYMBOL where URL_symbol is a defined Environmental Variable in the CAISDI/soap server and where its value is a URL identifying the CA Service Desk server. For example, if “URL_SYMBOL=WS2” then Environmental Variable “URL_WS2” must be defined to the CAISDI/soap server with a value which is the URL for the CA Service Desk server. See the CASOAPE Environmental Variables (see page 52).

USD_ID

Optional. Designates the CA Service Desk logon ID that will be used to access the CA Service Desk. If omitted and URL_SYMBOL is specified, then if USER_symbol is defined in the CAISDI/soap server’s Environmental Variables its value is used, otherwise the SD_USER value from the CAISDI/soap server’s Environmental Variables is used. If omitted and URL_SYMBOL is not specified, then the SD_USER value from the CAISDI/soap Environmental Variables is used. While this ID will be used to open CA Service Desk tickets, this ID will not appear on the tickets themselves. The contact identified by the REPORTED_BY parameter will be used for that purpose. Each product using this interface provides a product control member containing the REPORTED_BY specification. You will have the opportunity to alter the REPORTED_BY designations when the individual products are configured for this interface.

USD_PW

Optional. Designates the password to use with the aforementioned ID. If omitted and URL_SYMBOL is specified, then if PASS_symbol is defined in the CAISDI/soap server’s Environmental Variables its value is used, otherwise the SD_PASS value from the CAISDI/soap server’s Environmental Variables is used. If omitted and URL_SYMBOL is not specified, then the SD_PASS value from the CAISDI/soap Environmental Variables is used. The password value is suppressed in all reports. If you specify an actual ID and password here, you should take measures to secure the parameter library that contains the ELSSTART member.

This step is complete when you have updated the ELSSTART member with the CA Service Desk server URL, ID, and password.

Allocate the Event Library

During the installation of the CAI.CAIEVENT data set, the common event library for CAISDI/els is created. While some products may provide their own event libraries, a central event library makes customization easier.

This step is complete when CAI.CAIEVENT has been allocated.

Configure the CAISDI/els Interface for Each CA Product

Once you have updated the ELSSTART member with this information, you are ready to begin configuring the CAISDI/els interface for the CA products that will use it. The details for configuring specific CA products is covered in the next chapter “Configuring CA Products for the CAISDI” in this guide, but first consider the nature of the configuration task. For each CA product to be configured, you must:

- Add a DEFINE control statement to the ELSSTART member
- Optionally update that CA product's Product Control Member

As an alternative to keeping all DEFINE statements in the one ELSSTART member, you can set up one member product. Each product could then be started separately.

This section contains the following topics:

Adding the DEFINE Statement for Each CA Product

In the ELSSTART member, following the DEFAULTS control statement, you must include a DEFINE statement for each product that will be using this interface. The specific DEFINE parameter values to use are described in product-specific sections in Configuring CA Products for CAISDI (see page 75). These DEFINE control parameters and their values may be specified in upper/lower/mixed case. The values are forced to upper case internally. The parameters to be reviewed are:

PRODUCT

Identifies the CA product that will be using this interface. This value **must** be specified exactly as shown. When any given product accesses the CAISDI/els interface, it identifies itself internally with a fixed product code. This value must match the internal code for CAISDI/els to recognize the caller.

EVENTLIB

Specifies the library where this product's product control member and event members reside. The CAISDI/els event library structure allows several products to share the central event library (CAI.CAIEVENT) or each product may have its own event library. When each CA product is installed, an alternate event library name may be specified. For each product, specify the same event library name that was used when that product was installed.

PRODCNTL

Identifies the name of the product control member for a given product. This name **must** be specified exactly as shown. The product control member may optionally be customized for each product. It contains the CA Service Desk pseudo contact and asset names to be used in opening CA Service Desk tickets. The default pseudo contact and asset names are listed under each product in the chapter Configuring CA Products for CAISDI (see page 75) in this guide.

MASK

An optional parameter that specifies a generic range of members containing that product's event definitions. If a given product provides a MASK parameter, it **must** remain exactly as it is specified.

Updating Product Control Members

Each product using this interface provides a single product control member in the event library. You can use the product control members as they are or you may customize them to suit your environment. They are delivered with CA Service Desk pseudo contact IDs and pseudo asset names to be associated with any CA Service Desk ticket opened by this interface on that product's behalf. You may choose to change the contact IDs from the pseudo IDs to real IDs. With the exception of the PRODUCT control parameter, all parameter values are case sensitive. In any given product control member you will find a SETUP control statement that includes the following parameters:

PRODUCT

Required. This parameter identifies the CA product that will be using this interface. The value of PRODUCT **must not** be changed. It must match the PRODUCT code from the DEFINE command.

PRODNAME

Required. This parameter identifies the full product name. This name is used for reporting purposes only.

ASSIGNEE

Optional. This parameter identifies the CA Service Desk contact ID to be assigned to every ticket opened by this interface for this product. If you omit the ASSIGNEE parameter, the tickets may be unassigned. CA Service Desk provides several mechanisms to select an assignee. For example, if you have specified TEMPLATE, that template can set the assignee. Also, the CA Service Desk Assignee_set option can set so the REPORTED_BY contact is used when the ASSIGNEE field is not specified. Additionally, rules can be applied to set the assignee. See the CA Service Desk documentation for details on defining an assignee when one is not supplied. If specified, the contact whose ID is used must be of the type “Analyst”.

AFFECTED_END_USER

Required. This parameter identifies the CA Service Desk contact ID to be used as the affected end user for every ticket opened by this interface for this product.

REPORTED_BY

Required. This parameter identifies the CA Service Desk contact ID to be used as the individual who reports the problems of all the CA Service Desk tickets opened with this interface for this product. CA recommends you use the product-specific pseudo contact ID as it is distributed. That will provide a convenient way of querying the CA Service Desk for all tickets opened by this interface on behalf of each CA product.

ASSET

Optional. This parameter identifies a CA Service Desk asset (network resource) to be associated with the CA Service Desk tickets opened by this interface for this product. CA recommends you use the product-specific pseudo asset as it is distributed. That will provide a convenient way of querying the CA Service Desk for all tickets opened by this interface on behalf of any given CA product.

Note: If the CA Service Desk server is configured for ITIL compliance, the assets are labeled as “Configuration Items” while non-ITIL servers will label them as “Assets”.

TEMPLATE

Optional unless the CA Service Desk server is configured for ITIL compliance. The predefined template is *required* for ITIL configurations so that tickets open as *incidents*, not *requests*.

This parameter identifies a service desk template to be used in opening tickets in the CA Service Desk. The template can be used to set various fields in the ticket. If you specify TEMPLATE and the template is not available, the CAISDI/els interface for that product will **not** be enabled. Product-specific templates have been shipped with CA Service Desk, but they are not installed. The service desk administrator must install the templates using the PDM_LOAD utility. For details, see the section Loading Data Files (see page 38). We recommend omitting the TEMPLATE specification unless you are certain the template has been installed.

Certain fields such as ASSIGNEE may be set using the template, but if you also specify ASSIGNEE in the product control member, that will override the template. In other words, the template is the last place CA Service Desk will go for a data element.

LANG

Optional. This parameter identifies a two-character language code to be used in selecting Event Control Members that have been translated. See the section Notes on Translating Event Text (see page 69). The default is LANG=EN.

The specific DEFINE parameters to use and the default settings for the product control members are described in Configuring CA Products for CAISDI (see page 75). Note that some products use the CAISDI/els interface while others do not.

When you have completed this process, your ELSSTART parameter library member will look something like this:

```
DEFAULTS USD_AUTODEF=yes          Create contacts and assets
DEFINE PRODUCT=CA-1,EVENTLIB=caisdi.eventlib,PRODCNTL=al0cntl,MASK=l0*
DEFINE PRODUCT=allocate,EVENTLIB=caisdi.eventlib,PRODCNTL=xfgcntl,MASK=xf*
```

This step is complete when you have completed the updates for the ELSSTART member and have completed the customizing all of the individual CA product control members as outlined in Configuring CA Products for CAISDI (see page 75).

Updating Event Control Members

Each product using this interface provides one or more Event Control Members. You can use the Event Control members as they are or you may customize them for your environment. The member name is structured as a six-character event code and a two-character language code. For example, if the member name is L0Y001EN, the event code is L0Y001 and the language code is EN, meaning English.

There are three sections in an Event Control member. The summary, description, and options sections designated by the control words “SUMMARY:”, “DESCRIPTION:”, and “OPTIONS:” in column one, respectively. The sections must appear in this order.

The summary and description sections contain the text to be used in the CA Service Desk ticket. This text usually contains imbedded symbolic parameters that are resolved when the event is triggered.

Note: For CAISDI/els only, the text of the Event Control Members will appear to be fragmented. Please **do not** “clean up” the text. It has been structured this way to minimize the amount of ECSA required to store it. If the same text fragment appears more than once across all of the events for a given product, the text fragment is only stored once and all events reference that one text fragment.

The last section, the options section, contains a SET command with one or more parameters specified. You may want to customize these values to suit your environment. These are the valid SET parameters:

MAXCOUNT

Specifies the maximum number of occurrences allowed for this particular event expressed as a value from 1 to 999999. Each occurrence of this event causes the event's counter to be incremented. Once the limit is reached, no more tickets will be opened for this event. If MAXCOUNT is not specified, then no maximum occurrence limit is set. Unless MINTIME prevents an event from opening a ticket, each time the event is triggered a CA Service Desk ticket is opened.

MINTIME

Specifies the minimum amount of time that must have elapsed since the last occurrence of this event that caused a ticket to be opened. It is expressed in the format of HH:MM. This is a throttle control to prevent flooding the CA Service Desk with redundant tickets for system wide failures that may be recognized by many tasks at about the same time. If MINTIME is not specified, then there is no minimum time restriction. Unless MAXCOUNT prevents an event from opening a ticket, each time the event is triggered a CA Service Desk ticket is opened.

PRIORITY

Sets the Priority field in the CA Service Desk ticket. The possible values are 1, 2, 3, 4, 5, and NONE. The default is 2.

The event counters and timestamps for each event are maintained by CAISDI/els until either a SHUTDOWN or REMOVE command removes the product definition from the interface. Those values are then discarded. When a subsequent DEFINE command reestablishes the interface, the event counters and timestamps are re-initialized.

Notes on Translating Event Text

You may want to translate the text of some or all events into another language. Using our earlier example, if you want to translate L0Y001EN into another language, simply create a new member name that begins L0Y001 and use one of the following language codes:

- DA (Danish)
- DE (German)
- ES (Spanish)
- FI (Finnish)
- FR (French)
- IT (Italian)
- NO (Norwegian)
- PT (Portuguese)
- SV (Swedish)

If, for example, you translated L0Y001EN into German, you would create the member named L0Y001DE. Only the contents of the summary and description sections should be translated.

To select an alternate language for processing, you have two options:

- Include the LANG parameter on the DEFINE command for the product in the ELSSTART CAI.CAW0OPTN member. (You can code the LANG parameter in the DEFAULTS command if you want it to apply to all products.)
- Include the LANG parameter on the SETUP command in the Product Control Member. The LANG setting in the Product Control member overrides the LANG setting from the DEFINE command.

For this example, you would include LANG=DE on the control statement.

As mentioned previously, the text should be structured in text fragments to minimize the amount of ECSA required to store it. If the same clause appears more than once across all of the events for a given product, that clause should always be specified on a line by itself. This causes that clause to be stored once and all events will reference that one clause. If you translate the text, you must be mindful of the ECSA storage that is occupied while the CAISDI/els interface is active.

If you translate some, but not all events, CASIDI/els will substitute the English version for those you did not translate.

Consider the CAISDI/els Event Trigger Utility

Some CA products call CAISDI/els when a specific event occurs or condition arises. A call to CAISDI/els is referred to as triggering an event. The Event Trigger Utility can be used to trigger events manually.

For CAISDI/elmds, the Event Trigger Utility is CDYFELS1. For installations that use CASDIELS (standalone CAISDI/els), the utility is CSDETRIG. The same JCL used with CASDIELS can be used with CAISDI/elmds, substituting CDYFELS1 for CSDETRIG. For CAISDI/elmds, the JCL parsing rules for continuation are the same as for the ELSSTART configuration file.

The Event Trigger Utility is useful in verifying network connections and verifying that all components are connecting properly. You can invoke this utility from within production job streams to open CA Service Desk tickets based on condition codes. You can create your own events to augment the ones supplied by CA products. The only way you can trigger your own events is to use the Event Trigger Utility.

Sample Event Trigger Utility JCL:

```
//TRIG   JOB  (account)'Systems',CLASS=A
//*
//EVENT   EXEC PGM=CDYFELS1,PARM='Test message text'
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSIN    DD  *
TRIGGER PRODUCT=CA-1,EVENT=L0E231,JOB=MYJOB,DSN=TEST.DATA.SET.NAME,MSG="&PARM"
//SYSPRINT DD  SYSOUT=A
```

The Event Trigger Utility has two control parameters:

PRODUCT=

The product code of the CA product whose event you are triggering.

EVENT=

The six character event code you are triggering. All event member names are eight characters. A six character code followed by a two character language code. This parameter identifies the code, not the member name.

Any other parameters seen by the Event Trigger Utility are considered to be symbolic parameters to provide detailed event data. In the sample JCL, the JOB, DSN, and MSG parameters are set up as symbolic parameters. The values you assign &JOB, &DSN, and &MSG, replace them in the summary or description text of the L0E231 event.

Any value you specify with PARM= in the EXEC statement can be passed directly into the control statements or comments. The &PARM symbolic is replaced before the control statement is parsed, with the value you specified in PARM= of the EXEC statement. In this example, the MSG symbolic parameter is assigned the value "Test message text". This text replaces each occurrence of &MSG in the L0E231 event text. The parameter data passed through PARM= cannot exceed 120 characters.

Using the Event Trigger Utility, you can trigger events of your own making. Construct a product control member and a set of product event members for your own purposes. Execute the Event Trigger Utility as a job step whenever you want to open a CA Service Desk ticket.

Run CAISDI

This section describes how to run CAISDI. For information on using CAISDI with your CA mainframe products, see the documentation for that product.

Run CAISDI/soap

This section describes how to start and run the CAISDI/soap component of CAISDI.

This section contains the following topics:

Starting the CASDI/soap Address Space

To start the CAISDI/soap address space, enter the operating system START command. For example, issue the following command to start a copy of the CAISDI/soap address space using all start parameter defaults:

```
START CASOAPE
```

The following message indicates the CAISDI/soap address space completed initialization and is active:

```
CASD080I CASOAPE CAISDI/soap Client ready to accept commands
```

Stopping the CAISDI/soap Address Space

To stop the CAISDI/soap address space, enter the operating system STOP command. For example, issue the following command to stop the copy of the CAISDI/soap address space that was started using the preceding example:

```
STOP CASOAPE
```

CAISDI/soap Commands

CAISDI/soap operator commands can be entered via the MVS MODIFY command using the “APPL=” format.

DEBUG Command

Use the DEBUG command to set debug logging.

This command has the following format:

```
MODIFY CASOAP,APPL=DEBUG=Y|N
```

STOP Command

Use the CAISDI/soap STOP command as an alternate method to stop CAISDI/soap.

This command has the following format:

```
MODIFY CASOAP,APPL=STOP
```

TIMEOUT_IO Command

The TIMEOUT_IO command specifies the timeout value, in seconds, for socket sends and receives and specifies the maximum time to wait for data from the web server or to wait to be able to send data to the web server.

Note: Setting a value too low may close a working connection prematurely. A value of zero resets the timeout to the default value of 20 seconds. The maximum value is 3600.

This command has the following format:

```
MODIFY CASOAPE,APPL=TIMEOUT_IO=nnnn
```

nnnn

Indicates the timeout value in seconds.

TIMEOUT_CONNECT Command

Use the TIMEOUT_CONNECT command to set the timeout value, in seconds, for a socket connection to a Web Server. It will specify the maximum time to wait for the connection to complete.

Note: Setting a value too low may prevent a connection from completing that would otherwise complete. The maximum value is 3600. A value of zero resets the timeout to the default of 60 seconds.

This command has the following format:

```
MODIFY CASOAPE,APPL=TIMEOUT_CONNECT=nnnn
```

nnnn

Indicates the timeout value in seconds.

CAISDI/soap Messages

CAISDI/soap messages, suggested actions, and return codes are described in the CA Common Services for z/OS *Message Reference Guide*.

Run CAISDI/elmds

This section describes how to start and run the CAISDI/elmds component of CAISDI.

CAISDI/elmds Commands

This section describes CAISDI/elmds commands.

Starting the CAISDI/elmds Address Space

To start the CAISDI/elmds address space, enter the operating system START command. For example, issue the following command to start a copy of the CAISDI/elmds address space using all start parameter defaults:

```
START CDYFAPI
```

The following message indicates that the CAISDI/elmds address space completed initialization and is active:

```
CDYF001I CAISDI/elmds Initialized
```

Also, verify that the job output contains the following messages:

```
CDYF401I ELS initialized and available
CDYF400I MED initialized and available
CDYF001I CAISDI/elmds Initialized
```

Stopping the CAISDI/elmds Address Space

To stop the CAISDI/elmds address space, enter the operating system STOP or MODIFY command. For example, issue one of the following commands to stop the copy of the CAISDI/elmds address space that was started using the preceding example:

```
STOP CDYFAPI
```

or

```
MODIFY CDYFAPI,STOP
```

CAISDI/elmds Messages

CAISDI/elmds messages are described in the CA Common Services for z/OS *Message Reference Guide.*

Chapter 3: Configuring CA Products for CAISDI

This chapter provides information about all of the CA z/OS mainframe program products that support CAISDI.

Introduction

The following products currently support CAISDI:

- CA IDMS
- CA 1 Tape Management
- CA Allocate DASD Space and Placement
- CA Disk Backup and Restore
- CA TLMS Tape Management
- CA Vantage Storage Resource Manager
- CA Value Pack for DB2
- CA 7 Workload Automation
- CA Deliver
- CA Dispatch
- CA JARS Resource Accounting
- CA Jobtrac Job Management
- CA MIM Resource Sharing
- CA OPS/MVS Event Management and Automation
- CA Spool Enterprise Print Management
- CA SYSVIEW Performance Management
- CA View
- CA Data Compressor for DB2 for z/OS
- CA Database analyzer for DB2 for z/OS
- CA Fast Index for DB2 for z/OS
- CA Fast Load for DB2 for z/OS
- CA Fast Recover for DB2 for z/OS

- CA Fast Unload for DB2 for z/OS
- CA Index Expert for DB2 for z/OS
- Ca Log Analyzer for DB2 for z/OS
- CA Merge/Modify for DB2 for z/OS
- CA NetMaster Network Management for TCP/IP
- CA NetMaster Network Management for SNA
- CA NetMaster Network Automation
- CA NetSpy Network Performance
- CA NetMaster Network Operations for TCP/IP
- CA NetMaster File Transfer Management
- CA MICS Resource Management
- CA Partition Expert for DB2 for z/OS
- CA Plan Analyzer for DB2 for z/OS
- CA Quick Copy for DB2 for z/OS
- CA Rapid Reorg for DB2 for z/OS
- CA RC/Extract for DB2 for z/OS
- CA RC/Migrator for DB2 for z/OS
- CA RC/Query for DB2 for z/OS
- CA RC/Secure for DB2 for z/OS
- CA RC/Update for DB2 for z/OS
- CA Recovery Analyzer for DB2 for z/OS
- CA SQL Ease for DB2 for z/OS

This chapter contains a separate section for each CA product using the CAISDI. To implement CAISDI with a particular product, you may also need to refer to the recommended product-specific documentation. The tasks you must perform to enable CAISDI differ from product to product.

Note: CA Service Desk servers configured for ITIL compliance use slightly different terminology than servers that are not. In ITIL configurations, what are referred to in this guide as *tickets* are actually *incident tickets*; in non-ITIL configurations they are *request tickets*. CA Service Desk displays Incident and Request queues for ITIL configurations, while for non-ITIL configurations, only the Request queue displays. Also, in ITIL configurations, the term “Asset” is replaced with the term “Configuration Item.” For more information about the ITIL interface, see the CA Service Desk documentation, specifically the *ITIL User Guide*.

Note: Do not begin any product-specific implementation tasks until you have installed and configured the CAISDI/soap component and, depending upon the requirements of the CA products you are integrating with CA Service Desk and CAISDI/elmds components. The installation procedure for these components is described in the CA Common Services for z/OS *Installation Guide* and the section Configuration in this guide. The CAISDI/soap component is required by all products, and needs to be implemented only once; it can then be used by all products.

We recommend that you first install and implement CAISDI/soap. Then install CAISDI/elmds and configure as required by the CA products you are integrating (see the table in the section Required Components by CA Product (see page 31)) and initially implement CAISDI with one mainframe product. Once this is successful and you are familiar with the operation of CAISDI/soap, you can then integrate additional mainframe products.

CA Products

This section describes how to configure the CA products to use the CAISDI. The CA products all use the CAISDI/els interface.

This section contains the following topics:

CA IDMS

The following CAISDI/med PRODUCT and EVENT initialization statements define the interaction between CAISDI and CA IDMS to generate CA Service Desk tickets for CA IDMS Central Version abends:

```
/*===============================================================*/
/* Advantage  CA-IDMS Product Interface Definitions              */
/*                                                               */
/* Define Advantage CA-IDMS product ABEND events that are        */
/* to be captured by CAISDI/med, and how that information        */
/* is to be presented in the ticket to CA Service Desk.      */
/*===============================================================*/
/*                                                               */
/*   Required customization:                                     */
/*                                                               */
/*      SERVER  CAISDI/med name for Service Desk Server          */
/*                                                               */
/*                                                               */
/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,    */
/*          ASSIGNEE, and TEMPLATE are valid, provided the       */
/*          CA Service Desk Administrator loads predefined   */
/*          templates and contacts.  Otherwise, customize these  */
/*          parameters as directed by the Unicenter Service      */
/*          Desk Administrator.                                  */
/*                                                               */
/*===============================================================*/
PRODUCT CA-IDMS,                  /* Define product name         */+
        EVENT=ABEND,              /* Event qualifier             */+
        AFFECTED_END_USER=System_IDMS_User, /* Affected End User */+
        REPORTED_BY=System_IDMS_User, /* Reported By for tickets */+
        ASSIGNEE=System_IDMS_User,      /* Assigned contact name */+
        PRIORITY=2,               /* Priority for tickets        */+
     /* TEMPLATE='CA-IDMS,Low Priority',/*  Service Desk template */+
        SERVER=USDSRV01           /* Direct request to this server  */

EVENT ABEND,                    /* Capture ABEND events          */+
      COMPONENT='IDMS*',        /* Advantage CA-IDMS Component ID */+
      PRODUCT=CA-IDMS           /* Associated product statement  */
```

The preceding example shows how the PRODUCT and the EVENT statement are used to activate the capture of a named product's abend events and to define the associated CA Service Desk attributes to be associated with tickets that are opened as a result of the abend event.

Beginning with the EVENT statement, the crucial specification provided is the COMPONENT operand, IDMS. CA IDMS recovery routines always place the character string, IDMS, in the SDWACID field for all CA IDMS abend events. Therefore, this COMPONENT specification indicates that CAISDI/med abend event intercept should capture any detected abend event where the concatenation of the SDWACID and SDWASC fields begins with the characters IDMS. The asterisk (*) in the sample initialization statement is a wild card and indicates that any subsequent characters are considered a match condition.

The PRODUCT parameter operand, CA-IDMS, defines the CAISDI/med product name to associate with abend events captured due to a matching COMPONENT specification. It is this PRODUCT name specification that makes the connection to a specific CAISDI/med PRODUCT initialization statement.

The preceding PRODUCT statement defines the CA-IDMS product as referenced in the PRODUCT= parameter of the EVENT statement. The PRODUCT statement EVENT parameter qualifies that the product events associated with this product definition are abend events.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters define the names of predefined CA Service Desk specific entities that are to be assigned to any generated tickets. These named entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensures that tickets, generated by CAISDI/med on behalf of the named product, are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the Priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

For detailed information on these parameters, see Running CAISDI (see page 71).

CA 1 Tape Management

CA 1 Tape Management installs its event definitions into CAI.CAIEVENT. The product control member name is AL0CNTL, so include the following additional control statement in the CAISDI/els ELSSTART control member:

```
DEFINE PRODUCT=CA-1,EVENTLIB=cai.caievent,PRODCNTL=al0cntl,MASK=l0y*
```

The recommended values for the SETUP parameters are:

Parameter	Recommended Value
PRODUCT	CA 1
PRODNAME	CA 1 Tape Management
ASSIGNEE	System_CA1_User
AFFECTED_END_USER	System_CA1_User
REPORTED_BY	System_CA1_User
ASSET	CA 1
TEMPLATE	CA 1,Low Priority

CA Allocate DASD Space and Placement

The CA Allocate DASD Space and Placement product installs its event definitions into CAI.CAIEVENT. The product control member name is XGFCNTL, so include the following additional control statement in the CAISDI/els ELSSTART control member:

```
DEFINE PRODUCT=ALLOCATE,EVENTLIB=cai.caievent,PRODCNTL=xgfcntl,MASK=gf*
```

The recommended values for the SETUP parameters are:

Parameter	Recommended Value
PRODUCT	ALLOCATE
PRODNAME	CA Allocate DASD Space and Placement
ASSIGNEE	System_Allocate_User
AFFECTED_END_USER	System_Allocate_User
REPORTED_BY	System_Allocate_User
ASSET	CA Allocate DASD Space and Placement
TEMPLATE	CA-Allocate,Low Priority

CA Disk Backup and Restore

The CA Disk Backup and Restore product installs its event definitions into CAI.CAIEVENT. The product control member name is XGNCNTL, so include the following additional control statement in the CAISDI/els ELSSTART control member:

```
DEFINE PRODUCT=CA-Disk,EVENTLIB=cai.caievent,PRODCNTL=xgncntl,MASK=gn*
```

The recommended values for the SETUP parameters are:

Parameter	Recommended Value
PRODUCT	CA-DISK
CA Disk Backup and RestoreASSIGNEE	System_Disk_User
AFFECTED_END_USER	System_Disk_User
REPORTED_BY	System_Disk_User
ASSET	CA Disk Backup and Restore
TEMPLATE	CA-Disk,Low Priority

CA TLMS Tape Management

The CA TLMS Tape Management product installs its event definitions into CAI.CAIEVENT. The product control member name is ATLCNTL, so include the following additional control statement in the CAISDI/els ELSSTART control member:

```
DEFINE PRODUCT=TLMS,EVENTLIB=cai.caievent,PRODCNTL=atlcntl,MASK=tlv*
```

The recommended values for the SETUP parameters are:

Parameter	Recommended Value
PRODUCT	TLMS
PRODNAME	CA TLMS Tape Management
ASSIGNEE	System_TLMS_User
AFFECTED_END_USER	System_TLMS_User
REPORTED_BY	System_TLMS_User
ASSET	CA TLMS Tape Management
TEMPLATE	CA-TLMS,Low Priority

CA Vantage Storage Resource Manager

The CA Vantage Storage Resource Manager product installs its event definitions into its own CAIEVENT library. These members may be copied to the central CAI.CAIEVENT library or the CA Vantage Storage Resource Manager CAIEVENT library may be referenced in the ELSSTART member. The product control member name is XVPCNTL, so include the following additional control statement in the CAISDI/els ELSSTART control member:

```
DEFINE PRODUCT=VANTAGE,EVENTLIB=cai.caievent,PRODCNTL=xvpcntl,MASK=vp*
```

The recommended values for the SETUP parameters are:

Parameter	Recommended Value
PRODUCT	VANTAGE
PRODNAME	CA Vantage Storage Resource Manager
ASSIGNEE	System_Vantage_User
AFFECTED_END_USER	System_Vantage_User
REPORTED_BY	System_Vantage_User
ASSET	CA Vantage Storage Resource Manager
TEMPLATE	CA-Vantage,Low Priority

CA Products That Support CA Service Desk Integration

The following sections describe the CA products that support CA Service Desk Integration.

This section contains the following topics:

CA 7 Workload Automation

This product uses the CAISDI/els interface. CA 7 Workload Automation installs its event definitions in CAI.CA7.CAIEVENT. Do not copy these members into the CAI.CAIEVENT library. The product control member name is AL2CNTL, so include the following additional control statement in the CAISDI/els ELSSTART control member:

```
DEFINE PRODUCT=CA-7,EVENTLIB=cai.ca7.caievent,PRODCNTL=al2cntl
```

The recommended values for the SETUP parameters are:

Parameter	Recommended Value
PRODUCT	CA-7
PRODNAME	CA 7 Workload Automation
ASSIGNEE	System_CA7_User
AFFECTED_END_USER	System_CA7_User
REPORTED_BY	System_CA7_User
ASSET	CA 7 Workload Automation
TEMPLATE	CA-7,MEDIUM-LOW Priority

Tickets may be created based on events in the CA 7 Workload Automation address space. The interface is activated by adding a SERVICEDESK initialization file statement and a SERVDESK DD control file in the CA 7 Workload Automation address space. Refer to the CA 7 Workload Automation *Interfaces Guide* for a description of the SERVDESK rules used to determine what events will create tickets and what EVENTLIB members to use.

CA Deliver

The following CAISDI/med PRODUCT and EVENT initialization statements define the interaction between CAISDI and CA Deliver to generate CA Service Desk tickets for CA Deliver abends:

```
/*===============================================================*/
/* Unicenter CA-Deliver Product Interface Definitions            */
/*                                                               */
/* Define Unicenter CA-Deliver product ABEND events that are to be  */
/* captured by CAISDI/med, and how that information is to be     */
/* presented in the ticket to CA Service Desk.              */

/*                                                               */
/*   Required customization:                                     */
/*                                                               */
/*      SERVER  CAISDI/med name for Service Desk Server          */
/*                                                               */
/*                                                               */

/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,    */
/*          ASSIGNEE, and TEMPLATE are valid, provided the       */
/*          CA Service Desk Administrator loads predefined   */
/*          templates and contacts.  Otherwise, customize these  */
/*          parameters as directed by the Unicenter Service      */
/*          Desk Administrator.                                  */
/*                                                               */
/*===============================================================*/

PRODUCT CA-DELIVER,                    /* Product name            */+
        EVENT=ABEND,                   /* Event qualifier         */+
        JOBNAME=RMO*,                  /* Product JOB names       */+
        AFFECTED_END_USER=System_Deliver_User, /* Affected End User */+
        REPORTED_BY=System_Deliver_User, /* Reported by for tickets */+
        ASSIGNEE=System_Deliver_User,  /* Assigned contact name   */+
        PRIORITY=3,                    /* Priority for tickets    */+
     /* TEMPLATE= CA-Deliver,Priority unassigned', /* template name */+
        SERVER=USDSRV01           /* Direct request to this server  */


EVENT ABEND,                           /* Capture ABEND events    */+
      PRODUCT=CA-DELIVER,              /* Associated Product      */+
      COMPONENT='DLVR RMO*',        /*Unicenter CA-Deliver component*/+
      JOBNAME=RMO*,                    /* Product JOB names       */+
      EXCLUDE=((0522,),(X'522',))      /* Ignore session time-outs  */
```

This example shows how the PRODUCT and EVENT statements are used to activate the capture of abend events for CA Deliver. These statements also define the CA Service Desk attributes to be associated with the tickets that are opened as a result of the abend event.

The PRODUCT statement defines the product, in this case CA Deliver.

The JOBNAME parameter defines the job names to which this product statement applies. In the preceding sample, all jobs beginning with “RMO*” would qualify for the processes defined for this product statement.

The EVENT parameter indicates that the events associated with this PRODUCT definition are abend events.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters identify CA Service Desk entities that are assigned to any generated tickets. These entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensure that tickets generated by CAISDI/med on behalf of CA Deliver are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

The EVENT statement contains the crucial COMPONENT operand that is used to match this event statement with the actual event. For CA Deliver, this is set to a value of “DLVR RMO*”. This operand must be entered exactly as it is coded in this example. *Note that the space between DLVR and RMO is* significant.

CA Deliver recovery routines will place the character string “DLVR RMO” in the SDWACID field for all CA Deliver abend events. This COMPONENT specification indicates that CAISDI/med abend event intercept should capture any detected abend event where the concatenation of the SDWACID and SDWASC fields begins with the characters “DLVR RMO”. The asterisk (*) in the sample initialization statement is a wild card and indicates that any subsequent characters are considered a match condition.

The EXCLUDE parameter of the EVENT statement lists abend/reason code combinations for CA Deliver that should be ignored for problem reporting. In the sample the S522 abends are eliminated because they are normal in CA Deliver for XMS session time-outs.

The JOBNAME parameter of the EVENT statement defines the jobs where the abend event may occur. In the sample, RMO* is coded so that all intercepted abends are processed by this EVENT statement in CA Deliver jobs such as RMOSTC and RMODBASE. The JOBNAME parameter can be used to restrict the event to a specific job, or masked to handle a range of jobs. The JOBNAME parameter is also useful if multiple copies of CA Deliver are operating in a single z/OS operating system image. In this situation separate JOBNAME parameters can direct and categorize CA Service Desk tickets from each CA Deliver address space.

The PRODUCT parameter on the EVENT statement defines the CAISDI/med product associated with the abend events captured. It is this PRODUCT name specification that makes the connection back to the CAISDI/med PRODUCT initialization statement.

For detailed information on these parameters, see Running CAISDI (see page 71).

CA Dispatch

The following CAISDI/med PRODUCT and EVENT initialization statements define the interaction between CAISDI and CA Dispatch to generate CA Service Desk tickets for CA Dispatch abends:

```
/*===================================================================*/
/* Unicenter CA-Dispatch Product Interface Definitions               */
/*                                                                   */
/* Define Unicenter CA-Dispatch product ABEND events that are to be */
/* captured by CAISDI/med, and how that information is to be        */
/* presented in the ticket to CA Service Desk.               */

/*                                                                   */
/*   Required customization:                                         */
/*                                                                   */
/*      SERVER  CAISDI/med name for Service Desk Server              */
/*                                                                   */
/*                                                                   */

/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,        */
/*          ASSIGNEE, and TEMPLATE are valid, provided the           */
/*          CA Service Desk Administrator loads predefined   */
/*          templates and contacts.  Otherwise, customize these      */
/*          parameters as directed by the Unicenter Service          */
/*          Desk Administrator.                                      */
/*                                                                   */
/*===================================================================*/
PRODUCT CA-DISPATCH,                    /* Product name             */+
        EVENT=ABEND,                    /* Event qualifier          */+
        JOBNAME=*,                      /* Product JOB names        */+
        AFFECTED_END_USER=System_Dispatch_User, /* Affected End User*/+
        REPORTED_BY=System_Dispatch_User, /* Reported by            */+
        ASSIGNEE=System_Dispatch_User,  /* Assigned contact name    */+
        PRIORITY=3,                     /* Priority for tickets     */+
     /* TEMPLATE='Dispatch,Priority unassigned', /* template name   */+
        SERVER=USDSRV01         /* Direct requests to this server    */
EVENT ABEND,                            /* Capture ABEND events     */+
      PRODUCT=CA-DISPATCH,              /* Associated Product       */+
      COMPONENT='CADD CADDSPL R11 *',  /* Unicenter CA-Dispatch     */+
      JOBNAME=*,                        /* Product JOB names        */+
      EXCLUDE=(X'122',X'222',X'522')   /* Ignore x22 abends         */
```

This example shows how the PRODUCT and EVENT statements are used to activate the capture of abend events for CA Dispatch. These statements also define the CA Service Desk attributes to be associated with the tickets that are opened as a result of the abend event.

The PRODUCT statement defines the product, in this case CA Dispatch.

The JOBNAME parameter defines the job names to which this product statement applies. The sample has coded '*' so that all CA Dispatch-intercepted abends are processed by this EVENT statement in any address space including customer address space with intercepted reports. The JOBNAME parameter could be used to restrict the event to a specific JOB, or leave it as it is to handle all of them. The JOBNAME parameter is also useful if more than one copy of CA Dispatch is operating in a single z/OS operating system image. For example, separate JOBNAME parameter operand specifications could be used to uniquely direct and categorize CA Service Desk tickets from each respective CA Dispatch component address space.

The EVENT parameter indicates that the events associated with this PRODUCT definition are abend events.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters identify CA Service Desk entities that are assigned to any generated tickets. These entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensures that tickets generated by CAISDI/med on behalf of CA Dispatch are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

The EVENT statement contains the crucial COMPONENT operand that is used to match this event statement with the actual event. For CA Dispatch, this is set to a value of “CADD CADDSPL R11 *”. This operand must be entered exactly as it is coded in this example.

Note: The spaces between words are significant.

CA Dispatch recovery routines place the character string “CADD CADDSPL R11” in the SDWACID field for all CA Dispatch abend events. This COMPONENT specification indicates that CAISDI/med abend event intercept should capture any detected abend event where the concatenation of the SDWACID and SDWASC fields begins with the characters “CADD CADDSPL R11”. The asterisk (*) in the sample initialization statement is a wild card, and indicates that any subsequent characters are considered a match condition.

The EXCLUDE parameter of the EVENT statement lists abend/reason code combinations for CA Dispatch that should be ignored for problem reporting. (In the sample all *22 abends are eliminated.)

The JOBNAME parameter of the EVENT statement defines the jobs where the abend event may occur. In the sample, * is coded so that all CA Dispatch-intercepted abends are processed by this EVENT statement in any address space, including customer address space with intercepted reports. The JOBNAME parameter can be used to restrict the event to a specific JOB, or masked to handle a range of jobs. The JOBNAME parameter is also useful if multiple copies of CA Dispatch are operating in a single z/OS operating system image. In this situation separate JOBNAME parameters can direct and categorize CA Service Desk tickets from each CA Dispatch address space.

The PRODUCT parameter on the EVENT statement defines the CAISDI/med product associated with the abend events captured. It is this PRODUCT name specification that makes the connection back to the CAISDI/med PRODUCT initialization statement.

For detailed information on these parameters, see Running CAISDI (see page 71).

CA JARS Resource Accounting

This product interfaces directly with CAISDI/soap. To enable CA JARS Resource Accounting to work with CA Service Desk, Position 69 of the OPTION statement in the input Control Statements needs to be set to “Y”. Any value other than “Y” in position 69 of the OPTION statement is treated as “N”, and CA Service Desk tickets will not be opened.

You can also set up the CA JARS Service Desk Option Data Set. This data set is a typical 80-byte fixed-block data set or PDS member. Use this data set to:

- Set the Priority of CA JARS Service Desk tickets
- Send the tickets to a particular instance of CA Service Desk
- Provide logon/password information for the particular instance of CA Service Desk

Set the following parameters in the CA Service Desk Option Database:

PRI={1|2|3|4|5}

Sets the priority of the CA JARS Service Desk tickets. The default is 3.

URL={*servicedesk URL*}

Sets the web address of the instance of CA Service Desk where the ticket will be opened. The default is the value of SD_URL that is set in the CAISDI/soap address space parameters.

UID={*servicedesk Userid*}

Sets the user ID that is going to be used to authenticate CA JARS to the instance of CA Service Desk where the ticket will be opened. The default value is the value of SD_USER that is set in the CAISDI/soap address space parameters.

PSW={*servicedesk Password*}

Sets the password for the User ID indicated in the UID= parameter. This is used in authenticating CA JARS to the instance of CA Service Desk where the ticket will be opened.

These text entries in the CA Service Desk Option Data Set must begin in column 1 of the line. All of the parameters are optional; however, if the UID= parameter is set, the PSW= parameter must also be provided. In addition, any line in the CA Service Desk Option Data Set that contains an asterisk in column 1 is treated as a comment and is ignored.

Note: All of the keywords and parameter values are case sensitive.

The file, once built, is allocated in the CA JARS JCL with the CAISDPRM DD Name.

A typical Option Data Set would look like this:

```
*
* CA Service Desk Optional Parameters
*

URL=http://your.local.com:8080/axis/services/USD_WebServiceSoap?wsdl
UID=servicedesk01
PSW=psw12345
PRI=2
```

For more information, see the *CA JARS Resource Accounting User Guide*.

CA Jobtrac Job Management

The CA Jobtrac Job Management product installs its event definitions into CAI.CAIEVENT. The product control member name is CHDCNTL, so include the following additional control statement in the CAISDI/els ELSSTART control member:

```
DEFINE PRODUCT=JOBTRAC,EVENTLIB=cai.caievent,PRODCNTL=chdcntl,MASK=chd*
```

The recommended values for the SETUP parameters are:

Parameter	Recommended Value
PRODUCT	JOBTRAC
PRODNAME	CA Jobtrac Job Management
ASSIGNEE	System_Jobtrac_User
AFFECTED_END_USER	System_Jobtrac_User
REPORTED_BY	System_Jobtrac_User
ASSET	CA Jobtrac Job Management
TEMPLATE	CA-Jobtrac,MEDIUM-LOW Priority

CA MIM Resource Sharing

The following CAISDI/med PRODUCT and EVENT initialization statements define the interaction between CAISDI and CA MIM Resource Sharing to generate CA Service Desk tickets for CA MIM Resource Sharing abends:

```
/*===================================================================*/
/*    Unicenter CA-MIM Product Interface Definitions                 */
/*                                                                   */
/*    Define Unicenter CA-MIM product ABEND events that are to be    */
/*    captured by CAISDI/med, and how that information is to be      */
/*    presented in the ticket to CA Service Desk.            */
/*                                                                   */

/*   Required customization:                                         */
/*                                                                   */
/*      SERVER  CAISDI/med name for Service Desk Server              */
/*                                                                   */
/*                                                                   */

/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,        */
/*          ASSIGNEE, and TEMPLATE are valid, provided the           */
/*          CA Service Desk Administrator loads predefined   */
/*          templates and contacts.  Otherwise, customize these      */
/*          parameters as directed by the Unicenter Service          */
/*          Desk Administrator.                                      */
/*                                                                   */
/*===================================================================*/
PRODUCT CA-MIM,                   /* Product name                   */+
        EVENT=ABEND,              /* Event qualifier                */+
        JOBNAME=MIMGR,            /* Product started task name      */+
        AFFECTED_END_USER=System_MIM_User,  /* Affected End User    */+
        REPORTED_BY=System_MIM_User, /* Reported By for tickets     */+
        ASSIGNEE=System_MIM_User, /* Assigned contact name          */+
        PRIORITY=2,               /* Priority for tickets           */+
     /* TEMPLATE='CA-MIM,Low Priority', /* Service Desk template    */+
        SERVER=USDSRV01           /* Direct request to this server  */

EVENT ABEND,                      /* Capture ABEND events           */+
      PRODUCT=CA-MIM,             /* Associated product statement   */+
      COMPONENT='MiMgr*', /*Unicenter CA-MIM component (mixed case) */+
      JOBNAME=MIMGR,              /* Product started task name      */+
      EXCLUDE=((0051,0000),       /* Internal entry to VCF recovery */+
              (0051,X'10'))       /* Normal MIGRATE CTC to DASD     */
```

This example shows how the PRODUCT and the EVENT statement are used to activate the capture of a named product's abend events and to define the associated CA Service Desk attributes to be associated with tickets that are opened as a result of the abend event.

Beginning with the EVENT statement, the crucial specification provided is the COMPONENT parameter, MiMgr.

Note: The specification is made with uppercase and lowercase letters.

CA MIM Resource Sharing recovery routines always place the character string, MiMgr, in the SDWACID field for all CA MIM Resource Sharing abend events. Therefore, this COMPONENT specification indicates that CAISDI/med abend event intercept should capture any detected abend event where the concatenation of the SDWACID and SDWASC fields begins with the characters MiMgr. The asterisk (*) in the sample initialization statement is a wild card and indicates that any subsequent characters are considered a match condition.

The EXCLUDE parameter of the EVENT statement lists several abend code/reason code combinations for CA MIM Resource Sharing that are to be ignored for problem reporting. These abends are used in normal CA MIM Resource Sharing operation for Virtual Control File processing and should not be reported as errors.

The JOBNAME parameter of the EVENT statement names the job, MIMGR. The JOBNAME parameter is useful if more than one copy of CA MIM Resource Sharing is operating in a single z/OS operating system image. For example, perhaps the CA MII Data Sharing facility executes with a job name of CAMII while the CA MIA Tape Sharing facility executes with a job name of CAMIA. In this case, multiple EVENT statements with separate JOBNAME parameter specifications could be used to uniquely direct and categorize CA Service Desk tickets from each respective CA MIM Resource Sharing component address space.

Finally, the PRODUCT parameter defines the CAISDI/med product name to associate with abend events captured due to a matching COMPONENT specification. It is this PRODUCT name specification that makes the connection to a specific CAISDI/med PRODUCT initialization statement.

The preceding PRODUCT statement defines the CA-MIM product as referenced in the PRODUCT= parameter of the EVENT statement. Again, the JOBNAME parameter defines the job name of MIMGR as the qualifier to which CA MIM Resource Sharing address space this PRODUCT statement applies. As discussed earlier, the JOBNAME parameter is useful when the various CA MIM Resource Sharing facilities are operated in separate z/OS address spaces.

The PRODUCT statement EVENT parameter qualifies that the product events associated with this product definition are abend events.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters define the names of predefined CA Service Desk specific entities that are to be assigned to any generated tickets. These named entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensure that tickets generated by CAISDI/med on behalf of the named product are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the Priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

For detailed information on these parameters, see Running CAISDI (see page 71).

CA OPS/MVS Event Management and Automation

The following CAISDI/med PRODUCT and EVENT initialization statements define the interaction between CAISDI and CA OPS/MVS Event Management and Automation to generate CA Service Desk tickets for CA OPS/MVS Event Management and Automation product abends as well as internally detected PROBLEMs:

```
/*============================================================*/
/*    Unicenter CA-OPS/MVS Product Interface Definitions      */
/*                                                            */
/*    Define Unicenter CA-OPS/MVS product ABEND and PROBLEM   */
/*    events that are to be captured by CAISDI/med, and how   */
/*    that information is to be presented in the request ticket */
/*    to CA Service Desk.                                                */
/*                                                            */
/*   Required customization:                                  */
/*                                                            */
/*      SERVER  CAISDI/med name for Service Desk Server       */
/*                                                            */
/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,  */
/*          ASSIGNEE, and TEMPLATE are valid, provided the    */
/*          CA Service Desk Administrator loads predefined   */
/*          templates and contacts.  Otherwise, customize these */
/*          parameters as directed by the Unicenter Service   */
/*          Desk Administrator.                               */
/*                                                            */
/*============================================================*/

PRODUCT CA-OPS/MVS,              /* Define product name           */+
        EVENT=(ABEND,PROBLEM),   /* Event qualifiers              */+
        JOBNAME=OPSMAIN,         /* Product started task name     */+
        AFFECTED_END_USER=System_OPSMVS_User,   /*Affected End User */+
        REPORTED_BY=System_OPSMVS_User,  /* Reported By for tickets */+
        ASSIGNEE=System_OPSMVS_User, /*Assigned contact name       */+
        PRIORITY=2,              /* Priority for tickets          */+
     /* TEMPLATE='CA-OPSMVS,Priority unassigned',  /* Template     */+
        SERVER=USDSRV01          /* Direct request to this server  */

EVENT ABEND,                  /* Capture ABEND events             */+
      COMPONENT='OPS*',       /* Unicenter CA-OPS/MVS component ID */+
      PRODUCT=CA-OPS/MVS      /* Associated product statement     */
```

The preceding example shows how the PRODUCT and the EVENT statement are used to activate the capture of a named product's abend events and to define the associated CA Service Desk attributes to be associated with tickets that are opened as a result of the abend event. The PRODUCT statement EVENT parameter operands indicate that the product definition is to be used for both abend events captured by CAISDI as well as PROBLEM events generated by CA OPS/MVS Event Management and Automation itself.

Beginning with the EVENT statement, the crucial specification is the COMPONENT operand, OPS. CA OPS/MVS Event Management and Automation recovery routines always place the character string, OPS, in the SDWACID field for all CA OPS/MVS Event Management and Automation abend events. Therefore, this COMPONENT specification indicates that CAISDI abend event intercept should capture any detected abend event where the concatenation of the SDWACID and SDWASC fields begins with the characters OPS. The asterisk (*) in the sample initialization statement is a wild card and indicates that any subsequent characters are considered a match condition.

The PRODUCT parameter, CA-OPS/MVS, defines the CAISDI product name to associate with abend events captured due to a matching COMPONENT specification. It is this PRODUCT name specification that makes the connection to a specific CAISDI PRODUCT initialization statement.

The preceding PRODUCT statement defines CA OPS/MVS Event Management and Automation as referenced in the PRODUCT= parameter of the EVENT statement. The PRODUCT statement EVENT parameter qualifies that the product events associated with this product definition are abend events, and PROBLEM events. CA OPS/MVS Event Management and Automation can report on internally detected problems using a proprietary interface to CAISDI. Internal tickets generated using this interface specify a product name of “CA OPS/MVS Event Management and Automation” and an event type qualifier of “PROBLEM.” Therefore, a single CAISDI PRODUCT name of CA OPS/MVS Event Management and Automation with an event list of abend and PROBLEM are used to assign CA Service Desk tickets.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters define the names of predefined CA Service Desk specific entities that are to be assigned to any generated tickets. These named entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensures that tickets generated by CAISDI on behalf of the named product are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the Priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

For detailed information on these parameters, see Running CAISDI (see page 71).

CA Spool Enterprise Print Management

CA Spool Enterprise Print Management uses CAISDI/med to process two types of events:

- product messages
- abends

Message Events

To create CA Service Desk tickets for product messages, CA Spool Enterprise Print Management uses CAISDI/med services to interface with the CAISDI/soap task to open a CA Service Desk tickets based upon a CA Spool Enterprise Print Management message id. To activate this, in the CA Spool Enterprise Print Management parameters add SDI=YES. You control which CA Spool Enterprise Print Management message cause a CA Service Desk ticket to be opened by adding SDIREQ=YES to the MESSAGE statement for that message number. Optionally, for some or all of these messages, you can code EXIT=YES on the MESSAGE statement to invoke user exit ESFMSG0 before making the request to open a CA Service Desk ticket. With this exit the text of the CA Service Desk ticket can be updated or the creation of a CA Service Desk ticket can be skipped completely.

In addition to the changes to the CA Spool Enterprise Print Management parameters and MESSAGE statements, a PRODUCT initialization statement must be added to the CAISDI parameters to define the interaction between CAISDI and CA Spool Enterprise Print Management to generate the CA Service Desk tickets for the selected messages. Here is a sample of the PRODUCT statements:

```
/*============================================================*/
/* Unicenter CA-Spool Product Interface Definitions           */
/*                                                            */
/* Define Unicenter CA-Spool product ABEND and message events that  */
/* are to be captured by CAISDI/med, and how that information is    */
/* to be presented in the ticket to CA Service Desk.                */
/*                                                            */
/*   Required customization:                                  */
/*                                                            */
/*      SERVER  CAISDI/med name for Service Desk Server       */
/*                                                            /
/*                                                            */
/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,  */
/*          ASSIGNEE, and TEMPLATE are valid, provided the    */
/*          CA Service Desk Administrator loads predefined   */
/*          templates and contacts.  Otherwise, customize these  */
/*          parameters as directed by the Unicenter Service   */
/*          Desk Administrator.                               */
/*                                                            */
/*============================================================*/
PRODUCT CA-SPOOL,                       /* Product name            */+
        EVENT=(%ESF886),                /* Event qualifier         */+
        AFFECTED_END_USER=System_Spool_User, /* Affected End User  */+
        REPORTED_BY=System_Spool_User,  /* Reported by for tickets */+
        ASSIGNEE=System_Spool_User,     /* Assigned contact name   */+
        PRIORITY=2,                     /* Priority for tickets    */+
     /* TEMPLATE='CA-Spool,Priority unassigned', /* template name  */+
        SERVER=USDSRV01         /* Direct requests to this server  */

PRODUCT CA-SPOOL,                       /* Product name            */+
        EVENT=(ABEND,%ESF*),            /* Event qualifier         */+
        AFFECTED_END_USER=System_Spool_User, /* Affected End User  */+
        REPORTED_BY=System_Spool_User,  /* Reported by for tickets */+
        ASSIGNEE=System_Spool_User,     /* Assigned contact name   */+
        PRIORITY=3,                     /* Priority for tickets    */+
     /* TEMPLATE='CA-Spool,Priority unassigned', /* template name  */+
        SERVER=USDSRV01         /* Direct requests to this server  */
```

This sample shows two PRODUCT statements that are used to activate the capture of message events for CA Spool Enterprise Print Management. The PRODUCT statement defines the following:

- The product
- The event
- The name of the CAISDI/med server that will process this event
- The CA Service Desk attributes to be associated with the tickets that are opened as a result of the message events.

If all of the CA Service Desk attributes are the same for the message events, then only one PRODUCT statement would be necessary. However, the severity of messages can vary, and this sample shows how you can open CA Service Desk tickets with different priorities for different messages.

The EVENT parameter indicates the events associated with this PRODUCT definition. The event ID associated with the messages produced by CA Spool Enterprise Print Management is composed of the command character followed by the message id. For example, the event ID for message ESF355 could be -ESF355. This sample contains two PRODUCT statements, the first for message ESF886 and the second for all other CA Spool Enterprise Print Management messages by using the event name of %ESF*.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters identify CA Service Desk entities that are assigned to any generated tickets. These entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server. Note that the PRIORITY parameter is the only one that is different in the two preceding PRODUCT definitions. You could also set different templates and/or end user for the different messages.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensure that tickets generated by CAISDI/med on behalf of CA Spool Enterprise Print Management are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

For detailed information on these parameters, see Running CAISDI (see page 71).

Abend Events

The CAISDI/med PRODUCT and EVENT initialization statements are used to define the interaction between CAISDI and CA Spool Enterprise Print Management to generate CA Service Desk tickets for CA Spool Enterprise Print Management abends events. The following sample shows the addition of an Abend event definition to the Message Events defined in the preceding sample.

```
/*=============================================================*/
/* Unicenter CA-Spool Product Interface Definitions            */
/*                                                             */
/* Define Unicenter CA-Spool product ABEND and message events that  */
/* are to be captured by CAISDI/med, and how that information is    */
/* to be presented in the request ticket to CA Service Desk.              */
/*                                                             */
/*   Required customization:                                   */
/*                                                             */
/*      SERVER  CAISDI/med name for Service Desk Server        */
/*                                                             */
/*                                                             */
/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,  */
/*          ASSIGNEE, and TEMPLATE are valid, provided the     */
/*          CA Service Desk Administrator loads predefined                */
/*          templates and contacts.  Otherwise, customize these  */
/*          parameters as directed by the Unicenter Service    */
/*          Desk Administrator.                                */
/*                                                             */
/*=============================================================*/

PRODUCT CA-SPOOL,                       /* Product name            */+
        EVENT=(%ESF886),                /* Event qualifier         */+
        AFFECTED_END_USER=System_Spool_User, /* Affected End User  */+
        REPORTED_BY=System_Spool_User,  /* Reported by for tickets */+
        ASSIGNEE=System_Spool_User,     /* Assigned contact name   */+
        PRIORITY=2,                     /* Priority for tickets    */+
     /* TEMPLATE='CA-Spool,Priority unassigned', /* template name  */+
        SERVER=USDSRV01         /* Direct requests to this server  */

PRODUCT CA-SPOOL,                       /* Product name            */+
        EVENT=(ABEND,%ESF*),            /* Event qualifier         */+
        AFFECTED_END_USER=System_Spool_User, /* Affected End User  */+
        REPORTED_BY=System_Spool_User,  /* Reported by for tickets */+
        ASSIGNEE=System_Spool_User,     /* Assigned contact name   */+
        PRIORITY=3,                     /* Priority for tickets    */+
     /* TEMPLATE='CA-Spool,Priority unassigned', /* template name  */+
        SERVER=USDSRV01         /* Direct requests to this server  */

EVENT ABEND,                           /* Capture ABEND events    */+
      PRODUCT=CA-SPOOL,                /* Associated Product      */+
      COMPONENT='SPOOL-Main*',         /*Unicenter CA-Spool       */+
```

```
        JOBNAME=ESF*                    /* Product JOB names         */
```

This sample shows two PRODUCT statements that are used to activate the capture of message, and an EVENT ABEND statement that defines CA Spool Enterprise Print Management abend events to be captured by CAIDSI.

The PRODUCT statement defines the product, the event, the name of the CA CAISDI/med server that will process this event, and the CA Service Desk attributes to be associated with the tickets that are opened as a result of the defined event.

The EVENT parameter indicates the events associated with this PRODUCT definition. In addition to the message event IDs that were discussed previously, the second PRODUCT statement also has defined an event named ABEND. This event matches up with the EVENT ABEND statement that is discussed later in this section. To completely eliminate the message event, you would only need to code EVENT=ABEND on the PRODUCT statement.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters identify CA Service Desk entities that are assigned to any generated tickets. These entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensure that tickets generated by CAISDI/med on behalf of CA Spool Enterprise Print Management are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the Priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

The EVENT statement contains the crucial COMPONENT operand that is used to match this event statement with the actual event. For CA Spool Enterprise Print Management, this is set to a value of “SPOOL-Main.” This operand must be entered exactly as it is coded in this example. *Note that case is significant for this value.*

CA Spool Enterprise Print Management recovery routines place the character string “SPOOL-Main” in the SDWACID field for all CA Spool Enterprise Print Management abend events. This COMPONENT specification indicates that CAISDI/med abend event intercept should capture any detected abend event where the concatenation of the SDWACID and SDWASC fields begins with the characters “SPOOL-Main”.

The EXCLUDE parameter of the EVENT statement can be used to list abend/reason code combinations for CA Spool Enterprise Print Management that should be ignored for problem reporting. We recommend that all abends be reported.

The JOBNAME parameter of the EVENT statement defines the jobs where the abend event may occur. In the sample, ESF* is coded to specify that we want to create a ticket for abend events for task names beginning with ESF. The JOBNAME parameter can be used to restrict the event to a specific job or task, or masked to handle a range of jobs or tasks. The JOBNAME parameter is also useful if multiple copies of CA Spool Enterprise Print Management are operating in a single z/OS operating system image. In this situation, separate JOBNAME parameters can direct and categorize CA Service Desk tickets from each CA Spool Enterprise Print Management address space.

The PRODUCT parameter on the EVENT statement defines the CAISDI/med product associated with the abend events captured. This PRODUCT name specification makes the connection back to the CAISDI/med PRODUCT initialization statement.

For detailed information on these parameters, see Running CAISDI (see page 71).

CA SYSVIEW Performance Management

The following CAISDI/med PRODUCT and EVENT initialization statements define the interaction between CAISDI and CA SYSVIEW Performance Management to generate CA Service Desk tickets for CA SYSVIEW Performance Management product abends:

```
/*=================================================================*/
/*    Unicenter CA-SYSVIEW Product Interface Definitions           */
/*                                                                 */
/*    Define Unicenter CA-SYSVIEW product ABEND events that are    */
/*    to be captured by CAISDI/med, and how that information is    */
/*    to be presented in the ticket to CA Service Desk.                  */
/*                                                                 */
/*   Required customization:                                       */
/*                                                                 */
/*      SERVER  CAISDI/med name for Service Desk Server            */
/*                                                                 */
/*                                                                 */
/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,      */
/*          ASSIGNEE, and TEMPLATE are valid, provided the         */
/*          CA Service Desk Administrator loads predefined   */
/*          templates and contacts.  Otherwise, customize these    */
/*          parameters as directed by the Unicenter Service        */
/*          Desk Administrator.                                    */
/*                                                                 */
/*=================================================================*/

PRODUCT CA-SYSVIEW,              /* Define product name           */+
        EVENT=ABEND,             /* Event qualifier               */+
        AFFECTED_END_USER=System_Sysview_User, /* Affected End User */+
        REPORTED_BY=System_Sysview_User, /* Reported By for tickets */+
        ASSIGNEE=System_Sysview_User,  /* Assigned contact name    */+
        PRIORITY=2,              /* Priority for tickets          */+
     /* TEMPLATE='CA-Sysview,Priority unassigned',  /* Template    */+
        SERVER=USDSRV01          /* Direct request to this server  */

EVENT ABEND,                  /* Capture ABEND events             */+
      COMPONENT='GSVX*',      /* Unicenter CA-SYSVIEW component ID */+
      PRODUCT=CA-SYSVIEW      /* Associated product statement      */
```

The preceding example shows how the PRODUCT and the EVENT statement are used to activate the capture of a named product's abend events and to define the associated CA Service Desk attributes to be associated with tickets that are opened as a result of the abend event.

Beginning with the EVENT statement, the crucial specification provided is the COMPONENT operand, *GSVX*. CA SYSVIEW Performance Management recovery routines always place the character string, GSVX, in the SDWACID field for all CA SYSVIEW Performance Management abend events. Therefore, this COMPONENT specification indicates that CAISDI abend event intercept should capture any detected abend event where the concatenation of the SDWACID and SDWASC fields begins with the characters GSVX. The asterisk (*) in the sample initialization statement is a wild card and indicates that any subsequent characters are considered a match condition.

The PRODUCT parameter operand, CA SYSVIEW Performance Management, defines the CAISDI product name to associate with abend events captured due to a matching COMPONENT specification. It is this PRODUCT name specification that makes the connection to a specific CAISDI PRODUCT initialization statement.

The preceding PRODUCT statement defines the CA SYSVIEW Performance Management product as referenced in the PRODUCT= parameter of the EVENT statement. The PRODUCT statement EVENT parameter qualifies that the product events associated with this product definition are abend events.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters define the names of predefined CA Service Desk specific entities that are to be assigned to any generated tickets. These named entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensures that tickets, generated by CAISDI on behalf of the named product, are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the Priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

For detailed information on these parameters, see Running CAISDI (see page 71).

CA View

The following CAISDI/med PRODUCT and EVENT initialization statements define the interaction between CAISDI and CA View to generate CA Service Desk tickets for CA View abends:

```
/*================================================================*/
/* Unicenter CA-View Product Interface Definitions                */
/*                                                                */
/* Define Unicenter CA-View product ABEND events that are to be   */
/* captured by CAISDI/med, and how that information is to be      */
/* presented in the request ticket to CA Service Desk.                  */
/*                                                                */
/*   Required customization:                                      */
/*                                                                */
/*      SERVER  CAISDI/med name for Service Desk Server           */
/*                                                                */
/*                                                                */
/*   NOTE:  Sample values for AFFECTED_END_USER, REPORTED_BY,     */
/*          ASSIGNEE, and TEMPLATE are valid, provided the        */
/*          CA Service Desk Administrator loads predefined              */
/*          templates and contacts.  Otherwise, customize these   */
/*          parameters as directed by the Unicenter Service       */
/*          Desk Administrator.                                   */
/*                                                                */
/*================================================================*/

PRODUCT CA-VIEW,                        /* Product name            */+
        EVENT=ABEND,                    /* Event qualifier         */+
        JOBNAME=SAR*,                   /* Product JOB names       */+
        AFFECTED_END_USER=System_View_User, /* Affected End User   */+
        REPORTED_BY=System_View_User,   /* Reported by for tickets */+
        ASSIGNEE=System_View_User,      /* Assigned contact name   */+
        PRIORITY=3,                     /* Priority for tickets    */+
     /* TEMPLATE='CA-View,Priority unassigned', /* template name   */+
        SERVER=USDSRV01         /* Direct requests to this server  */

EVENT ABEND,                           /* Capture ABEND events     */+
      PRODUCT=CA-VIEW,                 /* Associated Product       */+
      COMPONENT='VIEW SAR*',           /*Unicenter CA-View component*/+
      JOBNAME=SAR*,                    /* Product JOB names        */+
      EXCLUDE=((0522,),(X'522',))      /* Ignore session time-outs */
```

This example shows how the PRODUCT and EVENT statements are used to activate the capture of abend events for CA View. These statements also define the CA Service Desk attributes to be associated with the tickets that are opened as a result of the abend event.

The PRODUCT statement defines the product, in this case CA View.

The JOBNAME parameter defines the jobname(s) to which this product statement applies. In the preceding sample, all jobs beginning with 'SAR*' would qualify for the processes defined for this product statement.

The EVENT parameter indicates that the events associated with this PRODUCT definition are abend events.

The SERVER parameter refers to the SERVER statement that defines the CA Service Desk server that is to receive tickets associated with this product definition.

The remaining parameters identify CA Service Desk entities that are assigned to any generated tickets. These entities should be predefined by the CA Service Desk administrator on the target CA Service Desk server.

The most important predefined entity is the CA Service Desk TEMPLATE name. The proper use of predefined CA Service Desk templates ensure that tickets generated by CAISDI/med on behalf of CA View are properly categorized, prioritized, and assigned.

The AFFECTED_END_USER parameter is used to define the contact name that appears in the Affected End User field of the ticket.

The REPORTED_BY parameter is used to define the contact name that appears in the Reported By field of the ticket.

The PRIORITY parameter defines the priority of the ticket.

The ASSIGNEE parameter is used to define the contact name that appears in the Assignee field of the ticket.

The EVENT statement contains the crucial COMPONENT operand that is used to match this event statement with the actual event. This is set to a value of 'VIEW SAR*' for CA View. This operand must be entered exactly as it is coded in this example.

Note: The space between VIEW and SAR in the parameter definition is significant.

CA View recovery routines will place the character string 'VIEW SAR' in the SDWACID field for all CA View abend events. This COMPONENT specification indicates that CAISDI/med abend event intercept should capture any detected abend event where the concatenation of the SDWACID and SDWASC fields begins with the characters VIEW SAR' The asterisk (*) in the sample initialization statement is a wild card and indicates that any subsequent characters are considered a match condition.

The EXCLUDE parameter of the EVENT statement lists abend/reason code combinations for CA View that should be ignored for problem reporting. (In the sample the S522 abends are eliminated because they are normal in CA View for XMS session time-outs.)

The JOBNAME parameter of the EVENT statement defines the job where the abend event may occur. In the sample, 'SAR*' is coded so that all intercepted abends are processed by this EVENT statement in CA View jobs such as SARSTC and SARDBASE. The JOBNAME parameter can be used to restrict the event to a specific JOB, or masked to handle a range of jobs. The JOBNAME parameter is also useful if multiple copies of CA View are operating in a single z/OS operating system image. In this situation, separate JOBNAME parameters can direct and categorize CA Service Desk tickets from each CA View address space.

The PRODUCT parameter on the EVENT statement defines the CAISDI/med product associated with the abend events captured. It is this PRODUCT name specification that makes the connection back to the CAISDI/med PRODUCT initialization statement.

For detailed information on these parameters, see Running CAISDI (see page 71).

CA DB2 Tools

CA DB2 Tools comprises a number of products for administration, backup and recovery, and performance management of DB2 databases.

Each of these products supports CA Service Desk integration:

- CA Value Pack for DB2
- CA Data Compressor for DB2 for z/OS
- CA Database analyzer for DB2 for z/OS
- CA Fast Check for DB2 for z/OS
- CA Fast Index for DB2 for z/OS
- CA Fast Load for DB2 for z/OS
- CA Fast Recover for DB2 for z/OS
- CA Fast Unload for DB2 for z/OS
- CA Index Expert for DB2 for z/OS
- Ca Log Analyzer for DB2 for z/OS
- CA Merge/Modify for DB2 for z/OS
- CA Partition Expert for DB2 for z/OS
- CA Plan Analyzer for DB2 for z/OS
- CA Quick Copy for DB2 for z/OS
- CA Rapid Reorg for DB2 for z/OS
- CA RC/Extract for DB2 for z/OS
- CA RC/Migrator for DB2 for z/OS

- CA RC/Query for DB2 for z/OS
- CA RC/Secure for DB2 for z/OS
- CA RC/Update for DB2 for z/OS
- CA Recovery Analyzer for DB2 for z/OS
- CA SQL Ease for DB2 for z/OS

These products all use the CAISDI/els interface. They share the same configuration, so configuring the interface for one will configure all of them.

CA DB2 Tools Setup Parameters

The CA DB2 Tools products install their event definitions into CAI.CAIEVENT. The product control member name is GENCNTL, so include the following additional control statement in the CAISDI/els ELSSTART control member:

```
DEFINE PRODUCT=DB2TOOLS,EVENTLIB=CAI.CAIEVENT,PRODCNTL=GENCNTL,MASK=GEN*
```

The recommended values for the SETUP parameters are:

Parameter	Recommended Value
PRODUCT	DB2TOOLS
PRODNAME	CA DB2 Tools
ASSIGNEE	System_DB2Tools_User
AFFECTED_END_USER	System_DB2Tools_User
REPORTED_BY	System_DB2Tools_User
ASSET	CA DB2 Tools
TEMPLATE	CA-DB2Tools,Low Priority

For more information on customizing the GENCNTL product control member, see the comments in that member.

To activate CAISDI for the CA DB2 Tools products, set the SDI parameter to “YES” in the SETUPxx member of PARMLIB.

CA NetMaster Network Management Products

These products interface directly with CAISDI/soap. CA NetMaster Network Management comprises a suite of products that monitor and manage z/OS IP and SNA network devices and activity.

Beginning with r11, each of the following products support CA Service Desk Integration:

- CA NetMaster Network Management for TCP/IP
- CA NetSpy Network Performance
- CA NetMaster Network Operations for TCP/IP
- CA NetMaster Network Management for SNA
- CA NetMaster Network Automation
- CA NetMaster File Transfer Management

CA NetMaster Alert Monitor

The CA NetMaster Alert Monitor displays real-time, dynamic alerts for a wide variety of network and system operational and performance conditions. An alert can be generated when a specific critical event occurs or when a performance attribute exceeds a specified threshold.

With the Alert Monitor, CA NetMaster Network Management customers have control over all aspects of alerting, including:

- What alerts are raised, and why
- Individual alert attributes, such as severity and description text
- What the alert displays look like, using filtering, sorting, and ordering

For more information on using the Alert Monitor, see the CA NetMaster Network Management documentation.

Integrating CAISDI with the CA NetMaster Alert Monitor

This section describes how to integrate CAISDI with the CA NetMaster Alert Monitor.

Requirements

This section describes the requirements for configuring the CA NetMaster Alert Monitor with CAISDI.

CAISDI/soap

CA NetMaster Alert Monitor CAISDI integration uses the CAISDI/soap component. CA NetMaster does not require implementation of the CAISDI/med or CAIDSDI/els components, although you may require these for other products.

CA Service Desk HTML Rendering

By default CA NetMaster Network Management generates CA Service Desk ticket descriptions in HTML format. Your CA Service Desk server must be configured to render HTML tags in the description field correctly. For more information, see HTML Rendering (see page 42).

Instructions

For information about implementing and using CAISDI with CA NetMaster products, see the CA Network Management r11 *Common Administration Guide* and *Common User Guide*.

CA MICS Resource Management

This product interfaces directly with CAISDI/soap. CA MICS Resource Management intercepts user abends as documented in CA MICS Resource Management *Planning, Installation, Operation,* and *Maintenance Guid*e. To enable CA MICS Resource Management support for CAISDI, the following parameters must be added to the member *prefix*.MICS.PARMS(EXECDEF) for the individual CA MICS Resource Management Database Units.

SD_ACTIVE YES | NO

Enables CA MICS Resource Management support for CAISDI. Default is NO

SD_URL *url_name*

Sets the web address of the instance of the CA Service Desk where the ticket will be opened. The default is the value of SD_URL that is set in the CAISDI/soap address space parameters.

SD_USERID *userid*

Sets the User ID that is going to be used to authenticate CA MICS Resource Management to the instance of CA Service Desk where the ticket will be opened. The default value is the value of SD_USER that is set in the CAISDI/soap address space parameters.

SD_PASSWORD *password*

Sets the password for the User ID indicated in the UID= parameter. This is used in authenticating CA MICS Resource Management to the instance of CA Service Desk where the ticket will be opened.

SD_TEMPLATE *template_name*

Sets the name of the CA Service Desk template used to open CA Service Desk tickets. The name of the predefined template is CA-NeuMICS,Low Priority.

For more information, see the *CA MICS Resource Management Planning, Installation, Operation, and Maintenance Guid*e.

Appendix A: CAISDI/elmds Backout Procedure

If you have trouble migrating to CAISDI/elmds, you can run the standalone CAISDI/med or CAISDI/els components separately instead of as a component of CAISDI/elmds. If you need to run the standalone els or med components, then backout the component from elmds and configure the individual component separately.

To backout the med or els component, follow these steps:

1. Before starting CAISDI, set the desired backout parameters for elmds. See Configure the CAISDI/elmds Parameters (see page 55) for instructions on the following:
   - To backout med, set the CDYFCNFG NO_MED statement
   - To backout els, set the CDYFCNFG NO_ELS statement
2. Perform implementation steps for the standalone component being used:
   - For med, see How to Implement Standalone CAISDI/med (see page 113)
   - For els, see How to Implement Standalone CAISDI/els (see page 137)

How to Implement Standalone CAISDI/med

After installing standalone CAISDI/med using the procedures in the CA Common Services *Installation Guide*, follow this process to ensure you complete all the customization tasks for the standalone CAISDI/med component:

1. Customize and Stage the CAISDI/med JCL PROC (see page 114)
2. Define a CAISDI/med Product Security User ID (see page 58)
3. Customize the CA Service Desk parameters (see page 59)
4. Customize the CAISDI/med Product Parameters (see page 60)
5. Run Standalone CAISDI/med (see page 114)

Note: On a z/OS system, standalone CAISDI/med uses the CAISDI/soap address space as a medium through which it communicates with the CA Service Desk Web Server. CAISDI/soap uses the SOAP communications protocol to exchange data with the CA Service Desk Web Server, using HTTP over a TCP/IP network. CAISDI/soap must have logon access to the CA Service Desk server for standalone CAISDI/med to function. For more information, see the section How to Configure CAISDI/soap (see page 43).

Customize and Stage the CAISDI/med JCL PROC

A sample CAISDI/med started task JCL procedure is contained in member CASDIMED of the CAI.CAW0PROC data set.

To customize and stage the CAISDI/med JCL PROC

1. Copy CAI.CAW0PROC(CASDIMED) to SYS2.PROCLIB(CASDIMED) or to any JCL PROCLIB that is automatically searched as part of z/OS START command processing.
2. Customize the //MEDPARMS DD statement by specifying the name of the CAISDI/med parameter data set and (member) that contains CAISDI/med initialization statements.

Note the PARM parameter in the EXEC statement. The values contained in the PARM field are the same parameters that can be specified on the CAISDI/med INIT statement. Any INIT statement keyword can be specified on the EXEC PARM field. This provides a means to override these values on the z/OS START command.

This step is complete when you have customized and staged the CAISDI/med JCL PROC to a searched z/OS JCL procedure library.

Run Standalone CAISDI/med

This section describes how to start and run the standalone CAISDI/med component of CAISDI.

This section contains the following topics:

CAISDI/med Control Statements (see page 114)
CAISDI/med Commands (see page 124)
CAISDI/med Messages (see page 137)

CAISDI/med Control Statements

This section describes CAISDI/med control statements.

Customizing the CAISDI/med Address Space

During start up, CAISDI/med reads and processes statements from an initialization data set. These statements are used to customize the operation of CAISDI/med. Additionally, the initialization statements provide a means to tailor the CA Service Desk tickets that are opened for a given CA product. The MEDPARMS DD statement in the CASDIMED JCL procedure points to the CAISDI/med initialization data set.

CAISDI/med Initialization Statement Syntax

Initialization statements are processed as 80-byte control card images. However, only card columns 1-71 are examined for parsing purposes. Card columns 72-80 are ignored. Comments are supported and begin with a /* and terminate with a */. Comments can extend over any number of card images. Cards containing spaces in card columns 1-71 are ignored, and can be used as separation between comment statements or control card specifications. Control card operands are not column-dependent and can begin in any card column from 1 to 71. At least one space should be placed between the control card statement and the first control card parameter. Commas or blanks should separate sequences of control card parameters and their operand values.

The following example illustrates the general rules for control card data entry.

```
/*-------------------------------------------------------
 This is a comment that extends over multiple cards
 -------------------------------------------------------*/
STATEMENT PARAMETER=OPERAND1,PARAMETER2=OPERAND2 /*COMMENT*/
```

Operand specifications for parameters that support a list of operand values should have the entire operand specification enclosed in parentheses and have the individual operand values separated by commas. The following example shows the general rule for coding operand list sequences:

```
STATEMENT PARAMETER=(OPERAND1,OPERAND2,...)   /*COMMENT*/
```

Control card continuation is supported over any number of control card images. Control card continuations should be placed following a parameter/operand sequence. To continue a control card, a plus sign (+) should be placed in any card column following the last operand on the current card image, up to column 71. Comments are permitted between the last operand and the plus sign. However, the comment must end before the plus sign.

The following example illustrates the general rules for continuing control card specifications:

```
/*-------------------------------------------------
 This is a comment that extends over multiple cards
-------------------------------------------------*/
STATEMENT PARAMETER=OPERAND1,    /* OPTIONAL COMMENT */ +
          PARAMETER=(OPERAND2a,  /* OPTIONAL COMMENT */ +
          OPERAND2b)             /* OPTIONAL COMMENT */
```

Defining CAISDI/med Address Space Options

This section describes how to define CAISDI/med address space options.

INIT Control Statement

The INIT statement is used to define initialization settings that control the execution of the CAISDI/med address space. The INIT statement is unique in that it can be specified as the PARM= value on the // EXEC card in the CASDIMED started task procedure as well as in the CAISDI/med initialization data set.

Use this syntax for the INIT statement:

```
INIT [DEBUGMSG={YES|NO}] [REUSE={YES|NO}] [MEDNAME=SDIA]
```

DEBUGMSG

This keyword parameter controls the production of additional diagnostic messages for various events within the CAISDI/med address space. You can specify one of the following:

YES - produce diagnostic messages

NO - do not produce diagnostic messages

REUSE

This parameter indicates whether the CAISDI/med should attempt to reuse its common storage intercepts and control blocks if obtained during a previous execution. Specify NO or YES.

Important! We strongly recommend that you specify REUSE=YES as the default. However, certain PTFs may require that CAISDI/med be restarted with REUSE=NO. In any other case, REUSE=NO should not be used, unless directed by CA Support.

MEDNAME

This parameter is used to uniquely identify a given instance of CAISDI/med in a given z/OS image. The value can be one- to four-characters long and include alpha, numeric, or special characters. This parameter is useful should an installation run more than one copy of the CAISDI/med per z/OS image. The default value for this parameter is SDIA. You should specify no other value unless directed to do so by CA Support.

The following is an example of the INIT control statement:

```
INIT REUSE=NO,
     MEDNAME=SDIA
```

Identifying a CA Service Desk Server

This section describes how to use the SERVER control statement to identify a CA Service Desk Server.

SERVER Control Statement

The SERVER statement is used to define an instance of a CA Service Desk server to the CAISDI/med.

Note: At least one valid SERVER statement must be specified in the MEDPARMS initialization parameter data set; otherwise, CAISDI/med will not start.

Use this syntax for the SERVER statement:

```
SERVER server_name, HOSTNAME=name, TIMEOUT=seconds, URL_SYMBOL=name
```

server_name

This required positional parameter defines a one- to eight-character name that is used to identify the given instance of a CA Service Desk server. The server_name is referred to on PRODUCT statements to direct CA Service Desk tickets, generated by a given product, to a specific CA Service Desk server.

The server_name also appears in several operator command response messages where appropriate.

HOSTNAME

This optional parameter defines the CAICCI host name where the CA SOAP Client Service address space is executing. The CA SOAP Client Service address space provides the gateway for the actual SOAP transaction being invoked. If not specified, the default assumes that the CA SOAP Client Service address space is running on the local host.

TIMEOUT

This optional parameter defines the number of seconds the CA SOAP Client Service should wait for a response from the CA Service Desk Web service before returning a timeout condition.

The time-out value can be a number between 10 seconds and 3600 seconds. The default value is 30 seconds.

URL_SYMBOL

Optional. Used to identify the CA Service Desk server on which you want the tickets opened. If omitted, the SD_URL value from the CAISDI/soap address space's Environmental Variables are used. You may select an alternate CA Service Desk server by assigning a “symbol” to URL_SYMBOL where URL_symbol is a defined Environmental Variable in the CAISDI/soap server and where its value is a URL identifying the CA Service Desk server. For example, if “URL_SYMBOL=WS2” then Environmental Variable “URL_WS2” must be defined to the CAISDI/soap server with a value which is the URL for the CA Service Desk server. See the CASOAPEnvironmental Variables (see page 52).

The following is an example of the SERVER control statement:

```
SERVER USDSRV01,
       HOSTNAME=XYZHOST1,
       TIMEOUT=30,
       URL_SYMBOL=WS2
```

Excluding Abend Event Reporting

This section describes how to use the EXCLUDE control statement to exclude abend event reporting.

EXCLUDE Control Statement

The EXCLUDE statement is used to specify initialization settings that define events that should be ignored by the CAISDI/med. For instance, abend codes that are to be ignored for all CA products by the CAISDI/med are defined using the EXCLUDE statement. Note the abend code specifications provided on the EXCLUDE statement are used in addition to abend code specifications provided on any given product specific EVENT statements.

Use this syntax for the EXCLUDE statement:

```
EXCLUDE ABEND
        CODES=((abend[,reason])[,(abend[,reason])]...[,(abend[,reason])])
```

ABEND

This required positional parameter indicates that the exclusion criteria provided is for abend events.

CODES

This optional parameter indicates a list of specific abend codes that are to be ignored by the CAISDI/med abend event intercept. Each list element is comprised of a positional abend code followed by an optional positional abend reason code.

The positional abend code can be specified as a hexadecimal value for a system abend code such as X'222' or it can be specified as a decimal value for a user abend code such as 1222. The optional positional abend reason code can be specified as a decimal value, such as 12, or as a hexadecimal value, such as X'C'. Up to sixteen positional abend code / reason code combinations can be specified on a single EXCLUDE ABEND initialization statement. If more than sixteen abend codes are desired for exclusion from CAISDI/med processing, define additional EXCLUDE ABEND statements in the MEDPARMS data set.

The following is an example of the EXCLUDE control statement:

```
EXCLUDE  ABEND,
         CODES=((X'047'),
                (X'13E'),(X'33E'),
                (X'322'))
```

Defining General Recording Options for a Specific CA Product

This section describes how to use the PRODUCT control statement to define recording options for a specific CA product.

PRODUCT Control Statement

The PRODUCT statement is used to establish initialization values that both define the representation of a CA product and provide specific values for all CA Service Desk tickets generated on behalf of the defined CA product.

Note: At least one valid PRODUCT statement must be specified in the MEDPARMS initialization parameter data set; otherwise, CAISDI/med will not start.

Use this syntax for the PRODUCT statement:

```
PRODUCT product_name
        JOBNAME=name
        EVENT=(name[,name][,name] . . . [,name])
        AFFECTED_END_USER=(lastname [,firstname][,middlename])
        REPORTED_BY=(lastname [,firstname][,middlename])
        ASSIGNEE=(lastname [,firstname][,middlename])
        PRIORITY=value
        TEMPLATE=name
        SERVER=name
```

product_name

This required positional parameter defines the one- to 28-character CA product name mask. The product name mask is compared against requests made by callers of the CAISDI/med address space. The mask value can contain any alpha, numeric, special, or national character. If special characters are included in the mask, the mask should be enclosed in single quotes ('). If a single quote is included in the mask, it should be represented by two consecutive single quotes (''). Wild card support is provided. A percent sign (%) can be used to match any single character, while an asterisk (*) can be used to match all trailing characters inclusive of the asterisk.

JOBNAME

This optional parameter specifies a one- to 8-character job name mask that must be associated with the abend event for a match condition to occur. This parameter is useful if a given instance of a CA product executes in multiple address spaces in the same z/OS image. The mask value can contain any alpha, numeric, or national character. Wild card support is provided. A percent sign (%) can be used to match any single character, while an asterisk (*) can be used to match all trailing characters inclusive of the asterisk.

If this parameter is omitted, JOBNAME match is not performed for the named product.

EVENT

This optional parameter specifies a list of one to 16 qualifiers for the events being reported by the given product. A given type qualifier is a one- to eight- character mask that must be associated with the reported event for a match condition to occur. This parameter is useful if different events reported by a CA product are to have different and unique CA Service Desk attributes, or even to be directed to different CA Service Desk servers. The mask value can contain any alpha, numeric, or national character. Wild card support is provided. A percent sign (%) can be used to match any single character, while an asterisk (*) can be used to match all trailing characters inclusive of the asterisk.

If this parameter is omitted, event qualifier matching is not performed for the named product.

AFFECTED_END_USER

This parameter defines the CA Service Desk contact name to which newly created tickets are assigned. This contact name must be pre-defined on the CA Service Desk server.

A contact name is comprised of a required one- to 30-character last name, an optional one- to 30-character first name, and an optional one- to 30-character middle name. When either of the optional names is specified, the operand values should be separated by commas and the entire operand string should be enclosed in parentheses.

An AFFECTED_END_USER name is required by CA Service Desk to create a ticket. If this value is omitted, or is not pre-defined, a default CA Service Desk contact name of System_SD_User is used to create a new ticket.

REPORTED_BY

This parameter defines the CA Service Desk creator name established for newly created tickets. This creator name must be a contact name pre-defined on the CA Service Desk server. This name is comprised of a required one- to 30-character last name, an optional one- to 30-character first name, and an optional one- to 30-character middle name. When either of the optional names is specified, the operand values should be separated by commas and the entire operand string should be enclosed in parentheses.

A REPORTED_BY name is required by CA Service Desk to create a ticket. If this value is omitted, or is not pre-defined, a default CA Service Desk contact name of System_SD_User is used to create a new ticket.

ASSIGNEE

This parameter defines the CA Service Desk contact name that is displayed in the Assignee field of newly created tickets. This contact name must be predefined on the CA Service Desk server. The contact name is comprised of a required one- to 30-character last name, an optional one- to 30-character first name, and an optional one- to 30-character middle name. When either of the optional names is specified, the operand values should be separated by commas and the entire operand string should be enclosed in parentheses.

If a TEMPLATE name operand is supplied on the product statement but the ASSIGNEE name operand is omitted, the ASSIGNEE name defined in the CA Service Desk template is used, if defined.

If a TEMPLATE name operand is not supplied on the PRODUCT statement and the ASSIGNEE name operand is omitted or is not predefined, the ticket may be unassigned. CA Service Desk provides several mechanisms to select an assignee. For example, the CA Service Desk Assignee_set option can be set such that the REPORTED_BY contact name is used when the ASSIGNEE field has not been specified. CA Service Desk rules can also be applied to set the ASSIGNEE contact field in the ticket. See the CA Service Desk documentation for details on defining an ASSIGNEE contact when one is not supplied. If specified, the ASSIGNEE contact whose ID is used must be of the type "Analyst".

PRIORITY

This parameter defines one of the defined CA Service Desk priority values to be assigned for newly created tickets. Acceptable operand values are 1, 2, 3, 4, 5, and NONE.

If a TEMPLATE name operand is specified on the PRODUCT statement, but the PRIORITY value operand is omitted, no PRIORITY value is required as it is assumed that the appropriate priority is defined within the named CA Service Desk template.

If a TEMPLATE name operand is not supplied on the product statement, a PRIORITY value is required by CA Service Desk to create a ticket. If this value is omitted, or is not pre-defined, a default CA Service Desk priority of NONE is used to create a new ticket.

SERVER

This required parameter defines the one- to eight-character name of a SERVER initialization statement, which in turn, denotes the CA Service Desk sever will receive CA Service Desk tickets generated by this product.

Note: This is a required parameter and must be specified on the PRODUCT initialization statement.

TEMPLATE

Optional unless the CA Service Desk server is configured for ITIL compliance. The predefined template is required for ITIL configurations so that tickets open as incidents, not requests.

This parameter defines the one- to 64- character CA Service Desk template name to be used for newly created tickets. This template name must be pre-defined on the CA Service Desk server. A template can be used to assign ticket attributes such as Impact, Severity, Asset, and Group. Effective use of templates will categorize and assign tickets properly.

If this value is omitted, or is not pre-defined, no CA Service Desk template specification is used when creating a new ticket. In this case, valid operand values should be specified for the AFFECTED_END_USER, REPORTED_BY, and PRIORITY parameters.

If the TEMPLATE name parameter operand is specified, ASSIGNEE and PRIORITY are not required, and if omitted, no defaults are supplied for these respective parameter operands. In this case, it is assumed that these ticket attributes are defined within the named template.

If ASSIGNEE or PRIORITY are specified in addition to a TEMPLATE name, the ASSIGNEE or PRIORITY values will override their respective values defined in the CA Service Desk template.

The following is an example of the PRODUCT control statement:

```
PRODUCT CA-OPS/MVS,
        EVENT=(ABEND,PROBLEM),
        JOBNAME=OPSMAIN,
        AFFECTED_END_USER=System_OPSMVS_User,
        REPORTED_BY=System_OPSMVS_User,
        ASSIGNEE=System_OPSMVS_User,
        PRIORITY=2,
        TEMPLATE='CA-OPSMVS,Priority unassigned',
        SERVER=USDSRV01
```

Defining Product-Specific Abend Event Recording

This section describes how to use the EVENT control statement to define product-specific abend event reporting.

EVENT Control Statement

The EVENT statement is used to define CAISDI/med abend events for defined CA products that should be recorded as a CA Service Desk ticket.

Use this syntax for the EVENT statement:

```
EVENT ABEND
      COMPONENT=name,

EXCLUDE=((abend_code[,reason_code])[,(abend_code[,reason_code])]...[,(abend_code[
,reason_code])])
      JOBNAME=name
      PRODUCT=name
```

ABEND

This required positional parameter specifies that an event type of abend is being defined for a given CA product.

COMPONENT

This required parameter is a one- to 28-character name mask that is matched against an intercepted abend event to determine whether to capture and report the abend. Specifically, the component name mask is compared to the concatenation of the SDWACID and SDWASC fields in the CAISDI/med software LOGREC intercept. The mask value can contain any alpha, numeric, or national character. Wild card support is provided. A percent sign (%) can be used to match any single character, while an asterisk (*) can be used to match all trailing characters inclusive of the asterisk.

EXCLUDE

This optional parameter indicates a list of specific abend codes that are to be ignored by the CAISDI/med abend event intercept. Each list element is comprised of a positional abend code followed by an optional positional abend reason code.

The positional abend code can be specified as a hexadecimal value for a system abend code such as X'222' or it can be specified as a decimal value for a user abend code such as 1222. The optional positional abend reason code can be specified as a decimal value, such as 12, or as a hexadecimal value, such as X'C'. From one to sixteen positional abend code / reason code combinations can be specified.

JOBNAME

This optional parameter specifies a one- to eight-character job name mask that must be associated with the abend event for a match condition to occur. This parameter is useful if a given instance of a CA product executes in multiple address spaces in the same z/OS image. The mask value can contain any alpha, numeric, or national character. Wild card support is provided. A percent sign (%) can be used to match any single character, while an asterisk (*) can be used to match all trailing characters inclusive of the asterisk.

PRODUCT

This required parameter refers to a defined PRODUCT statement that is to be used to associate any captured CA product abend with the appropriate CA Service Desk ticket attributes.

The following is an example of the EVENT control statement:

```
EVENT ABEND,
      COMPONENT='IDMS*',
      PRODUCT=CA-IDMS
```

CAISDI/med Commands

This section describes CAISDI/med commands.

Starting the CAISDI/med Address Space

To start the CAISDI/med address space, enter the operating system START command. For example, issue the following command to start a copy of the CAISDI/med address space using all start parameter defaults:

```
START CASDIMED
```

The following series of messages indicates that the CAISDI/med address space completed initialization and is active:

```
CASD201I CA Service Desk Interface/mainframe event director
               CAISDI/med; r1.0     SP00D0Y10
               (c) 2007 CA International, Inc.
CASD242I CAISDI/med initialization in progress
CASD246I CAISDI/med initialization complete
CASD262I CAISDI/med is active
```

To start the CAISDI/med address space, using new common storage intercepts, issue the following START command:

```
START CASDIMED,REUSE=NO
```

Stopping the CAISDI/med Address Space

To stop the CAISDI/med address space, enter the operating system STOP command. For example, issue the following command to stop the copy of the CAISDI/med address space that was started using the preceding example:

```
STOP CASDIMED
```

Passing Commands to the CAISDI/med Address Space

The CAISDI/med address space supports operator commands through the standard operating system MODIFY command mechanism. For example, issue the following command to pass a command to the copy of the CAISDI/med address space that was started using the preceding example:

```
MODIFY CASDIMED,command text
```

Verify Status of CAISDI/med Address Space Communications

Follow these steps to verify the status of CAISDI/med address space communications.

1. After the CAISDI/med address space is active, issue a CAISDI/med DISPLAY TASK command and verify that all expected server tasks initialized successfully.

```
MODIFY MED,DISPLAY TASK
CASD351I DISPLAY TASKS:
  Name     Status       TCB      STE
-------- ------------ -------- --------
COMMAND  Active       007DF540 7F5456E0
EVENT    Active       007DF7D0 7F5444B0
REQUEST  Active       007DFB58 7F545080
USDSRV01 Active       007DFCF0 7F543448
CONTROL  Active       007DFE88 7F54D9B0
```

   The COMMAND, EVENT, REQUEST, and CONTROL tasks are general tasks that should always be monitored. The USDSRV01 task is the service task name as defined on the SERVER statement in the MEDPARMS member. For each SERVER statement defined in the MEDPARMS member, you should see a corresponding task in the DISPLAY TASK command output.

2. You also want to verify that logon access to the CAISDI/soap is completed successfully. Check the active CAISDI/med address space JOBLOG and look for a successful server logon message:

   ```
   CASD287I USDSRV01 CA Service Desk Login successful
   ```

   You should see one of these messages for each server you defined on the MEDPARMS member.

Displaying CAISDI/med INIT Statement Specifications

To display values associated with the CAISDI/med INIT statement, issue the following command:

```
MODIFY CASDIMED,DISPLAY INIT
```

Alternatively, the DISPLAY command can be shortened to the following:

```
MODIFY CASDIMED,D INIT
```

The DISPLAY INIT command lets you display the parameter specifications that are made on the CAISDI/med INIT statement.

Use this syntax for the DISPLAY INIT command:

```
DISPLAY INIT
```

INIT

This positional parameter displays settings from the CAISDI/med INIT statement.

The following is an example of the DISPLAY INIT command:

```
DISPLAY INIT
```

The command response for a DISPLAY INIT command might look similar to the following example:

```
MODIFY CASDIMED,D INIT
CASD342I DISPLAY INIT:
DEBUGMSG=NO   REUSE=YES  MEDNAME=SDIA
```

SET OPTION Command

The SET OPTION command lets you alter selected execution settings for the CAISDI/med address space.

Use this syntax for the SET OPTION command:

```
SET OPTION DEBUGMSG=[YES|NO]
```

OPTION

This positional parameter indicates that CAISDI/med address space level options are to be altered.

DEBUGMSG

This keyword parameter controls the production of additional diagnostic messages for various events within the CAISDI/med address space. Specify one of the following:

YES-Produce diagnostic messages.

NO-Do not produce diagnostic messages.

Default: NO

The following is an example of the SET OPTION command:

```
SET OPTION DEBUGMSG=YES
```

DISPLAY OPTIONS Command

The DISPLAY OPTIONS command lets you display parameter specifications that are in the CAISDI/med SET OPTION statement.

Use this syntax for the DISPLAY OPTIONS command:

```
DISPLAY OPTIONS
```

OPTIONS

This positional parameter displays settings from the CAISDI/med SET OPTION statement.

The following is an example of the DISPLAY OPTIONS command:

```
DISPLAY OPTIONS
```

Displaying CAISDI/med Product Interface Information

To display values associated with each product as defined to CAISDI/med, issue the following command:

```
MODIFY CASDIMED,DISPLAY PRODUCTS
```

Alternatively, the DISPLAY command can be shortened to the following:

```
MODIFY CASDIMED,D PRODUCTS
```

To display values associated with a specific product or group of products as defined to CAISDI/med, issue the following command:

```
MODIFY CASDIMED,DISPLAY PRODUCTS NAME=product_name
```

The *product_name* value can be the name of a specific product or a product name mask.

The DISPLAY PRODUCTS command lets you display the definitions created by the PRODUCT initialization statements defined in the MEDPARMS data set.

Use this syntax for the DISPLAY PRODUCTS command:

```
DISPLAY PRODUCTS [NAME=mask]
```

PRODUCTS

This positional parameter displays specifications created by the PRODUCT initialization statements.

NAME

The optional keyword parameter NAME can specify a one to twenty-eight character mask that should be used as a selection filter against the PRODUCT name when producing the command display. The mask value can contain any alpha, numeric, or national character. Wild cards are allowed: a percent sign (%) matches any single character and an asterisk (*) matches all trailing characters, inclusive of the asterisk.

The following is an example of the DISPLAY PRODUCTS command:

```
DISPLAY PRODUCTS NAME=CA*
```

Also consider the following example:

```
MODIFY CASDIMED,D PRODUCTS
CASD347I DISPLAY PRODUCTS

CA-MIM
 JOBNAME=MIMGR    SERVER=USDSRV01 PRIORITY=2
 TEMPLATE:
   MIMTEMP

 AFFECTED_END_USER:
   System_CA-MIM_User
 REPORTED_BY:
   System_CA-MIM_User
 ASSIGNEE:
   N/A
 EVENTS:
   ABEND
```

Displaying CAISDI/med EVENT ABEND Statement Specifications

To display the values that have been established with the EVENT ABEND statements, issue the following command:

```
MODIFY CASDIMED,DISPLAY EVENTS
```

Alternatively, the DISPLAY command can be shortened to the following:

```
MODIFY CASDIMED,D EVENTS
```

The DISPLAY EVENTS command lets you display the parameter specifications that are made on CAISDI/med EVENTS statements.

Use this syntax for this DISPLAY EVENTS command:

```
DISPLAY EVENTS
```

EVENTS

This positional parameter displays settings from the CAISDI/med EVENTS statements.

The following is an example of the DISPLAY EVENTS command:

```
DISPLAY EVENTS
```

Also consider the following example:

```
MODIFY CASDIMED,D EVENTS
CASD376I DISPLAY EVENTS:
ABENDS:
   PRODUCT=CA-MIM
   COMPONENT=MiMgr*
   JOBNAME=*
   EXCLUDE ABENDS:
      Code     Reason
      U0051  x'00000000'
      U0051  x'00000010'

   PRODUCT=CA-OPS/MVS
   COMPONENT=OPS*
   JOBNAME=*
   EXCLUDE ABENDS:
      No ABEND exclusion list


   PRODUCT=CA-SYSVIEW
   COMPONENT=GSVX*
   JOBNAME=*
   EXCLUDE ABENDS:
      No ABEND exclusion list

   PRODUCT=CA-VIEW
   COMPONENT=VIEW SAR*
   JOBNAME=SAR*
   EXCLUDE ABENDS:
      Code     Reason
      U0522
      S522
```

Displaying CAISDI/med EXCLUDE ABEND Statement Specifications

To display the values that have been established with the EXCLUDE ABEND statement, issue the following command:

```
MODIFY CASDIMED,DISPLAY EXCLUDE
```

Alternatively, the DISPLAY command can be shortened to the following:

```
MODIFY CASDIMED,D EXCLUDE
```

The DISPLAY EXCLUDE command lets you display the parameter specifications that are made on the CAISDI/med EXCLUDE statement.

Use this syntax for the DISPLAY EXCLUDE command:

```
DISPLAY EXCLUDE
```

EXCLUDE

This positional parameter displays the settings from the CAISDI/med EXCLUDE statement.

The following is an example of the DISPLAY EXCLUDE command:

```
DISPLAY EXCLUDE
```

Also consider the following example:

```
MODIFY CASDIMED,D EXCLUDE
CASD375I DISPLAY EXCLUDE:
ABENDS:
   Code     Reason
   S047
   S13E
   S33E
   S322
   U0051  x'00000010'
```

Displaying CAISDI/med Server Information

To display the values and status associated with each server as defined to CAISDI/med, issue the following command:

```
MODIFY CASDIMED,DISPLAY SERVERS
```

Alternatively, the DISPLAY command can be shortened to the following:

```
MODIFY CASDIMED,D SERVERS
```

To display the values and status associated with a specific server or group of servers as defined to CAISDI/med, issue the following command:

```
MODIFY CASDIMED,DISPLAY SERVERS NAME=server_name
```

The *server_name* value can be the name of a specific server or a server name mask.

The DISPLAY SERVERS command lets you display the definitions created by the SERVER initialization statements defined in the MEDPARMS data set. The display also shows the connection state of the associated CAISDI/med server interface task with its respective CA Service Desk server.

Use this syntax for the DISPLAY SERVERS command:

```
DISPLAY SERVERS [NAME=mask]
```

SERVERS

This positional parameter displays specifications created by the SERVER initialization statements.

NAME

The optional keyword parameter NAME can specify a one to twenty-eight character mask that should be used as a selection filter against the SERVER name when producing the command display. The mask value can contain any alpha, numeric, or national character. Wild cards are allowed: a percent sign (%) matches any single character and an asterisk (*) matches all trailing characters, inclusive of the asterisk.

The following is an example of the DISPLAY SERVERS command:

```
DISPLAY SERVERS NAME=USD*
```

Also consider the following example:

```
MODIFY CASDIMED,D SERVERS
CASD349I DISPLAY SERVERS:
USDSRV01  STATUS=Disconnected  HOSTNAME=SYS01
    TIMEOUT=20
   URL=http://usdserver.yourdomain.com
```

Displaying CAISDI/med Module Information

To display the values and status of CAISDI/med modules, issue the following command:

```
MODIFY CASDIMED,DISPLAY MODULES
```

Alternatively, the DISPLAY command can be shortened to the following:

```
MODIFY CASDIMED,D MODULES
```

The DISPLAY MODULES command lets you display the release and maintenance level of the CAISDI/med address space and well as detailed information about the CAISDI/med load modules loaded by the CAISDI/med address space.

Use this syntax for the DISPLAY MODULES command:

```
DISPLAY MODULES
```

MODULES

This positional parameter displays the CAISDI/med load module information.

The following is an example of the DISPLAY MODULES command:

```
DISPLAY MODULES
```

Also consider the following example:

```
MODIFY MED,DISPLAY MODULES
CASD344I DISPLAY MODULES:
CASD345I Release r1.0     SP00
CASD346I MEDMAIN   origin 07700000  length 00000000
         MEDMCT   07700000 00001000 CSD1001  04/03/05 13.33

CASD346I MEDCOMM   origin 068FA000  length 000060E0
         MEDCOMM  068FA000 00001000 CSD1001  04/03/05 13.33
         MEDAPI   068FB000 00000798 CSD1001  04/03/05 13.33
         MEDAPIPC 068FB798 00002830 CSD1001  04/03/05 13.33
         MEDRMGR  068FDFC8 00000398 CSD1001  04/03/05 13.33
         MEDLOGR  068FE360 00000760 CSD1001  04/03/05 13.33
         MEDUPOST 068FEAC0 00000C30 CSD1001  04/03/05 13.33
         MEDXMPST 068FF6F0 000009F0 CSD1001  04/03/05 13.33
```

Displaying CAISDI/med Task Information

The CAISDI/med address space is a multitasking address space. To display the names and current status of each of the CAISDI/med tasks, issue the following command:

```
MODIFY CASDIMED,DISPLAY TASKS
```

Alternatively, the DISPLAY command can be shortened to the following:

```
MODIFY CASDIMED,D TASKS
```

or

```
MODIFY CASDIMED,D TASK
```

The DISPLAY TASKS command lets you display the operating state of the various functional tasks within the CAISDI/med address space.

Use this syntax with the DISPLAY TASKS command:

```
DISPLAY TASKS
```

TASKS

This positional parameter displays the operating state of the CAISDI/med tasks.

The following is an example of the DISPLAY TASKS command:

```
DISPLAY TASKS
```

Also consider the following example:

```
MODIFY CASDIMED,DISPLAY TASKS
CASD351I DISPLAY TASKS:
  Name     Status       TCB      STE
-------- ------------ -------- --------
COMMAND  Active       006E0E88 7F5786E0
EVENT    Active       006E8160 7F5764B0
REQUEST  Active       006E83F0 7F578080
USDSRV01 Active       006E8680 7F5754F0
CONTROL  Active       006E8910 7F5799B0
```

Changing CAISDI/med SERVER Statement Specifications

To change values associated with a given CAISDI/med SERVER statement, issue the following command:

```
MODIFY CASDIMED,SET SERVER name keyword=value
```

The SET SERVER statement identifies that a given SERVER is being changed and *name* identifies the specific SERVER statement. The values to be changed are defined as keyword=value.

The following is an example of the SET SERVER command:

```
SET SERVER USDSRV01,
    HOSTNAME=XYZHOST1,
    TIMEOUT=30,
   URL_SYMBO>=WS2
```

For example, to change the CAISDI/soap server Service time-out value for a SERVER named, USDSRV01, issue the following command:

```
MODIFY CASDIMED,SET SERVER USDSRV01 TIMEOUT=60
```

The SET SERVER command lets you alter selected specification for an existing CAISDI/med server interface task. Any dynamic changes are reflected as long as the CAISDI/med address space remains active. To make permanent specification changes, the appropriate SERVER initialization statements in the MEDPARMS data set should be updated.

Note: Changing the specifications of a CAISDI/med server interface task will cause it to automatically disconnect from its current associated CA Service Desk server and immediately attempt to connect using the altered specifications.

Use this syntax for the SET SERVER command:

```
SET SERVER server_name
    HOSTNAME=name
    TIMEOUT=seconds
    URL_SYMBOL=name
```

SERVER

This positional parameter indicates that a CAISDI/med server interface task specification is to be altered.

server_name

This positional parameter is the name of the target CAISDI/med server interface task specifications to be altered. You can use the DISPLAY SERVERS command to obtain a list of the CAISDI/med server names.

HOSTNAME

This keyword parameter defines the CAICCI host name where the CA SOAP Client Service address space is executing.

TIMEOUT

This keyword parameter defines the number of seconds the CA SOAP Client Service address space should wait for a response from the CA Service Desk Web service before returning a timeout condition.

The time-out value can be a number between 10 seconds and 3600 seconds.

URL_SYMBOL

Optional. Used to identify the CA Service Desk server on which you want the tickets opened. If omitted, the SD_URL value from the CAISDI/soap server address space's Environmental Variables is used. You may select an alternate CA Service Desk server by assigning a “symbol” to URL_SYMBOL where URL_symbol is a defined Environmental Variable in the CAISDI/soap server and where its value is a URL identifying the CA Service Desk server. For example, if “URL_SYMBOL=WS2” then Environmental Variable “URL_WS2” must be defined to the CAISDI/soap server with a value which is the URL for the CA Service Desk server. See the CASOAPE Environmental Variables (see page 52).

The following is an example of the SET SERVER command:

```
SET SERVER USDSRV01,
    HOSTNAME=XYZHOST1,
    TIMEOUT=30,
    URL_SYMBOL=WS2
```

Also consider the following example:

```
CASD369I SET SERVER:
CASD371I SET SERVER processing complete
```

Starting a CAISDI/med Task

To start a previously stopped CAISDI/med task, issue the following command:

```
MODIFY CASDIMED,START name
```

Alternatively, the START command can be shortened to the following:

```
MODIFY CASDIMED,S name
```

The *name* operand is the name of the CAISDI/med task that is to be scheduled to start. The DISPLAY TASKS command can be used to identify all valid task names and their current status.

The START command lets you start a CAISDI/med task that was previously stopped by either a STOP task command or an execution time error.

Use this syntax for the START command:

```
START task_name
```

task_name

This positional parameter is the name of the target task that is to be started. You can use the DISPLAY TASKS command to obtain a list of the CAISDI/med task names.

Note: When a CAISDI/med server interface task starts, it will attempt to login to its associated CA Service Desk server.

The following is an example of the START command:

```
START USDSRV01
```

As another example, to start a previously stopped CAISDI/med task and to cause a log in to the associated CA Service Desk server, enter the following command:

```
MODIFY CASDIMED,START USDSRV01
```

Stopping a CAISDI/med Task

The STOP command lets you halt execution of a CAISDI/med task that is currently active. Note that certain CAISDI/med control related tasks cannot be stopped with a STOP command.

Note: Certain CAISDI/med control related tasks cannot be stopped with a STOP command.

Use this syntax for the STOP command:

```
STOP task_name
```

task_name

This positional parameter is the name of the target task that is to be halted. You can use the DISPLAY TASKS command to obtain a list of the CAISDI/med task names.

Note: When a CAISDI/med server interface task stops, it logs off from its associated CA Service Desk server.

The following is an example of the STOP command:

```
STOP USDSRV01
```

As another example, to stop a CAISDI/med task that is communicating with a CA Service Desk server, and cause a log off from that server, enter the following command:

```
MODIFY CASDIMED,STOP name
```

Alternatively, the STOP command can be shortened to the following:

```
MODIFY CASDIMED,P name
```

The *name* operand is the name of the CAISDI/med task that is to be scheduled to stop. The DISPLAY TASKS command can be used to identify all valid task names. Note the CONTROL task name and the COMMAND task name are not eligible to be stopped.

Forcing Termination of a CAISDI/med Task

Normally, a CAISDI/med task should be stopped using a STOP command (see page 136). However, in the event that a given target task is not responding to a STOP command, it is possible to force the termination of a task by issuing the KILL command.

The KILL command lets you force the termination of a CAISDI/med task that is non-responsive. The target CAISDI/med task is terminated with a User 222 abend.

Important! Use this command as a last resort. Termination of a CAISDI/med task may result in the failure of the CAISDI/med address space. Use this command only if you are prepared to stop and start the CAISDI/med address space.

Use this syntax for the KILL command:

```
KILL task_name
```

task_name

This positional parameter is the name of the target task that is to be terminated. You can use the DISPLAY TASKS command to obtain a list of the CAISDI/med task names.

Important! Terminating the CAISDI/med task named CONTROL causes the entire CAISDI/med address space to terminate.

The following is an example of the KILL command:

```
KILL USDSRV01
```

As another example, to terminate, with a U0222 abend, the CAISDI/med task that has not responded to a STOP command in the preceding example, enter the following command:

```
MODIFY CASDIMED,KILL USDSRV01
```

Alternatively, the KILL command can be shortened to the following:

```
MODIFY CASDIMED,K name
```

The *name* operand is the name of the CAISDI/med task that is to be scheduled to stop. The DISPLAY TASKS command can be used to identify all valid task names and their current status.

CAISDI/med Messages

CAISDI/med messages are described in the CA Common Services for z/OS *Message Reference Guide*, along with suggested actions.

How to Implement Standalone CAISDI/els

After installing standalone CAISDI/els using the procedures in the *Installation Guide*, follow this process to ensure you complete all the customization tasks for the standalone CAISDI/els component:

1. Customize and Stage the CASDIELS JCL PROC (see page 138)
2. Allocate the Event Library (see page 64)
3. Set Up CAISDI/els Startup Parameters (see page 62)
4. Configure the CAISDI/els Interface for Each CA Product (see page 65)
5. Run Standalone CAISDI/els (see page 138)

Customize and Stage the CAISDI/els JCL PROC

A sample CAISDI/els startup JCL procedure is contained in member CASDIELS of the CAI.CAW0PROC data set. To customize and stage the CAISDI/els JCL PROC, follow these steps.

1. Copy CAI.CAW0PROC(CASDIELS) to SYS2.PROCLIB(CASDIELS) or to any JCL PROCLIB that is automatically searched as part of z/OS START command processing.
2. Customize the //STEPLIB DD statement by specifying the name of the runtime APF-authorized load library into which the CAISDI/els executable load modules have been copied as described in the CA Common Services for z/OS *Installation Guide*. Or, if the CA Common Services CAILIB is in the LINKLIST, the STEPLIB can be removed.
3. Customize the //SYSIN DD statement by specifying the name of the CAISDI/els parameter data set that contains CAISDI/els initialization statements. By default, this would be your CAI.CAW0OPTN data set, but you may wish to copy the ELSSTART, ELSSTOP, ELSREMOV, ELSLIST, and ELSREQS members to a shared PARMLIB of your choosing. The ELSSTART member will be customized in a subsequent configuration step.

This step is complete when you have customized and staged the CAISDI/els JCL PROC to a searched z/OS JCL procedure library.

Run Standalone CAISDI/els

This section describes how to start and run the standalone CAISDI/els component.

This section contains the following topics:

Starting CAISDI/els

The CAISDI/els component is started using the CASDIELS procedure:

```
//CASDIELS     PROC PRODUCT=,
//             CMD=ELSSTART,
//             SYSOUT='X'
//*-----------------------------------------------
//CASDIELS EXEC PGM=CSDECNTL,PARM='&PRODUCT'
//STEPLIB  DD  DSN=CAI.CAW0LIB,DISP=SHR
//SYSIN    DD  DSN=CAISDI.PARMLIB(&CMD),DISP=SHR
//SYSPRINT DD  SYSOUT=&SYSOUT
//REPORT   DD  SYSOUT=&SYSOUT
```

The CAISDI parameter library contains several members designed to facilitate the starting and stopping of the CAISDI/els component. The ELSSTART member, which you previously configured, contains the control statements necessary to establish CAISDI/els. For example, the ELSSTART member might look like this:

```
DEFAULTS SD_AUTODEF=yes            Create contacts and assets
DEFINE PRODUCT=CA-1,EVENTLIB=cai.caievent,PRODCNTL=al0cntl,MASK=l0*
DEFINE PRODUCT=ALLOCATE,EVENTLIB=cai.caievent,PRODCNTL=xfgcntl,MASK=xf*
```

The CAISDI/soap address space, CASOAP, must be up and available for CAIDSDI/els to start up.

The following operator command can be issued from the system console or may be included in your IPL procedures to ensure the interface is always activated after an IPL:

```
S CASDIELS
```

The CAISDI/els Interface Controller, CSDECNTL, loads the event definitions from the various products into shared memory. For each product defined to this interface, CSDECNTL will create a Product Event Table. The collection of Product Event Tables and other memory-resident control blocks is referred to as the Event Control Structure. Once the Event Control Structure has been initialized, CAISDI/els is enabled. While the CAISDI/els Interface Controller is started with a console command, it is not a “started task” in the traditional sense. It establishes the interface and then terminates. If there were no problems, CSDECNTL completes with return code of zero. The interface is ready for use. If the return code is not zero, examine the processing log written to the SYSPRINT DD statement.

Stopping CAISDI/els

The following operator command can be issued from the system console to shut down CAISDI/els:

```
S CASDIELS,CMD=ELSSTOP
```

This invokes the CAISDI/els Interface Controller with a SHUTDOWN command. This removes the Event Control Structure and frees all of the ECSA storage except for the 64 byte Event Anchor Block (EAB) and a 36 byte PC routine stub. These are retained for the next time you start the CAISDI/els interface.

Runtime Options

The CAISDI/els component requires access to the CAISDI/soap component to communicate with CA Service Desk. To reduce your configuration tasks, CAISDI/els uses a discovery service of CAICCI to find all available CAISDI/soap z/OS SOAP clients in the environment. Each one is tested to see if the target CA Service Desk server can be accessed through it. The list of SOAP clients through which access can be made is then sorted by response time. The fastest connection is used as the primary connection; all others are queued up as failover candidates should the primary connection fail.

CAISDI/els Control Statements

The CAISDI/els Interface Controller, CSDECNTL, provides several control statements for use in managing the interface. These control statements give you the ability to shut down the interface, disable, or remove a specific product's interface, report on event activity, and report on the status of CA Service Desk tickets that are still in open status.

This section describes the CAISDI/els Interface Controller control statements.

General Considerations

This section discusses general considerations for using CAISDI/els control statements.

Using the &PARM Symbolic

The CAISDI/els Interface Controller allows the use of a special symbolic parameter named &PARM. In any control statement or comment, you can embed the &PARM symbolic parameter. For example:

```
//ENABLE   JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL,PARM='ca-1'
//STEPLIB  DD  DSN=CAI.CAW0LIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
ENABLE PRODUCT=&PARM
```

The value of the PARM= for the JCL EXEC statement sets the value of the &PARM symbolic to “CA-1”. The control statement “ENABLE PRODUCT=&PARM” is interpreted as if you had specified “ENABLE PRODUCT=CA-1” yourself. z/OS allows up to 100 characters to be passed via the PARM= parameter. Internally, the length of the control statement including the substitution of the &PARM value must *not* exceed 200 characters.

Syntax Notes

Note the following syntax conventions for command statements:

- Comments can be inserted in command statements by specifying an asterisk (*) in column 1 of the statement, followed by the comment. For example:

  ```
  *The following statement suppresses the password.
  ```

- All values must be enclosed in single or double quotes.

  Comments can be added on a command line following one blank space to the right of the parameter value:

  ```
  usd_id="ServiceDesk",     This is the Service Desk ID
  ```

- Parameters in command statements must be separated by commas.

  Use a plus sign (+) at the end of a command line to continue the command on the next line. For example:

  ```
  defaults usd_id='Service'+
              'Desk'
  ```

- All commands (define, default, and so on) and their parameters (for example, product= or prodname=) can be entered in any case. They are forced to upper case by the parser one word at a time as they are being evaluated.

  The following parameter values are forced to upper case:

PRODUCT	USD_AUTODEF
PRODCNTL	LANG
EVENTLIB	PRIORITY
MASK	EVENT
TRACE	

  Note: The values for the PRODUCT and LANG parameters are forced to upper case in *both* the command stream and from the product control member.

  All other parameter values are processed in mixed case. For the following parameters, the mixed case is crucial:

URL_URL	ASSIGNEE
USD_ID	REPORTED_BY
USD_PW	ASSET

AFFECTED_END_USER TEMPLATE

Defining Products to CAISDI/els

Use the DEFAULTS and DEFINE control statements to define CA products to CAISDI/els.

DEFAULTS Control Statement

The DEFAULTS control statement sets the default values for all subsequent DEFINE control statements in the current command stream. These defaults are required parameters for the DEFINE control statement.

Use this syntax for the DEFAULTS control statement:

```
DEFAULTS
    [URL_SYMBOL=symbol
    [USD_ID=user_id]
    [USD_PW=password]
    [USD_AUTODEF={YES|NO}]
    [LANG=lang]
```

URL_SYMBOL

Optional. Used to identify the CA Service Desk server on which you want the tickets opened. If omitted, the SD_URL value from the CAISDI/soap server address space's Environmental Variables is used. You may select an alternate CA Service Desk server by assigning a “symbol” to URL_SYMBOL where URL_symbol is a defined Environmental Variable in the CAISDI/soap server and where its value is a URL identifying the CA Service Desk server. For example, if “URL_SYMBOL=WS2” then Environmental Variable “URL_WS2” must be defined to the CAISDI/soap server with a value which is the URL for the CA Service Desk server. See the CASOAPE Environmental Variables (see page 52).

USD_ID

Optional (deprecated with elmds). The logon ID to be used in accessing CA Service Desk. If omitted, the ID will be taken from the CAISDI/soap SOAP client address space configuration file from the SD_USER variable.

USD_PW

Optional (deprecated with elmds). The password to be used with the CA Service Desk logon ID. If omitted, the password will be taken from the CAISDI/soap SOAP client address space configuration file from the SD_PASS variable.

USD_AUTODEF

Controls whether CAISDI/els will automatically define contacts and assets if they are not already defined.

Default: USD_AUTODEF=YES

LANG

A two-character code designating the language to be used in the summary and description texts of the CA Service Desk tickets opened by CAISDI/els. Valid codes are:

- DE (German)
- ES (Spanish)
- EN (English)
- FR (French)
- IT (Italian)
- DA (Danish)
- FI (Finnish)
- PT (Portuguese)
- SV (Swedish)
- NO (Norwegian)

Default: LANG=EN

In this example, the CAISDI/soap default values for the URL, ID, and password are being overridden. Normally, you would not want to do this, but it is beneficial for testing purposes.

```
DEFAULTS URL='http://usdserver.yourdomain.com:8080/axis/services/'+
'USD_WebServiceSoap?wsdl',
         USD_AUTODEF=yes,          Create contacts and assets
         USD_ID="ServiceDesk",     This is the Service Desk ID
         USD_PW="ServiceDesk"      This password will be suppressed
```

DEFINE Control Statement

The DEFINE control statement establishes CAISDI/els with a given CA product.

Use this syntax for the DEFINE control statement:

```
DEFINE
    PRODUCT=prodname
    EVENTLIB=dsn
    PRODCNTL=prod_ctl_mem
    URL_SYMBOL=symbol
    USD_ID=uid
    USD_PW=password
    [LANG=lang]
    [MASK=mask]
    [USD_AUTODEF={YES|NO}]
```

PRODUCT

Required. A product code that uniquely identifies a given CA product.

EVENTLIB

Required. The fully qualified partitioned data set name containing the product control member and the product event members. In most cases, this will be CAI.CAIEVENT.

PRODCNTL

Required. Identifies the name of the product control member.

MASK

Required for some products. A generic mask used to identify event member names where the event library may contain multiple products' event members or nonevent members.

URL_SYMBOL

Optional. Used to identify the CA Service Desk server on which you want the tickets opened. If omitted, the SD_URL value from the CAISDI/soap address space's Environmental Variables is used. You may select an alternate CA Service Desk server by assigning a “symbol” to URL_SYMBOL where URL_symbol is a defined Environmental Variable in the CAISDI/soap server and where its value is a URL identifying the CA Service Desk server. For example, if “URL_SYMBOL=WS2” then Environmental Variable “URL_WS2” must be defined to the CAISDI/soap server with a value which is the URL for the CA Service Desk server. See the CASOAPE Environmental Variables (see page 52).

USD_ID

Optional (deprecated with elmds). The logon ID to be used in accessing CA Service Desk. If omitted, the ID will be taken from the CAISDI/soap SOAP client address space configuration file from the SD_USER variable.

USD_PW

Optional (deprecated with elmds). The password to be used with the CA Service Desk logon ID. If omitted, the password will be taken from the CAISDI/soap SOAP client address space configuration file from the SD_PASS variable.

LANG

Optional. A code designating the language to be used in the summary and description texts of the CA Service Desk SD tickets opened by CAISDI/els. Valid codes are:

- DE (German)
- ES (Spanish)
- EN (English)
- FR (French)
- IT (Italian)
- DA (Danish)
- FI (Finnish)
- PT (Portuguese)
- SV (Swedish)
- NO (Norwegian)

Default: LANG=EN

USD_AUTODEF

Optional. Controls whether CAISDI/els will automatically define contacts and assets if they are not already defined.

Default: USD_AUTODEF=YES

Consider the following example:

```
//DEFINE   JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
DEFINE PRODUCT=CA-1,EVENTLIB=cai.caievent,PRODCNTL=al0cntl,MASK=l0*
DEFINE PRODUCT=ALLOCATE,EVENTLIB=casisd.eventlib,PRODCNTL=xfgcntl,MASK=xf*
```

Shutting Down CAISDI/els

To completely shut down CAISDI/els for all products, use the SHUTDOWN control statement.

SHUTDOWN Control Statement

The SHUTDOWN control statement shuts down CAISDI/els for all CA products.

Use this syntax for the SHUTDOWN command:

```
SHUTDOWN
```

Example

```
//SHUTDOWN JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
SHUTDOWN
```

Alternately, the following operator command can be issued from the system console to shut down CAISDI/els:

```
S CASDIELS,CMD=ELSSTOP
```

The ELSSTOP CAI.CAW0OPTN member contains a SHUTDOWN command.

Removing One Product Definition While Leaving the Rest Active

When CAISDI/els is first started, the DEFINE control statements create Product Event Tables in shared memory. If you want to remove one of those definitions while leaving the other product definitions active, you can use the either the REMOVE command or the ELSREMOV member.

REMOVE Control statement

The REMOVE control statement removes CAISDI/els for a given product.

Use this syntax for the REMOVE control statement:

```
REMOVE PRODUCT=prodname
```

PRODUCT

A product code that uniquely identifies a given CA product.

Consider the following example:

```
//REMOVE   JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
REMOVE PRODUCT=CA-1
```

ELSREMOV

The CAI.CAW0OPTN library contains the ELSREMOV member that can be used to remove a given product from the CAISDI/els interface using a console command. It contains a REMOVE command. For example, if you wanted to remove the CA 1 Tape Management product from the CAISDI/els interface while leaving the interface active for all other products using the interface, issue this console command:

```
S CASDIELS,CMD=ELSREMOV,PRODUCT=CA-1
```

Refreshing the Event Definitions After Maintenance

Occasionally, maintenance for a given product may affect one of its event library members. The event definitions for that product must be refreshed to pick up the changes applied by maintenance. The affected product must be removed and redefined to the CAISDI/els interface.

Consider the following example:

```
//REFRESH   JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
REMOVE PRODUCT=CA-1
DEFINE PRODUCT=CA-1,EVENTLIB=cai.caievent,PRODCNTL=al0cntl,MASK=l0*
```

Disabling and Reenabling CAISDI/els for One Product

When CAISDI/els is first started, the DEFINE control statements create Product Event Tables in shared memory. If you want to disable CAISDI/els for a given product, use the DISABLE command.

Disabling the interface for a given product is similar to removing the product definition: no CA Service Desk tickets will be opened on behalf of that product. However, the current event statistics are preserved in shared memory when you use the DISABLE command.

When you want to reactivate the CAISDI/els interface for a product that has been disabled, use the ENABLE command.

DISABLE Control statement

The DISABLE control statement disables CAISDI/els for a given product without removing it altogether. Event statistics and each event's current status is retained.

Use this syntax for the DISABLE control statement:

```
DISABLE PRODUCT=prodname ALL
```

PRODUCT

Specifies a product code that uniquely identifies a given CA product. The PRODUCT and ALL parameters are mutually exclusive.

ALL

Refers to all CA products configured with the CAISDI/els. The PRODUCT and ALL parameters are mutually exclusive.

Consider the following example:

```
//DISABLE   JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
DISABLE PRODUCT=CA-1
```

ENABLE Control statement

The ENABLE control statement reverses the effects of the DISABLE control statement.

Use this syntax for the ENABLE control statement:

```
ENABLE PRODUCT=prodname ALL
```

PRODUCT

A product code that uniquely identifies a given CA product. The PRODUCT and ALL parameters are mutually exclusive.

ALL

Refers to all CA products configured with CAISDI/els. The PRODUCT and ALL parameters are mutually exclusive.

Example

```
//ENABLE   JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
ENABLE PRODUCT=CA-1
```

Reports

This section describes CAISDI/els reports.

Reporting CAISDI/els Interface Event Statistics

A product's events are monitored in the Product Event Table of each respective product. You may report on the current status of the event tables using the LIST command.

Reporting Open CA Service Desk Tickets

The CAISDI/els interface can report on the current status of any CA Service Desk ticket that it opened. The LIST command, along with the REQUESTS parameter, instructs the CAISDI/els component to query CA Service Desk. The connection to CA Service Desk is based on the products that are currently defined to CAISDI/els. The information required to complete the connection to the appropriate CA Service Desk is stored in the Product Event Table.

LIST Control Statement

The LIST control statement generates reports based on the currently defined products, the status of CA Service Desk tickets that have been opened, and the status of the product events.

Use this syntax for the LIST control statement:

```
LIST
    [PRODUCT=prodname]
    [REQUESTS]
    [ASSET=asset_name]
    [REPORTED_BY=contact_name]
    [AFFECTED_END_USER=contact_name]
    [ASSIGNEE=contact_name]
```

PRODUCT

A product code that uniquely identifies a given CA product.

REQUESTS

Directs the LIST control statement to generate the REQUESTS report by querying the CA Service Desk for the currently open tickets that were opened through this interface. This includes not only the request tickets in non-ITIL configured servers, but also incident tickets in ITIL configured servers. The ASSET, REPORTED_BY, ASSIGNEE, and AFFECTED_END_USER parameters provide the selection criteria for that query. If no other parameters are specified, the AFFECTED_END_USER associated with each product's current definition is used as the selection criteria.

If the REQUESTS parameter is omitted, the LIST control statement will generate the EVENTS report, which lists the currently defined products and the status of all of their events.

ASSET

The name of the asset that was associated with the CA Service Desk ticket.

REPORTED_BY

The contact name that opened the CA Service Desk ticket. This is normally the REPORTED_BY contact name that is specified in the product's product control member.

AFFECTED_END_USER

The name of the CA Service Desk contact who is designated as the affected end user. This is normally the AFFECTED_END_USER contact name that is specified in the product's product control member.

ASSIGNEE

The CA Service Desk contact name who is currently assigned to the ticket. This is normally the ASSIGNEE contact name that is specified in the product's product control member.

ELSLIST

The CAI.CAW0OPTN library contains the ELSLIST member that can be used to generate a report on currently defined products, the current status of any CA ticket that has been opened and the status of product events. It contains a LIST command. For example, if you wanted to review the products that are currently defined to the CAISDI/els interface, issue this console command:

```
S CASDIELS,CMD=ELSLIST
```

List Reports

There are two types of reports that can be obtained with the LIST control statement:

If you omit the REQUESTS keyword, CAISDI/els produces a detailed EVENTS report. This is a list of all defined events and their current status. If you include the PRODUCT parameter, you can report on a single product's events; otherwise, all events for all products are reported.

If you include the REQUESTS keyword, CAISDI/els will query the CA Service Desk and produce a REQUESTS report. This is a report of all CA Service Desk tickets in open status that match your other criteria. You may list the open tickets for one product by including the PRODUCT parameter. If you omit the PRODUCT parameter, all tickets for each product currently defined will be reported, one product at a time. You may specify any one or all of ASSET, REPORTED_BY, AFFECTED_END_USER, and ASSIGNEE as additional selection criteria. Only tickets where all criteria are true will be reported. If you omit all criteria, the AFFECTED_END_USER contact from each product's definition is used just as if you had specified the AFFECTED_END_USER parameter.

This example reports the events for one product:

```
//LIST  JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
LIST PRODUCT=CA-1
```

To report on all events for all products using CAISDI/els, omit the PRODUCT parameter:

```
//LIST  JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAILIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
LIST
```

The previous example produces a report, written to the DD named REPORT, listing all events for each product, one product at a time, showing the event counts and the last time the event was triggered.

The following example lists CA Service Desk tickets for one product:

```
//LIST  JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
LIST REQUESTS,PRODUCT=CA-1
```

In this case, the CA Service Desk contact identified to CAISDI/els as the AFFECTED_END_USER is used as the selection criteria. Only tickets that are currently open are reported. The report is written to the DD named REPORT.

In the following example, the user has changed the AFFECTED_END_USER value so that there are tickets still open from two different affected end users. However, this user consistently opened all CA Service Desk tickets using the software asset named “CA 1 Tape Management.”

```
//LIST  JOB  '1234','Systems',CLASS=A
//CASDIELS EXEC PGM=CSDECNTL
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//SYSIN    DD  *
LIST REQUESTS,ASSET='CA 1 Tape Management'
```

The previous example lists all open tickets for the CA 1 Tape Management product and is written to the DD named REPORT. The report shows who is currently assigned to each ticket, when it was first opened, and when it was last updated.

ELSREQS

The CAI.CAW0OPTN library contains the ELSREQS member that can be used to generate the same report as the LIST REQUEST OPTION using a console command. It contains a LIST REQUEST command. For example, if you want to review all the open tickets that are currently defined to the CAISDI/els interface, issue the following console command:

```
S CASDIELS,CMD=ELSREQS
```

CAISDI/els Event Trigger Utility Commands

All CA products using CAISDI/els call it when a specific event occurs or when a condition arises. The call to CAISDI/els is referred to as “triggering the event”. The CAISDI/els Event Trigger Utility, CSDETRIG, may be used to trigger events manually. This utility is useful in verifying the network connections and ensuring all components are connecting properly. You may also incorporate this utility into production job streams where conditional step execution would invoke this utility and open CA Service Desk tickets based on condition codes. In this case, you may want to create your own events to augment the ones supplied by the product. Any events you create may be triggered only by the Event Trigger Utility.

Consider this sample JCL:

```
//TRIG   JOB  (account)'Systems',CLASS=A
//*
//EVENT   EXEC PGM=CSDETRIG,PARM='Test message text'
//STEPLIB  DD  DSN=CAI.CAW0LOAD,DISP=SHR
//SYSIN    DD  *
TRIGGER PRODUCT=CA-1,EVENT=L0E231,
    JOB=MYJOB,DSN=TEST.DATA.SET.NAME,MSG=”&PARM”
//SYSPRINT DD  SYSOUT=A
```

The Event Trigger Utility only has two control parameters:

PRODUCT=

The product code of the product whose event you wish to trigger.

EVENT=

The six character event code you wish to trigger. All event member names are eight characters; a six character code followed by a two character language code. This parameter identifies the code, not the member name.

Any other “parameters” seen by the Event Trigger Utility are taken to be symbolic parameters to be used in providing detailed event data. In the previous example, the “parameters” JOB, DSN, and MSG are set up as symbolic parameters. If the event L0E231 contains &JOB, &DSN, and &MSG embedded in the summary and/or description texts, then the values you assigned them will be substituted into the summary and/or description texts. Note the use of the PARM= on the JCL EXEC statement. Any value you specify on the PARM= may be passed directly into the control statements or comments. The &PARM symbolic will be replaced with the value you specified in the PARM= of the JCL EXEC statement before the control statement is parsed. In this example, the MSG symbolic parameter would be assigned the value “Test message text” and any place in the L0E231 event text where &MSG was found, this text string would then replace it. Note that the length of the control statement, including the substitution of the &PARM value, must *not* exceed 200 characters.

Using the Event Trigger Utility, you can trigger events of your own making. You simply construct a product control member and a set of product event members for your own purposes. Execute the Event Trigger Utility as a job step whenever you want to open a CA Service Desk ticket.

CAISDI/els Messages

CAISDI/els messages, suggested actions, and return codes are described in the CA Common Services for z/OS *Message Reference Guide*.

Index

&

C

D

E

I

J

K

L

M

P

R

S

T

U