CA General Transaction Server

User Guide

Version 12.0.00

CA Technologies Product References

This document references the following CA products:

- CA JCLCheck™ Workload Automation (CA JCLCheck)
- CA Scheduler® Job Management (CA Scheduler JM)
- CA Workload Automation CA 7® Edition (CA WA CA 7 Edition, formerly CA Workload Automation SE)
- CA Workload Automation Restart Option for z/OS Schedulers (formerly CA 11™ Workload Automation Restart and Tracking)

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Contents

Chapter 1: Introduction

The CA General Transaction Server (GTS) is a component used by different CA products to schedule transactions or services. These CA products that exploit CA GTS Services are called GTS clients, or clients for short. CA GTS handles both traditional transaction processing (short-lived requests) and the execution of long-running services. GTS clients make use of services as cross-memory callers, transactions/services inside the CA GTS address space and across z/OS image and Sysplex boundaries.

CA GTS supports z/OS Automatic Restart Management (ARM) controlled restart ability, thus maximizing availability. Clients interfacing with CA GTS can be dynamically started and stopped using unified operator commands. The application of maintenance for these products does not require a cycling of the CA GTS address space.

This section contains the following topics:

Components

One CA GTS server can support a variety of CA GTS clients running in the same address space, possibly interacting with each other. CA GTS provides the following components:

- A unified, dynamic console interface which dispatches client or server commands.
- A logger (I/O) component to provide coordination of dynamic allocation, open and close of data sets in the heavily sub-tasked GTS environment.
- An Automatic Restart Manager interface component that registers and deregisters with the z/OS service.
- A logical communications manager that manages data traffic between CA GTS systems including data blocking and unblocking, and encryption/decryption. Cryptology requires the presence of Integrated Cryptographic Services Facility (ICSF).
- A cross-memory interface processes requests that originate from outside the GTS address space, yet within the same operating system.
- An XCF communication manager that allows multiple CA GTS address spaces to route requests and responses between each other within the same Sysplex.
- A TCP/IP server for CA GTS-Plex configuration, that may be defined as peer-to-peer without a 'master' whose failure might require failover processing with associated service outage across the CA GTS-Plex.

Client Interaction

To allow a client to interact with a CA GTS product, a minimum of one CA GTS address space must be active on the same LPAR as the requestor. In general, no more than one CA GTS system is required, but multiple CA GTS address spaces can be started if situations require it.

Every CA GTS in a CA GTS-Plex (that is, a network of interconnected CA GTS systems) must have a unique identifier. This identifier serves internal purposes, but also allows the routing of requests to uniquely identifiable targets. Take care to provide the correct definition to avoid unintended interactions between multiple CA GTS address spaces on the same system. This includes separating XCF and TCP/IP information as well as the usage of separate CA GTS identifiers.

The ID of a CA GTS server can be specified either on the OS PARM on the EXEC PGM= statement of the JCL using the ID() keyword or the GSINIT*xx* member of the data set allocated under the CAGSPARM DD statement.

If during startup of a CA GTS, it is determined that a CA GTS is trying to join the network that has the name of a CA GTS already active in the network, an error message is produced. The CA GTS in question must be stopped, and the ID changed and restarted (or otherwise the cause of the problem investigated).

The requirement for a 'local' CA GTS exists even if the code to be invoked should execute on a 'remote system' (an LPAR other than the one that the client issues the request).

Chapter 2: Installation

This section contains the following topics:

System Requirements

This section describes the operating system and hardware requirements for CA GTS.

Operating System Requirements

CA GTS operates under any z/OS operating system supported by IBM. Individual GTS clients may require minimum operating system levels higher than CA GTS itself.

CA GTS requires the IBM Integrated Cryptographic Services Facility (ICSF) to support encrypted transmissions. If unavailable, encryption is not supported.

CA GTS executes in any JES2 or JES3 environment supported by IBM.

CA GTS executes APF-authorized in the standard problem program key (protect key 8).

CA GTS installation and maintenance requires SMP/E.

CA GTS can use Cross Memory Services (XMS), Cross-Systems Communication Facility (XCF), and/or TCP/IP. Use of these services is controlled through user-specified parameters.

With releases prior to IBM z/OS V1.8, address spaces using Cross-Memory Services (XMS) in a space-switching mode would have the address space controlling the XMS environment (CA GTS) become unusable until after the next IPL. With z/OS V1.8, IBM has allowed these address spaces to become reusable. If a site is executing an earlier release of z/OS, it should be aware of this XMS restriction. See the GSINIT parameter XMEM if CA GTS should be started without using XMS.

Hardware Requirements

CA GTS operates on an IBM mainframe that fully supports the z/OS operating system.

Memory Requirements

CA GTS generally requires a minimum of 4 MB below the 16 MB line, and a minimum of 32 MB above the 16 MB line. If a site supports REGION=0M executions to not limit the address space region size, CA GTS should execute with REGION=0M since it efficiently manages its storage.

CA GTS does not obtain any common storage areas, such as CSA/ECSA. It does use a name/token pair to anchor its common information.

Installation

Ensure that CA GTS has been installed with a supported workload automation package. Starting with CA GTS r11, CA GTS is installed when one of the following workload automation packages is installed:

- CA JCLCheck
- CA Scheduler
- CA WA CA 7 Edition (CA 7)
- CA WA Restart Option for z/OS Schedulers (CA 11)

The SMP/E function CD51C00 for r12.0 (CD51B00 for r11.0) is received, applied, and accepted into the same SMP/E environment as the workload automation product. Once the installation has been done, then all the CA GTS data sets are allocated and the configuration of CA GTS can proceed.

Chapter 3: Configuration and Security

This section contains the following topics:

Configuration

This section contains information on how to configure CA GTS to your installation's requirements.

JCL Procedure

Use the following sample JCL procedure to execute CA GTS.

We recommend running CA GTS as a started task (STC).

CA GTS uses cross-memory services. The address space will become unavailable upon shutdown unless XMEM(NO) is specified in the GSINIT00 initialization parameter. You should consult your GTS client, product documentation for the proper setting. If a client product requires GTS cross-memory services and XMEM(NO) is specified, the client will report that a GTS connection cannot be made.

A sample procedure for CA GTS can be found in your GTS SMP/E target library CD51PROC; the member name is CAGTS.

The following is a description of the sample procedure:

```
//*****************************************************************************
//***  Sample General Transaction Server (GTS) procedure
//***
//***  Change the following parameters
//***
//***   ID=               GTS ID (optional)
//***   GTSLOAD= GTS Load library
//***   GTSSUFF= GTS Suffix dataset
//***   GTSOPTN= GTS Options library
//***   TCPDATA= TCPIP Data library
//***
//*** See the CA General Transaction Server User Guide for more info.
//*****************************************************************************
//CAGTS PROC ID=,
//                GTSLOAD=CAI.CAGTS.CD51LOAD,     GTS LOAD LIBRARY
//                GTSSUFF=CAI.CAGTS.SUFFIX.DSN,   GTS SUFFIX DSN
//                GTSOPTN=CAI.CAGTS.CD51OPTN,     GTS OPTION LIBRARY
//                TCPDATA=TCPIP.TCPIP.DATA        TCPIP DATA LIBRARY
//STEP1 EXEC PGM=CAGSMNLD,PARM='ID(&ID)'
//STEPLIB DD DISP=SHR,DSN=&GTSLOAD
//CASUFFIX DD DISP=SHR,DSN=&GTSSUFF
//CAGSPARM DD DISP=SHR,DSN=&GTSOPTN
//SYSTCPD DD DISP=SHR,DSN=&TCPDATA
//GTSSNAP DD SYSOUT=*,DCB=(RECFM=F,LRECL=180)
//GTSPRINT DD SYSOUT=*,DCB=(RECFM=F,LRECL=180)
//GTSDEBUG DD SYSOUT=*,DCB=(RECFM=F,LRECL=180)
```

This procedure requires the identifier of the CA GTS to be supplied on the z/OS START command, for example:

```
START CAGTS,ID=SYS1
```

The STEPLIB DD statement points to an APF-authorized library containing the CA GTS code. This DD statement may be omitted if the code is located in an authorized LINKLIST data set.

CASUFFIX points to a data set containing the overridden member names for CAGSPARM. All CAGSPARM member names not explicitly overridden are presumed to have a suffix of '00'. For more information, see the CASUFFIX DD Statement (see page 13).

The CAGSPARM DD statement points to a partitioned data set that contains the member names controlling operation of the CA GTS system. For more information, see the CAGSPARM DD Statement (see page 14).

The SYSTCPD DD statement points to the data set containing system-wide TCP/IP definitions. This DD statement may be omitted from the JCL Procedure if there is only one TCP/IP system executing in the environment. If there are multiple TCP/IP environments, then code the SYSTCPD DD statement pointing to the definition library for the TCP/IP under which CA GTS should execute.

The GTSSNAP DD statement is only necessary when requesting a SNAP debugging session for one of the GTS clients. This should only be performed at the request of CA Support.

GTSPRINT and GTSDEBUG should point to standard SYSOUT. They hold informational and debug messages.

CASUFFIX DD Statement

The CASUFFIX DD statement points to a sequential data set or PDS member with record format fixed (RECFM=F), 80-byte records (LRECL=80), and any suitable blocking. This data set can contain any number of statements overriding suffixes for the CAGSPARM member to be used.

By switching the name of the data set allocated to the CASUFFIX DD statement, it is possible to specify a number of different suffixes. The proper switching of data sets might be efficiently implemented using system symbols.

One suggestion is to point the CASUFFIX DDname to a member in the options partitioned data set. Although the system symbolics inside GTS parameter library members are not supported, you could control the file where the CASUFFIX DDName points to using a system symbolic like &SYSCLONE in the started task JCL and then have one GTS parameter PDS that supports more than one system.

```
//CASUFFIX DD DISP=SHR,DSN=gts.parms(&SYSCLONE)
```

The syntax is as follows:

- An asterisk (*) in column 1 indicates a comment line
- Besides a control statement, a line may contain TSO style comments (/* xxx */)

Format:

NAME(*parm-name*) SUFFIX(*nn*)

NAME(*parm-name*)

Every line overrides exactly one member using the NAME() that defines the root of a member name.

SUFFIX(*nn*)

The SUFFIX() keyword specifies the two-character suffix of the member to use from the CAGSPARM library, and *nn* are any two alphanumeric characters including the characters @, # and %.

Example: Overriding the CA GTS initialization options, the XCF connectivity options, and the LOGGER options.

```
NAME(GSINIT) SUFFIX(20)     /*use GSINIT20  */
NAME(XCF) SUFFIX(20)        /*use GSXCF20   */
NAME(LOGGER) SUFFIX(20)     /*use LOGGER20  */
```

CAGSPARM DD Statement

The CAGSPARM DD statement is a PO or PDSE data set with record format fixed (RECFM=F), 80-byte records (LRECL=80), and any suitable blocking. The CAGSPARM DD statement contains members that contain all relevant parameters for the individual CA GTS components and any clients which may be operating under CA GTS control. All members of this data set, unless otherwise indicated by a client, employ a suffixing scheme. Every member in this data set ends in a two-character identifier which may be alphanumeric characters and the characters @, # and %.

The default for this identifier is '00' unless explicitly overridden (see CASUFFIX DD Statement). This approach allows the use of a single data set for many different CA GTS systems, each using its own suffix.

The CAGSPARM DD statement is required. If the statement is not present, a user-ABEND of 1000 at startup accompanies diagnostic messages.

CAGSPARM Parameters

The CAGSPARM parameters and their syntax rules are discussed in this section.

Samples of CAGSPARM members can be found in your GTS SMP/E target library CD51OPTN. The sample member names have a suffix of 00.

Statement Syntax Rules

All CAGSPARM members support TSO style syntax:

- The '+' continuation character is used if a particular statement will not fit on one line.
- An asterisk (*) in column one indicates a comment statement. Comments are ignored and may be inserted in the middle of a member.
- On individual lines, TSO style comments (/* comment */) are permitted.
- Whitespace is ignored.
- In general, multiple parameters may be coded on a single line.
- When multiple identical keywords are found, the last keyword will be the authoritative one.
- When a keyword value contains embedded blanks, it must be enclosed in quotes.

The following describes the different parameter members CA GTS uses, and their supported specifications. The parameter members assume a suffix of 00.

CLIENT00

This member defines all client products actively controlled by this CA GTS address space. The content of this member is either set up by the client's product installation procedure or documented by the client product. Each line contains exactly one product definition.

The CLIENT member supports the following keywords:

```
PRODUCT(  +
IDENTITY(xxxxxxxx)   +
MODULE(xxxxxxxx) )
```

IDENTITY

Designates the client identifier used by CA GTS; this can be one to eight alphanumeric characters. This identifier can later be used to start and stop the product dynamically. For information, see the chapter "Console Commands."

MODULE

States the program name containing the client startup code; this can be one to eight alphanumeric characters. CA GTS uses information contained here to load and initialize the client feature.

Example:

```
PRODUCT(IDENTITY(UF) MODULE(CAGSUFLD))
```

GSINIT00

All general definitions for the CA GTS server are set in this member. The following keywords may be specified on individual lines or combined; you can have one or more on a line. All parameters can either be specified in GSINIT00 or on the OS PARM (EXEC PGM=,PARM=);however, the length of the EXEC parameter is limited. It is suggested to specify the ID of the CA GTS on the EXEC parm and all other parameters in this member. Parameters on the OS PARM take precedence over definitions in this member.

The GSINIT member supports the following keywords:

ARMPOLICY(*xxxxxxxxxxxxxxx*)
ARMREGISTER(NO|YES)
DEBUG(YES|ON|NO|OFF)
ID(CAGENSRV|*xxxxxxxxxxxxxxxx*)
SECCLASS(DATASET|*xxxxxxxx*)
SECPREFIX(*xxx*)
XMEM(YES|NO)
XCF(YES|NO)
TCP(YES|NO)

ARMPOLICY

Specifies the name of the policy the MVS Automatic Restart Manager is to use; this can be one to 16 alphanumeric characters. For more information about Automatic Restart processing, see IBM's *MVS Sysplex Services Guide*.

ARMREGISTER

Indicates whether CA GTS registers with the MVS Restart Manager; an Automatic Restart Policy should also be specified if registration is requested. The default is NO.

DEBUG

Indicates whether CA GTS generates debugging messages; this should only be turned on at the request of CA Support. The default is NO or OFF.

ID

Defines the name of the CA GTS system; this can be one to 16 alphanumeric characters. Although this parameter can be coded on the OS parm, you can specify it here. This name must be unique across all CA GTS systems that connect either through XCF or TCP/IP.

A System Level Name/Token pair is created with the ID value. There is an IBM requirement that you don’t create a Name/Token pair that starts with the letters A through I.

You may want to consider including “GTS” along with the system name (since all GTS IDs must be unique). For example on system “SYSA” you could specify ID(SYSAGTS).

The default is CAGENSRV.

SECCLASS

Defines the SAF security class in which CA GTS may call for resource authorization; this can be one to eight alphanumeric characters. An example of such a class would be FACILITY. The default is DATASET.

SECPREFIX

Specifies the SAF resource name prefix; this can be zero to 40 alphanumeric characters. To separate CA GTS client class definitions from definitions for other components, specify a literal which will be pre-pended to any security resource names outlined by the CA GTS client products.

XMEM

Indicates whether CA GTS establishes a cross-memory environment; establishing a cross-memory environment causes the CA GTS address space (ASID) to become unavailable once it is terminated. You should check with the products that will be using GTS for their requirements for XMEM. For example, CA 11 requires XMEM(YES). The default is YES.

XCF

Indicates whether CA GTS establishes CA GTS to CA GTS connections using XCF; the default is YES.

TCP

Indicates whether CA GTS establishes CA GTS to CA GTS connections using TCPIP; the default is NO.

GSXCF00

The definitions for XCF connectivity are placed into this member.

```
DEBUG(YES|ON|NO|OFF)
GROUP(CAGSERV|xxxxxxxx)
```

DEBUG

Generates debugging messages if set to YES or ON. DEBUG should only be turned on at the request of CA Support. The default is NO or OFF.

GROUP

States the group name of the XCF group; this can be one to eight alphanumeric characters. This group communicates between CA GTS systems active on the same Sysplex with the same name. All CA GTS systems in the same Sysplex that should communicate with each other must use the same group name. You can specify any arbitrary name conforming to IBM XCF group naming conventions. The default is CAGSERV.

If you intend to separately activate multiple instances of CA GTS-Plex in the same Sysplex, you must choose distinct XCF group names. For example, this permits one CA GTS-Plex to support certain clients while another CA GTS-Plex supports a different set of clients. Starting two GTS' on the same LPAR using the same XCF group name will generate an error and operation refused.

LOGGER Command

The LOGGER command interacts with the unified logger which allows all components and clients in the CA GTS address space to safely write to multiple data sets without conflicts.

Before it can use a data set or SYSOUT class, the corresponding DD statement must be activated. This can either occur in the LOGGER00 member of CAGSPARM or using a console command.

The console command has the following format:

F *gts*,LOGGER *keyword*

DEBUG(YES|ON|NO|OFF)

Modifies the generation of debugging messages. This should only be turned on at the request of CA Support.

INFO

Displays a list of all data sets/SYSOUT, both opened and closed, defined to the CA GTS logger component. The logger is only able to write to DD statements that were defined to CA GTS, either through the CAGSPARM LOGGER*xx* member or a console command.

CLOSE DD(*xxxxxxxx*)

Closes a DD statement currently available to the logger. This allows for spinning off of SYSOUT data sets or the copying of data sets without the need of shutting down CA GTS system.

OPEN DD(*xxxxxxxx*)

Opens a DD statement currently defined and available to CA GTS. Use this command with the CLOSE command to handle spin-off of SYSOUT and dynamic allocation of new data sets.

FREE DD(*xxxxxxxx*)

Deallocates a DD statement and makes it available to allocation by other address spaces. DD statements that were freed may subsequently be allocated again.

ALLOC(*subparameters*)

Indicates a DD statement to be allocated to CA GTS using the DD(), DSN()|SYSOUT() and DISP() keywords. Using the TIMESTAMP() keyword, an optional timestamp may subsequently be inserted into the records written to this DD.

DD(*xxxxxxxx*)

Specifies the one-to-eight character DDNAME to use for allocating a data set or SYSOUT class

DSN(*xxx…xx*)

States the one-to-44 alphanumeric name of a data set. Use this to allocate to the DD statement using the DD keyword. The DSN and SYSOUT keywords are mutually exclusive.

SYSOUT(*x*)

Names the one-character output class to allocate to the DD statement specified using the DD keyword. The use of DSN and SYSOUT is mutually exclusive.

DISP(SHR|MOD|OLD)

Indicates the disposition to use for the allocation of the data set specified with the DSN keyword. Do not specify a disposition when allocating to a SYSOUT class.

TIMESTAMP(YES|ON|NO|OFF)

Determines whether to add a timestamp to every record output for this DD statement

```
F GTS,LOGGER INFO
CAGS02180I Logger dataset information for GTS CA11 follows
CAGS02190I DD: GTSSNAP , DSN: ERNRO03.CA7GS.STC58994.D0000103.?          , DSORG: PS  , DISP: MOD, TYPE: SYSOUT
CAGS02200I LRECL:   141, BLKSIZE:   141, RECFM: F  , Open
CAGS02190I DD: GTSPRINT, DSN: ERNRO03.CA7GS.STC58994.D0000101.?          , DSORG: PS  , DISP: MOD, TYPE: SYSOUT
CAGS02200I LRECL:   121, BLKSIZE:   121, RECFM: F  , Open
CAGS02190I DD: GTSDEBUG, DSN: ERNRO03.CA7GS.STC58994.D0000102.?          , DSORG: PS  , DISP: MOD, TYPE: SYSOUT
CAGS02200I LRECL:   121, BLKSIZE:   121, RECFM: F  , Open
```

TCPIPC00

The TCPIPC00 member describes the GTS-to-GTS TCP/IP client connectivity. Each GTS has a server and a client component. This member describes the client information, whereby the TCPIPS00 member describes the CA GTS Server information. Not all CA GTS applications require network connections. This includes the two applications that are included with CA GTS: ECHO and USSF. You should only define these connections if they are needed by the applications that will be using CA GTS.

The TCPIPC00 member supports the following keywords:

```
DEBUG(YES|NO|ON|OFF)
SERVER(xxxxxxxx)
Connection(PORT(nnnnn) +
HOSTNAME(url.name) +
HOSTADDR(ip.address) +
KEY(encryption_key) )
```

DEBUG

Indicates whether debugging messages should be generated; this should only be turned on at the request of CA Support. The default is NO or OFF.

SERVER

Specifies the name of the TCP/IP server to use; this can be one to eight characters. This keyword is only necessary if the code cannot otherwise determine which TCP/IP is available and the standard identifier 'TCPIP' is not active. To allow for the automatic recognition of the proper TCP/IP, allocate the SYSTCPD DD statement to the TCP/IP data set containing the system definitions. **Specifying SERVER() should only be a last recourse or necessary if GTS does not automatically pick the proper identifier.** Only one SERVER keyword is permitted in this member.

CONNECTION

Defines a connection to another CA GTS system; each CA GTS that you want to connect requires a CONNECTION keyword with subparameters to define the connection. Each CONNECTION keyword will contain one or more of the following subparamters:

Subparameters

HOSTNAME

Specifies the host name; this can be up to 255 alphanumeric characters. If the host names are not fully qualified, TCP/IP will complete the domain information if so configured. Code either the HOSTNAME or the HOSTADDR for a connection, but not both.

CA GTS will attempt to connect to this domain name using the specified port for the purposes of routing requests. Thus the port specification should point to another CA GTS system.

HOSTADDR

Specifies the IP address of another CA GTS system to which this CA GTS should connect; this can be seven-to-45 alphanumeric characters. For IPv4, the format of the IP address is *xxx.xxx.xxx.xxx*. For IPv6, the format of the IP address is *:xxxx:xxxx:xxxx:xxx.xxx.xxx.xxx*. CA GTS internally converts an IPv4 address to an IPv6 compatibility address. Code either HOSTADDR or HOSTNAME for each CA GTS connection, but not both.

PORT

Designates the numeric value of the port that CA GTS should use to connect to another CA GTS system; the value is four or five numerals in length and must be in the range 1024-65535. **The port specification must match the definition made in the TCPIPS00 member of the 'other' CA GTS system**. Ensure that this port does not conflict with any other product or the CA GTS systems cannot establish a successful connection. There is no default.

CA GTS attempts to connect to these addresses and ports for the purposes of routing requests.

KEY

Indicates the encryption key to use for this connection; encryption keys are 16 character hexadecimal. When specified, CA GTS uses the IBM Integrated Cryptographic Services Facility (ICSF) to encrypt traffic to this node. When not specified, or if ICSF is not available, the traffic between the nodes will not be encrypted.

TCPIPS00

This member defines GTS-to-GTS server information.

The TCPIPS00 member supports the following keywords:

DEBUG(YES|ON|NO|OFF)
PORT(49152|*nnnnn*)
SERVER(*xxxxxxxx*)

DEBUG

Indicates whether debugging messages should be generated; this should only be turned on at the request of CA Support.

PORT

Designates the numeric identifier of the port that CA GTS should listen on for incoming connection requests; the value is four or five numerals in length. The number must be in the range 1024-65535. This value should match the PORT() value in all TCPIPC00 members of the connecting 'other' GTS'. The default is 49152.

You may need to consult with your z/OS TCP/IP administrator to find a port number not in use. You may also need to setup a security system rule/definition to give GTS permission to use this PORT.

SERVER

Identifies the name of the TCP/IP server to use; this can be one to eight characters. This keyword is only necessary if the code cannot otherwise determine which TCP/IP is available and the standard identifier 'TCPIP' is not active. To allow for the automatic recognition of the proper TCP/IP, allocate the SYSTCPD DD statement to the TCP/IP data set containing the system definitions. **Specifying SERVER() should only be a last recourse or necessary if GTS does not automatically pick the proper identifier.**

Client Parameters

Client programs defined to CA GTS might also require parameters to be defined. Client parameter members are named:

PARM*cc*00

cc = the client ID defined in the CLIENT00 member. For details on these parameter members, see the client documentation.

The two clients distributed with CA GTS are described in the following chapters: "USSF Client of CA GTS (see page 27)" and "ECHO Client of CA GTS (see page 33)". For information on other products that use CA GTS, refer to their documentation.

SYSTCPD DD Statement

If your system uses multiple TCP/IP servers or nonstandard conventions for the naming of these servers, either a SYSTCPD DD statement or the specification of the SERVER() keywords in all TCP/IP related members in CAGSPARM is required.

If you decide simply to specify the SYSTCPD DD statement, the user ID associated with the CA GTS STC *must* be authorized to read this data set.

We recommend using the SYSTCPD DD statement instead of individual SERVER() keywords because it automatically adjusts to your configuration changes.

Security Information

CA GTS executes as a started task in the operating system. It also needs access to its parameters. For individual features executing as clients to CA GTS, consult the client documentation.

If a site validates started tasks executing in the operating system, then a started task ID must be set. Use the name of the CA GTS started task as the user ID. No special attributes are needed.

If you have chosen a security class name that is not already defined to the security system, you will need to add that class name to the security tables. For example, if you have coded GSINIT(SECCLASS(CAGTSSEC)) then CAGTSSEC will need to be added to the SAF tables (RACF Class Descriptor Table, CA eTrust ACF2 SAFPROT Table, or TSS Resource Descriptor Table).

Individual CA GTS clients may cause GTS to call the SAF interface. When security checks are made by the clients or by CA GTS on behalf of clients, CA GTS will use the resource class identified in the GSINIT SECCLASS parameter. Resource rules will use the GSINIT SECPREFIX pre-pended to the client's resource name. Allowing SECCLASS and SECPREFIX permits multiple CA GTS systems in the operating environment and ensures that the different systems are performing unique security calls.

Individual clients of CA GTS may have their own security consideration, so it is encouraged that you refer to the CA GTS client documentation for security needs.

Defining GTSPLEX Connections

If you are planning to use CA GTS clients that require systems to be interconnected, you need to define connections between the systems and provide unique names for each CA GTS.

Note: The two clients provided with CA GTS do not require interconnected CA GTS systems, they are: USSF Client and ECHO Client.

In addition to the information provided here, the product requiring the connections will provide additional information on connection requirements.

When using connected CA GTS systems, each CA GTS in a GTSPLEX must have a unique ID. The ID can be specified in the GSINIT00 member of the CAGSPARM library, or be specified as a PARM in CAGTS proc. If a duplicate name is found while connections are being established, a message will be issued and the connection will not be able to be completed.

There are two methods that can be used for connecting CA GTS systems together: XCF and TCP/IP.

XCF

The XCF connection works within a SYSPLEX and provides a very fast connection over a Channel connection or through a Coupling Facility. XCF connections are limited to a SYSPLEX's boundaries. Connections outside a SYSPLEX require defining TCP/IP connections. These are discussed in the next section.

The XCF connection is controlled by the XCF() keyword in the GSINIT00 CAGSPARM member.

XCF(ON) or XCF(YES) tries to establish or connect to an XCF GROUP. XCF(NO) or XCF(OFF) prevents the XCF connections from being active. XCF(YES) is the default. If you don't want these connections to be established, you must change the Option to OFF or NO.

The GSXCF00 CAGSPARM member contains overrides for the XCF connection. The GROUP keyword allows you to change the name of the XCF Group that will be used. The default name of CAGSERV will be used if the member is empty or does not exist. Specifying GROUP(name) allows you to change this name.

If you do not make any changes to the default installation, CA GTS will start the XCF connection using group CAGSERV.

TCP/IP

When CA GTS connections are needed outside of a SYSPLEX, you can enable TCP/IP. The TCP/IP connection is controlled by the TCP() keyword in the GSINIT00 CAGSPARM member.

TCP(ON) or TCP(YES) attempts to open a connection using TCP/IP. TCP(NO) or TCP(OFF) disables the TCP/IP connections. TCP(NO) is the default.

The TCP/IP interface uses a TCP server and a TCP client. They have their own parameter members that need to be changed.

TCPIPS00 defines the server parameters needed to open an inbound connection with TCP/IP. The PORT() keyword defines the port number used to accept connections from other CA GTS systems; the default is 49152. For more information, see TCPIPS00 (see page 21).

Example: Defining the server port number

```
PORT(49152)
```

TCPIPC00 defines the client parameters to indicate the server systems that will be connected. For each CA GTS you want to connect, you must specify a Connection keyword. Each Connection keyword defines the following:

- PORT() — the server port
- HOSTNAME() — the network URL for the server
- HOSTADDR() — the IP address for the server

Example: Defining the client parameters

```
CONNECTION(PORT(49152) HOSTNAME(system1.ca.com))
CONNECTION(PORT(22222) HOSTADDR(161.192.1.1))
```

The PORT() specified in the TCPIPC00 member must match the PORT() specified in the TCPIPS00 member on the system with which you want to establish a connection. To communicate between the systems, each system must have CONNECTION statements pointing to the other CA GTS with the correct port numbers.

CA GTS does support IPV6, and the full IPV6 format address is supported in the HOSTADDR keyword.

The parameters discussed here and some optional keywords not discussed are described in detail in this chapter, see TCPIPC00 (see page 19).

Example: Defining TCP connections between two GTS systems

```
System A hlq.CD51OPTN:
Member TCPIPS00:
    PORT(49152)
Member TCPIPC00:
CONNECTION(PORT(49152) HOSTNAME(System B's hostname))


System B hlq.CD51OPTN:
Member TCPIPS00:
PORT(49152)
Member TCPIPC00:
CONNECTION(PORT(49152) HOSTNAME(System A's hostname))
```

Chapter 4: USSF Client of CA GTS

The USS FILE TRACKING (USSF) utility provides file watching and tracking capabilities for files created under IBM's UNIX System Services (USS) file system. When a file is created or modified on the USS file system, USSF sends a user specified command to CA 7, CA Jobtrac, or CA Scheduler. This allows sites to control the execution of CA 7, CA Jobtrac, or CA Scheduler tasks based upon the file creation or modified event that occurred under UNIX System Services.

The USSF utility tracks file creation and updated events by scanning the USS file system for user specified files. USSF maintains a history of tracked file creation and updated events for the target files in the UFHIST data set for comparison of future file events. This is accomplished by storing the date/time stamp of the file when a match is found. If the date/time stamp of the file is updated, the specified command is issued to the specified scheduling system.

This section contains the following topics:

Installation Steps

Perform the following steps to install USSF:

1. If CA GTS has not been previously installed, you must install it. For more information, see the chapter Installation (see page 9).
2. Review, modify, and run the D51USSF member found in your GTS SMP/E target library CD51JCL.
3. Add the UFINIT and UFHIST DD statements to the CA GTS started task using the files created in the D51USSF job.
4. Add the member PARMUF*nn* to the CA GTS PARMLIB with the following line:

   ```
   DEBUG(NO)
   ```

5. Add the following statement to the CLIENT*nn* member of the CA GTS PARMLIB:

   ```
   PRODUCT(IDENTITY(UF) MODULE(CAGSUFLD))
   ```

6. Review the documentation on the configuration of the UFINIT file that will list the files to be "watched" or monitored by USSF.

Configuration

The USSF utility reads the control member in the UFINIT library to determine what files to track and what commands to issue. The control member name must be specified in the JCL that executes the USSF utility using the UFINIT DD statement.

There are two types of entries in the UFINIT file. One type is the specification for the file to be tracked with its associated command. The format of that entry is detailed later. The other type of entry is the keyword INTERVAL.

INTERVAL

The INTERVAL keyword defines a four-character numeric value that is used to indicate how often USSF wakes up to scan the file system. The value is in seconds and must be four characters in length. The default value for interval is 10 seconds.

Example: Wait Two Seconds

In the following example, USSF waits two seconds after completion of the previous scan to rescan the file system.

```
Interval=0002
```

Note: The interval specifies a frequency between scans. If USSF is monitoring 10 files, a scan-cycle will include a search for all 10 file names specified. The "interval" value is a wait time between cycles. If INTERVAL is specified more than once, the last entry will be in effect. The interval can be modified in real time.

More information:

Console Commands (see page 37)

File to be Tracked Entry Format

Use the following format to track files:

```
Filename,Scheduling System;Command
```

Filename

Defines a fully qualified path name to the file on the USS file system.

Example:

```
/u/user/joeuser/payroll.file
```

Note: The leading forward slash "/" is required in the path name.

Scheduling System

Defines the scheduling system; valid values are the following:

7; (or null)

Send the command to CA 7 (default)

J;

Send the command to CA Jobtrac

S;

Send the command to CA Scheduler

Command

Specifies any valid command format that is supported by the specified scheduling system. For CA Jobtrac or CA Scheduler, see the appropriate documentation.

For CA 7, commands are directed to U7SVC. For documentation on supported U7SVC commands, see the *CA 7 Interfaces Guide*.

For example, you can use:

TEST=YES

Directs the CA 7 command to the "Test" copy of CA 7 in compatibility mode. Use the TEST=YES keyword after the filename.

CA7=*aaaa*

In CA 7 r11 and above environments, the CA7= keyword can be used to direct the command to one of the eight possible CA 7 instances on a given LPAR. The instance name is a maximum of four characters and can be the alias name.

Note: The file to be tracked entry can be up to 5000 characters in length.

Example: UFINIT Member

```
Interval=0003

/u/user/system/payroll.file,7;TEST=YES;/login master;demand,job=jobtest1;/logoff
```

In the preceding example, USSF scans the file system for file name of payroll.file in the "/u/users/system/" directory every three seconds. If found, the CA 7 Login, Demand, and Logoff commands will be issued to the "test" copy of CA 7 through the U7SVC interface.

Security Considerations

The user ID associated with the batch job or started task that executes USSF must be authorized to read the UNIX System Services file system in order to report on file creations and updates under USS. This could be the entire file system from "root" down ("/") or a specific path to a directory where all of the files will be located for monitoring by USSF. Contact your Security Administrator for assistance with setting up the appropriate security access for USSF.

File Descriptions

USS uses the following files:

UFHIST

The UFHIST file maintains a list of files being watched and keeps a date/time stamp for each file to be used for comparison in subsequent searches of the file system.

The file has the following format:

```
RECFM=VB,LRECL=5000,BLKSIZE=27998
```

UFINIT

The UFINIT file contains the control member that specifies the files to be "tracked" or monitored. USSF will scan the search for an occurrence of each file listed in the control member, validate that the date and time are greater than the last occurrence, and if valid, issue the command to the specified CA z/OS scheduling solution.

The file has the following format:

```
RECFM=VB,LRECL=5000,BLKSIZE=27998
```

Processing

On startup, the UFINIT control member is processed and the UFHIST file is rebuilt to contain the new information. For files that were defined previously, the last modified date that was contained in the UFHIST file is retained.

Once initialization is complete, the files that are specified in the UFHIST file control member are checked to see if they have been modified, if so, the command associated with the modified file is issued to the specified CA scheduling system. Once all the files are checked, a wait for the interval is issued and then the file check is repeated. This continues until termination or a refresh is requested. Termination causes an orderly shutdown. Refresh reads the UFINIT file and rebuilds the UFHIST file. A refresh can be requested in real time.

The UFINIT file control member is read at startup, updates the UFHIST file, and then begins monitoring. If errors are found in the UFINIT file, a WTO will be issued showing the error and the line number in the control member where it occurred.

More information:

Console Commands (see page 37)

Chapter 5: ECHO Client of CA GTS

The Event Console Handler Option (ECHO) is a feature that lets sites receive messages from any CA Workload Automation product and display them on one or more z/OS consoles. This eliminates the need to log on to multiple platforms, terminals, or workstations to monitor job schedules. ECHO is an alternative for sites that are unable to run Event Console, the recommended solution for monitoring CA Workload Automation product output. This can occur when the site does not have any Windows-based systems available for Event Console.

No menus, panels, or screens are associated with ECHO. It consists of a single load module, CAGSECLD. CAGSECLD is part of the CA GTS component of CA Common Services for z/OS. It runs under the CA GTS long running job or started task. ECHO redisplays CA Workload Automation product messages using the CAGSEC600I Write-To-Operator (WTO) message.

This section contains the following topics:

Pre Job/STC Submission Setup

You should consider the following items before running ECHO:

- ECHO reroutes CA Workload Automation message displays to z/OS consoles based on the ROUTCDE and CONSNAME parameters passed to the job/STC. Depending on the number of jobs scheduled, thousands of messages could be routed to a console. We strongly recommend that sites not route these messages to their primary or alternate master consoles. Sites should also check the route code settings for these consoles to ensure they do not accept route code 11, the ECHO default route code if none is provided. Also, if the route codes specified on the ROUTCDE parameter are disabled for a particular console and you specify that console's name in the CONSNAME parameter, it receives rerouted messages.

  Use the DISPLAY CONSOLES,ACTIVE operator command to see the names and route codes for all consoles (master, alternate, MCS, SMCS). The DISPLAY EMCS,FULL,CN=*consname* operator command provides detailed information for extended MCS consoles.

- Sites may want to consider automating responses to the CAGSEC602E, CAGSEC620E, CAGSEC624E, and CAGSEC625E error messages. Any of these messages cause ECHO to terminate.
- Because ECHO accepts MODIFY commands through CA GTS, you should review who can issue this command. The MODIFY command can have five options:

  1. STOP
  2. DISPLAY
  3. ROUTCDE(*value*)
  4. CONSNAME(*value*)
  5. TRACE(*value*)

     These options are discussed in detail later. Follow your site's conventions in determining the level of security granularity you want to assign these options.
- To receive CA Workload Automation product messages, set up the CA Workload Automation product to route its messages as if they were going to a Unicenter Event Console. Each product has its own methodology for doing this. The following shows where you can find this information:

CA 7 Workload Automation

The Master Station Message Routing (MSMR) capability performs this function. For more information about Routing Master Station (Browse) Messages to an Event Console, see the *CA 7 Workload Automation Systems Programmer Guide*. You must specify the CCI receivers from which you want to receive messages in the MSGRCNTL DD data set used by MSMR.

CA Scheduler

Event Console information describes the methodology for CA Scheduler. See the *CA Scheduler Interfaces Guide*.

CA Jobtrac

Event Console logging information describes the methodology for CA Jobtrac. See the *CA Jobtrac Extended Scheduling Services Guide*.

AutoSys JM

For information, see the *AutoSys JM* documentation.

NSM

This is active by default.

Universal Job Management Agent

This is active by default; controlled by the Console Daemon Node setting in the Job Management Agent Configuration.

Installation Steps

Perform the following steps to install the Event Console Handler Option (ECHO):

1. If CA GTS has not been previously installed, you must install it. For information, see the chapter Installation (see page 9).
2. Add the ECTRACE DD statement to the CA GTS started task.
3. Add the member PARMEC*nn* to the CA GTS PARMLIB with the following lines:

   ```
   DEBUG(NO)
   TRACE(OFF)
   ROUTCDE(n,n,n,…)
   CONSNAME(xxxx)
   ```

4. Add the following statement to the CLIENT*nn* member of the CA GTS PARMLIB:

   ```
   PRODUCT(IDENTITY(EC) MODULE(CAGSECLD))
   ```

CA GTS PARMLIB Parameters and JCL

ECHO accepts three runtime parameters in any order: ROUTCDE, CONSNAME, and TRACE. No parameters are required. If no parameters are provided, default values are assigned. If the value supplied is incorrect, default values are assigned. The following topics describe the parameter defaults and descriptions.

ROUTCDE Parameter

The ROUTCDE parameter assigns routing codes to the WTOs issued by ECHO. These routing codes determine what consoles receive the WTOs. Valid ROUTCDE values are 1 to 28. Up to six two-byte route codes can be specified at a time. For example, to set the route codes to 1, 2, and 11 supply ROUTCDE(1,2,11) as the parameter. The default route code of 11 is not assigned if at least one route code is within the valid range. If you specified ROUTCDE(1,8,400) the route code settings would be 1 and 8, with 400 discarded.

Default: 11

CONSNAME Parameter

The CONSNAME parameter assigns a console name to the WTO issued by the program. The console matching the console name also receives the WTO. The CONSNAME must be 2 to 8 characters in length and consist of alphanumeric or national characters. No validation is performed to ensure the CONSNAME supplied actually exists and is active.

Default: blank

TRACE Parameter

The TRACE parameter causes a snap dump of the buffer received from CAICCI. It requires an ECTRACE DD statement. This parameter should only be used when requested by CA Support because it can result in thousands of lines written to the STC/JOB ECTRACE DD statement. Valid TRACE parameters are YES or ON and NO or OFF.

Default: NO

The following is an example of the DD statement to be added to the CA GTS JCL:

```
//ECTRACE DD SYSOUT=*
```

Chapter 6: Console Commands

This section contains the following topics:

CA GTS System Commands

Communication with an active CA GTS address space and its clients is permitted through the z/OS MODIFY command. The general syntax of the CA GTS modify commands, with the exception of HELP, is as follows:

F *gts,component keywords*

gts

The name of the CA GTS address space.

component

Represents a component or client name.

keywords

The action that the component should perform.

CA GTS always returns the response to a command to the requestor, which facilitates automation of commands using an automation product such as CA OPS/MVS.

Two layers of HELP are provided: All components (the first word of a command) can be listed by issuing a MODIFY command against the GTS address space with the HELP command verb. This will list all available command verbs. To list the components, enter the following:

```
F gts,HELP
```

```
CAGS01310I The following command verbs are supported:
CAGS01320I Comm. Manager      : COMM              (  4)
CAGS01320I TCPCLNT            : TCPCLNT           (  7)
CAGS01320I TCPSRVR            : TCPSRVR           (  6)
CAGS01320I Logger             : LOGGER            (  5)
CAGS01320I XCF Comm. Mgr.     : XCF               (  3)
CAGS01320I Command Manager    : SYSTEM            (  1)
CAGS01330I Issue HELP CMD(xxx) for more help on a specific command
```

Further information about the subcommands available through these components may be obtained by issuing the command

F gts,HELP COMMAND(*component*)

```
F GTS,HELP CMD(LOGGER)

CAGS01340I Help for command LOGGER follows:
CAGS01345I The following subcommands are available:
CAGS01345I
CAGS01345I DEBUG(ON|OFF):     Turn logger debugging on or off
CAGS01345I INFO:              Display detailed dataset/log information
CAGS01345I ALLOC DD(.dd.):    Allocate a DD statement for logger use
CAGS01345I FREE DD(.dd.):     Deallocate a DD statement from the logger
CAGS01345I OPEN DD(.dd.):     Open a logger DD statement
CAGS01345I CLOSE DD(.dd.):    Close logger DD statement
CAGS01345I
CAGS01345I The allocation related commands accept the following keywords:
CAGS01345I
CAGS01345I DSN(dsname):       Allocate a dataset with the given name
CAGS01345I SYSOUT(x):         Allocate a SYSOUT class
```

The following sections describe the individual GTS command verbs and the keywords supported by them.

COMM Command

The COMM command is the communications manager. It is responsible for maintaining information about connectivity between CA GTS systems. If this component is not aware of a connection, regardless of what information is shown by the physical XCF/TCP/IP routers, it cannot travel to and from this destination.

This command has the following format:

F *gts*,COMM *keyword*

DEBUG(YES|ON|NO|OFF)

Modifies the generation of debugging messages. This should only be turned on at the request of CA Support.

INFO

Displays all currently active connections between CA GTS systems. Multiple connections between GTS' may be shown using different communications methods. Two TCP/IP connections may exist, one a client-to-server, the other server-to-client.

```
F GTS,COMM INFO

CAGS01370I Host: CA31                , Sysplex: PLEXC1  , GTS: CA31    , connection type: TCP/IP - Server
CAGS01370I Host: CA31                , Sysplex: PLEXC1  , GTS: CA31    , connection type: TCP/IP - Client
CAGS01370I Host: CA11                , Sysplex: PLEXC1  , GTS: CA11    , connection type: XCF
```

TCPCLNT Command

The TCPCLNT command interacts with the GTS-to-GTS TCP/IP client supervisor. Underneath the client supervisor a separate task communicates with every other GTS. The supervisor manages and registers the established connections.

This command has the following format:

F *gts*,TCPCLNT *keywords*

DEBUG(YES|ON|NO|OFF)

Modifies the generation of debugging messages. This should only be turned on at the request of CA Support.

INFO

Displays all currently active connections between the physical TCP/IP client component and the other GTS servers. The result of this command should display one connection per GTS.

```
F GTS,TCPCLNT INFO

CAGS02910I TCP/IP GTS-to-GTS client info for GTS CA11 follows
CAGS02920I Local IP is: ::FFFF:141.202.65.11, Host: usilca11.ca.com, port 7590
CAGS02940I Connections follow:
CAGS02930I Remote IP is: ::FFFF:141.202.65.31, Host: CA31, port 7590
CAGS02950I System: CA31, CLONE ID: 31, managed by CA31
```

TCPSRVR Command

The TCPSRVR command communicates with the GTS-to-GTS server. The server component controls a number of slave tasks that are spawned during communication establishment.

This command has the following format:

F *gts*,TCPSRVR *keyword*

DEBUG(YES|ON|NO|OFF)

Modifies the generation of debugging messages. This should only be turned on at the request of CA Support.

INFO

Displays all currently active connections between the physical TCP/IP GTS-to-GTS server component and any clients. It should show one connection per connected GTS.

```
F GTS,TCPSRVR INFO

CAGS03500I TCP/IP GTS-to-GTS Server info for GTS CA11 follows
CAGS03510I Local IP is: ::FFFF:141.202.65.11, Host: usilca11.ca.com, port 12001
CAGS03530I Connections follow:
CAGS03520I Remote IP is: ::FFFF:141.202.65.31, Host: usilca31.ca.com, port 1844
CAGS03540I System: CA31, clone ID: 31, managed by CA31
```

LOGGER Command

The LOGGER command interacts with the unified logger which allows all components and clients in the CA GTS address space to safely write to multiple data sets without conflicts.

Before it can use a data set or SYSOUT class, the corresponding DD statement must be activated. This can either occur in the LOGGER00 member of CAGSPARM or using a console command.

The console command has the following format:

F *gts*,LOGGER *keyword*

DEBUG(YES|ON|NO|OFF)

Modifies the generation of debugging messages. This should only be turned on at the request of CA Support.

INFO

Displays a list of all data sets/SYSOUT, both opened and closed, defined to the CA GTS logger component. The logger is only able to write to DD statements that were defined to CA GTS, either through the CAGSPARM LOGGER*xx* member or a console command.

CLOSE DD(*xxxxxxxx*)

Closes a DD statement currently available to the logger. This allows for spinning off of SYSOUT data sets or the copying of data sets without the need of shutting down CA GTS system.

OPEN DD(*xxxxxxxx*)

Opens a DD statement currently defined and available to CA GTS. Use this command with the CLOSE command to handle spin-off of SYSOUT and dynamic allocation of new data sets.

FREE DD(*xxxxxxxx*)

Deallocates a DD statement and makes it available to allocation by other address spaces. DD statements that were freed may subsequently be allocated again.

ALLOC(*subparameters*)

Indicates a DD statement to be allocated to CA GTS using the DD(), DSN()|SYSOUT() and DISP() keywords. Using the TIMESTAMP() keyword, an optional timestamp may subsequently be inserted into the records written to this DD.

DD(*xxxxxxxx*)

Specifies the one-to-eight character DDNAME to use for allocating a data set or SYSOUT class

DSN(*xxx…xx*)

States the one-to-44 alphanumeric name of a data set. Use this to allocate to the DD statement using the DD keyword. The DSN and SYSOUT keywords are mutually exclusive.

SYSOUT(*x*)

Names the one-character output class to allocate to the DD statement specified using the DD keyword. The use of DSN and SYSOUT is mutually exclusive.

DISP(SHR|MOD|OLD)

Indicates the disposition to use for the allocation of the data set specified with the DSN keyword. Do not specify a disposition when allocating to a SYSOUT class.

TIMESTAMP(YES|ON|NO|OFF)

Determines whether to add a timestamp to every record output for this DD statement

```
F GTS,LOGGER INFO
CAGS02180I Logger dataset information for GTS CA11 follows
CAGS02190I DD: GTSSNAP , DSN: ERNRO03.CA7GS.STC58994.D0000103.?          , DSORG: PS  , DISP: MOD, TYPE: SYSOUT
CAGS02200I LRECL:   141, BLKSIZE:   141, RECFM: F  , Open
CAGS02190I DD: GTSPRINT, DSN: ERNRO03.CA7GS.STC58994.D0000101.?          , DSORG: PS  , DISP: MOD, TYPE: SYSOUT
CAGS02200I LRECL:   121, BLKSIZE:   121, RECFM: F  , Open
CAGS02190I DD: GTSDEBUG, DSN: ERNRO03.CA7GS.STC58994.D0000102.?          , DSORG: PS  , DISP: MOD, TYPE: SYSOUT
CAGS02200I LRECL:   121, BLKSIZE:   121, RECFM: F  , Open
```

XCF Command

The XCF command interacts with the physical XCF communications manager. The purpose of this component is to maintain Sysplex-wide awareness of active LPARs and GTS-to-GTS connection on the basis of XCF.

This command has the following format:

```
F gts,XCF keyword
```

DEBUG(YES|ON|NO|OFF)

Modifies the generation of debugging messages. This should only be turned on at the request of CA Support.

INFO

Displays information about all systems in the Sysplex, including if other GTS' are active on these systems. If another GTS is active on a system, additional information is shown.

An existing connection should imply connectivity to this system, but the only authoritative answer is available via the COMM INFO command.

Example:

```
F GTS,XCF INFO

CAGS03960I General system information for GTS CA11 follows
CAGS03980I Current system is identified as:
CAGS03990I XCF: CA11, hardware: MF01, LPAR: CA11, GRS: CA11, SMF: CA11 active on Sysplex PLEXC1, clone ID: 11
CAGS04000I Operating system release: SP7.1.7, FMID: JBB772S, IPL'd on 16:05:19 - 2006
CAGS04010I This GTS is active in XCF group ROLF1, as member M489, token 020005B0 00500001
CAGS04030I Sysplex-wide information:
CAGS04040I System: CA31    , clone ID: 31, status: ACTIVE, GTS connected: NO , managed by: N/A
CAGS04040I System: XE61    , clone ID: 61, status: ACTIVE, GTS connected: NO , managed by: N/A
CAGS04040I System: CA11    , clone ID: 11, status: ACTIVE, GTS connected: YES, managed by: CA11
CAGS04050I Member name: M489            , token: 020005B0 00500001, JOB/STC/TSU: CA7GS
CAGS04040I System: XAD1    , clone ID: D1, status: ACTIVE, GTS connected: NO , managed by: N/A
```

SYSTEM Command

The SYSTEM command interacts with the main GTS supervisor task.

This command has the following format:

F *gts*,SYSTEM *keyword*

DEBUG(YES|ON|NO|OFF)

Modifies the generation of debugging messages. This should only be turned on at the request of CA Support.

STOP(*xxxxxxxx*)

Dynamically terminates an active product, whose one-to-eight character name is represented by *xxxxxxxx*, under GTS. The parameter supplied to this keyword should match the IDENTITY keyword specified in the CLIENT00 CAGSPARM member.

START(IDENTITY() MODULE())

Dynamically starts a GTS client under GTS. This identity and module should correspond to the data specified in the CLIENT00 member of CAGSPARM. MODULE() and IDENTITY() are subfields of this command.

MODULE

Specifies the one-to-eight character name of a GTS client bootstrap module.

IDENTITY

States the one-to-eight character internal identity of a client started under GTS.

DUMP

Causes CA GTS to take an SVC dump and continue operation. This command is preferable to the z/OS console DUMP command as it ensures that all GTS data areas and data spaces are included in the generated data.

USSF Commands

USS File Tracking can accept console commands while running. The following examples contain commands that are accepted:

Note: *gts* is the job or started task name of the CA GTS server under which USSF is running

Example: Shut Down USSF

The following example shuts down USSF.

```
F gts,USSF SHUTDOWN
```

Example: Refresh USSF

The following example rereads the UFINIT file control member and rebuilds the UFHIST file.

```
F gts,USSF REFRESH
```

Example: Change Scan Interval

The following example changes the scan interval to *nnnn* seconds. The default is 10 seconds.

```
F gts,USSF INTERVAL(nnnn)
```

ECHO Commands

Users with the authority to issue the MODIFY (F) command can change the parameter settings, and stop or start the product. Only one parameter is processed with each modify command. Additional parameters are ignored. The need to issue most of these commands should be infrequent.

Note: *gts* is the job or started task name of the CA GTS server under which USSF is running.

Example: Stop the product

This example stops the product.

```
F gts,ECHO STOP
```

Example: Display current settings

This example displays the current parameter settings.

```
F gts,ECHO DISPLAY
```

Example: Change route code settings

This example changes the route code settings.

F *gts*,ECHO ROUTCDE(1,5,21)

```
F gts,ECHO ROUTCDE(1,142,21)
```

In the preceding example, 142 is not a valid route code. In this situation, route codes 1 and 21 are accepted and 142 is discarded. If any route code value supplied is valid, it is used. If none is valid, the default route code 11 is used. Be advised that when you modify the route code, previously existing settings are wiped out.

Example: Change console name

This example changes the console name.

```
F gts,ECHO CONSNAME(MASTER1)
```

Example: Reset console name

This example resets the console name.

```
F gts,ECHO CONSNAME()
```

This stops messages being routed to a previously set CONSNAME parameter value.

Example: Dump CAICCI buffer

This example dumps the CAICCI buffer passed to the ECHO module.

```
F gts,ECHO TRACE(YES)
```

You must have the ECTRACE DD statement in your CA GTS job/task. We do not recommend that you set TRACE to YES unless requested by CA Support because it can generate thousands of lines of sysout.

Example: Turn off tracing

This example turns tracing off.

```
F gts,ECHO TRACE(NO)
```