ALCS Online Communication Table Maintenance (OCTM) User Guide

November 2011

ALCS Technical Support
alcs@uk.ibm.com

Contents

Introduction

Management of the ALCS communication table was enhanced with the introduction of the Online Communication Table Maintenance (OCTM) facility in ALCS V2.3.1. OCTM allows communications network changes to be defined in the ALCS communication table via an online process, eliminating the requirement to perform an offline communications generation. The primary benefits of OCTM are:

- Provides the ALCS end users with the ability to directly update the online communication table
- Reduces the offline communications generation process to the management of only non-terminal resources
- Eliminates the need for communication table consolidations

ALCS customers who wish to use OCTM for managing the terminal and X.25 PVC resources are required to implement a Communications End User System (CEUS). The CEUS is a set of programs which provide a front-end to OCTM and which interface with OCTM via the COMTC Communication Table Update macro. The CEUS will be unique for each ALCS system because it must provide functionality that meets the specific operational and network requirements of the ALCS system (and the users of that system).

A sample CEUS application can be obtained from IBM via the ALCS web site. This CEUS application utilizes the ALCS 3270 mapping support, providing screen maps that enable the ALCS user to submit online updates to the ALCS communication table via OCTM. ALCS users can modify the IBM CEUS so that it fits their specific requirements. For example, most ALCS users maintain installation specific network data in the ALCS communication table user data area. The CEUS must contain the appropriate functionality for validating and managing that network data.

OCTM does not support the full range of communication resources that are supported in the ALCS communication table. OCTM is specifically designed to support all the terminal resource types, plus the X.25 PVC's which support terminal connectivity (X.25 PVC types 1, 6 and 7). The following lists the eight communication resource types that can be managed by OCTM:

- The MQSeries terminal resource type, provided for terminals that connect to ALCS through an MQSeries queue.
- The Websphere Application Server terminal resource type, provided for terminals that connect to ALCS through the ALCS WAS bridge.
- The NetView operator ID resource type, provided for NetView operator terminals.
- The Other System terminal resource type, provided for terminals that access the ALCS system through another system (who owns and controls the terminal).
- The TCP/IP ALC terminal resource type, provided for terminals that connect to ALCS through a TCP/IP network.
- The VTAM 3270 terminal resource type, provided for 3270 terminals.
- The VTAM/ALCI ALC terminal resource type, provided for ALC terminals that connect to ALCS through ALCI and VTAM.

- The X.25 permanent virtual circuit (PVC) resource type, providing connectivity for ALC terminals that communicate with ALCS through NPSI and VTAM.
- The X.25 ALC terminal resource type, provided for ALC terminals that connect to ALCS through NPSI, VTAM and an X.25 PVC.

Summary of OCTM Components

This section of the user guide provides a summary of the major components of OCTM.

COMTC Macro Interface

The COMTC Communications Table Update monitor-request macro provides the primary interface between the CEUS and OCTM. There are twelve different options available on the COMTC macro, providing the following primary functions:

- Query status of communication resource or group(s)
- Allocate and unallocate a communications group
- Submit communications change requests
- Activate communications change requests

When the CEUS submits *change requests* for the ALCS communication table via the COMTC macro, those change requests may be adding new communication resources, changing current resources or deleting obsolete resources. The CEUS will issue a separate COMTC macro for the submission of each individual *change request* to OCTM. The CEUS can submit just a single change request (with a single COMTC macro) or a group of change requests (with multiple COMTC macros). When a group of *change requests* is being submitted, the CEUS can allocate an OCTM ***communications group*** that will be used to manage and identify that group. When the CEUS is ready to activate the group of change requests in the online communication table, the complete group can be activated together. The COMTC macro interface therefore provides functionality that allows the CEUS to ***allocate*** a communications group, use that communications group name when submitting each individual change request, check the status of that group, activate the complete group in the online communication, and finally ***unallocate*** the communications group.

When an ALCS system has been migrated to the OCTM facility, all permanent changes to the ALCS communication table (those to be retained over an ALCS restart) for terminal and X.25 PVC resources (types 1/6/7) must be submitted to ALCS via the COMTC macro. These changes can no longer be submitted to ALCS via the offline communications generation process. The ALCS COMCC monitor-request macro (COMCC - Alter communication resource information) can still be used for temporary changes to the ALCS communication table for all resource types.

The following is a list of the twelve options provided on the COMTC macro:

1. COMTC QUERY - Query status of communication resource or group
2. COMTC GROUPS - Obtain status of communications groups
3. COMTC ALLOCATE - Allocate a new communications group
4. COMTC ADD - Add a new communication resource
5. COMTC REPLACE - Change a communication resource
6. COMTC DELETE - Delete a communication resource
7. COMTC CANCEL - Cancel a communication change request
8. COMTC LOAD - Load communication change requests
9. COMTC BACKOUT - Back out communication change requests
10. COMTC CONFIRM - Confirm communication change requests
11. COMTC COMMIT - Commit communication change requests
12. COMTC UNALLOCATE - Unallocate a communications group

In addition to the COMTC macro there are three data macros (DSECT's):

- The Communication Resource Definition DSECT (CT1TM)
- The Communications Resource and Group Information DSECT (CT2TM)
- The Communications Groups Information DSECT (CT3TM)

See "Overview of COMTC Macro Interface" on page 9 for more information on each of the twelve COMTC macro options.

OCTM Database

The OCTM database contains details of every terminal resource (except Prime and RO Cras), plus X.25 PVC resources, types 1/6/7. During restart of the ALCS system, the online communication table is built from data provided by the offline communications generation plus data from the OCTM database.

The OCTM database is comprised of two system fixed file areas. The first system fixed file contains the ***base communications area*** and the second system fixed file contains the ***update communications area***. The communication resources that are defined in the ***base communications area*** are loaded into the online communication table during ALCS restart (these are the confirmed/committed resources). The communication resources that are defined in the ***update communications area*** are the recent in-progress communication table changes (which are not loaded during ALCS restart). Records in the OCTM database are 4K in size (L3) with each physical record holding a number of logical records. Each logical record contains details of a single communication resource. Those details include both the communications system data and the user data.

The first ten physical records in the ***base communications area*** contain the OCTM control record and the OCTM communications group name records. The control record contains information that is vital for the successful operation of the OCTM facility. The communications group name records control the management of the OCTM communications groups.

ALCS dynamically obtains available space from the L3 allocatable pool when it builds the OCTM database and when it expands it. ALCS users must ensure that their L3 allocatable pool is large enough to accommodate the needs of the OCTM database. See "Initialize the OCTM database" on page 14 for information on the L3 record space requirements of OCTM.

OCTM Communication Generation Parameters

Three parameters are provided on the ALCS communication generation COMGEN macro for OCTM. The first of these three is the **OCTM=** parameter. This parameter should be coded as **OCTM=YES** if the ALCS system is using the OCTM facility. The other two COMGEN parameters are **CRIRANGE=** and **ORDRANGE=**.

When the migration to OCTM is complete, the management of the communication resources in the ALCS communication table will be shared between the OCTM facility and the offline communications generation. The allocation of CRI addresses and resource ordinal numbers to new communication resources must therefore be carefully managed so that duplicate usage of them does not occur. The **CRIRANGE=** and **ORDRANGE=** parameters provide a mechanism for managing the allocation of new CRI's and ordinals. These COMGEN parameters enable explicit ranges of CRI addresses and resource ordinal numbers to be defined for the exclusive use of the offline communications generation. OCTM will not use any CRI addresses or ordinal numbers that are defined in these

COMGEN parameters. We therefore recommend that the **CRIRANGE=** and **ORDRANGE=** parameters are included in the base communications generation deck so that duplicate usage of CRI addresses and resource ordinals can be avoided.

Details of these three COMGEN parameters is provided in ALCS Installation and Customization.

OCTM Policing

An OCTM policing function monitors the contents of the base and update communications areas on the OCTM database. OCTM policing is activated once each hour and its primary function is to monitor the status of communication resources and groups, and to identify periods of inactivity since the last COMTC macro was issued by the CEUS. If a period of inactivity is detected, OCTM policing sends a warning message to the RO Cras, indicating that further action is required. OCTM also activates an ECB-controlled installation-wide exit that can be used to perform additional functions. For example, the policing exit program can suppress transmission of the warning message and can issue a COMTC macro to activate the next COMTC function for the communication resource or group.

When the OCTM policing function is checking for periods of inactivity (for a communications resource or group) it does this for four COMTC macros:

- COMTC LOAD
- COMTC CONFIRM
- COMTC COMMIT
- COMTC UNALLOCATE

For example, if a COMTC CONFIRM has been issued for a communications group, but after 2 days the group has still not been committed (via COMTC COMMIT), the policing exit can decide to suppress the warning message and issue a COMTC COMMIT for that communications group. See “Using the OCTM policing exit” on page 40 for more information about OCTM policing.

ZOCTM Operator Commands

The OCTM facility provides operator commands for the ALCS operations personnel. The **ZOCTM - Control online communication table maintenance** command enables operations personnel to control the usage of OCTM, perform OCTM database backup/restore functions and to display OCTM status information. The ZOCTM command is also used to activate the OCTM database build function during the migration to OCTM. The commands are restricted to Prime Cras and Alternate Cras (AT1 to AT16). Details of each ZOCTM command can be found in ALCS Operations and Maintenance.

ALCS Communications Report Program

The offline Communications Report File Generator program (DXCCOMOL) provides a report of the communication resources defined in the offline communications generation. If the OCTM facility is being used, DXCCOMOL can include in its report details of the communication resources that are managed by OCTM. The DXCCOMOL communications report provides a valuable summary of all the communication resources defined in the ALCS communication table.

When the OCTM facility is being used, the input for DXCCOMOL is the communications generation load module(s), plus a sequential file that contains all the communication resources defined in the OCTM database. The **ZOCTM BACKUP** command is used to create this sequential file (it copies the contents of every record on the OCTM database to the sequential file).

DXCCOMOL produces an output file that contains details of every communication resource. That output file is in a format that allows the IEBPTPCH utility to print selective details of each resource. If required, a utility can be used to sort the contents of the output file prior to printing it with IEBPTPCH (for example, sort the resources into resource ordinal number sequence). Information about running DXCCOMOL can be found in ALCS Operation and Maintenance.

OCTM Offline Support Program

The communication resource definitions on the OCTM database are normally modified by the CEUS (via the COMTC macro), but they can also be modified by the OCTM Offline Support program, DXCCTMOL. This offline program can be used to modify the OCTM database when changes are needed in a large number of communication resources. This offline program can also be used to validate and implement new ranges of CRI addresses and resource ordinals (for the exclusive use of the offline communications generation) and it can be used to create a dataset that contains an ALCS generation COMDEF macro for every communication resource on the OCTM database. The primary input to this offline program is the OCTM sequential file which is created by the OCTM database backup function (see “Running the OCTM database backup function” on page 19). The following is a summary of the functionality provided by DXCCTMOL:

1. Modify and Delete communication resources on the OCTM database

   Use this program to modify multiple communication definitions on the OCTM database. For example, change the "test" status of communications resources, change the "initial" status, change and reformat the user data, etc. You can also use this program to delete a large number of communication resources that require removal from the OCTM database. Use this offline program in conjunction with the installation-wide exit program DXCUTMOL to apply these changes to the OCTM database. The DXCUTMOL exit program must identify the communication resources to be deleted (and mark them for deletion) and also identify the communication resources to be modified (and apply the required modifications). The exit program is activated by DXCCTMOL for every communications resource on the input OCTM sequential file. DXCCTMOL creates an output OCTM sequential file that contains all the communication resources that have **not** been deleted, and for those that have been modified, the updated communications definitions.

2. Validate and implement new CRIRANGE and ORDRANGE parameters

   Use this program to validate and implement modified ranges of CRI addresses and resource ordinal numbers in the communications generation **CRIRANGE=** and **ORDRANGE=** parameters (these parameters specify ranges of CRI addresses and ordinal numbers that are for the exclusive use of the offline communications generation). This offline program identifies the communication resources on the OCTM database that are using any of the CRI addresses or resource ordinals that are within these new ranges and outputs a report listing the names of those communication resources. This offline program can then be used to update the OCTM database with the modified ranges of CRI addresses and ordinal numbers and to delete resources that have either a CRI address or ordinal number that is within the modified ranges. DXCCTMOL creates an output OCTM sequential file that contains the new ranges of CRI's and ordinals (in the OCTM control record) and also contains the communications resources that have **not** been deleted.

3. Create communications generation COMDEF macros

Use this program to create a sequential dataset that contains a COMDEF macro for every communication resource in the base communications area of the OCTM database. This enables communication generation decks to be created for the communication resources managed by OCTM.

For the functions in the above list which create on output OCTM sequential file, the OCTM database restore function can be used (see "Running the OCTM database restore function" on page 20) to process that sequential file and rebuild the OCTM database on the ALCS system.

The OCTM Offline Support program is provided via APARs AQ80877 and AQ80878 on ALCS V2.3.1.

Overview of COMTC Macro Interface

The COMTC macro allows the CEUS to obtain information about communication resources, submit change requests for those resources and to permanently update the online communication table. The following describes the primary CEUS functions that use the COMTC macro.

1. Query status of communication resource or group(s)

   Query the status and obtain information about a communication resource or a communications group. This is provided by the COMTC QUERY macro.
   Query the status of all currently allocated communications groups. This is provided by the COMTC GROUPS macro.

2. Allocate and unallocate a communications group

   Allocate a communications group name for a batch of communication change requests and unallocate the group name when it is no longer required. This is provided by the COMTC ALLOCATE and COMTC UNALLOCATE macros. A communications group is required when multiple change requests are being submitted to ALCS via the COMTC macro, and those change requests must be grouped together under the name of a communications group.

3. Submit communications change requests

   Submit different types of change request for the communication resources. This is provided by the COMTC ADD, COMTC REPLACE and COMTC DELETE macros. This enables new communication resources to be added, current resources to be changed, and obsolete resources to be deleted. The COMTC CANCEL macro can be used to cancel a previous COMTC ADD, REPLACE or DELETE.

4. Activate communications change requests

   Activate a single change request or a group of change requests in the online communication table. This is provided by the COMTC LOAD, COMTC CONFIRM and COMTC COMMIT macros. When all the required change requests have been submitted to ALCS, they are activated in the online communication table via a three stage process. Those three stages are called load, confirm and commit. After a load or confirm, the change requests can be backed out of the online communication table by a COMTC BACKOUT macro.

There are twelve different COMTC actions that can be requested by the CEUS. The following provides a brief description of each of those twelve COMTC actions.

- QUERY

  Query status and obtain information about any communication resource. Also, query status and obtain information about a communications group, including the list of communication resources that belong to the group. See “COMTC QUERY – Query status of communication resource or group” on page 46 for details of the COMTC QUERY macro. The information is provided by COMTC QUERY in the Communications Resource and Group Information DSECT (CT2TM). See “CT2TM – Communications Resource and Group Information DSECT” on page 123 for details of this DSECT.

- GROUPS

Query status and obtain information about every currently allocated communications group. When you need to allocate a new communications group, you may wish to determine if the communications group name is already in use. The COMTC GROUPS macro allows you to obtain a list of the currently active groups and the status of each group. See "COMTC GROUPS – Obtain status of communications groups" on page 51 for details of the COMTC GROUPS macro. Basic information about each communications group, together with status information, is provided by COMTC GROUPS in the Communications Groups Information DSECT (CT3TM). See "CT3TM – Communications Groups Information DSECT" on page 128 for details of this DSECT.

- ALLOCATE

  Allocate a new communications group. When you have a group of change requests to submit, use COMTC ALLOCATE to allocate a group name for them. It is easier to manage a group of changes when a group name has been allocated for them. When change requests have been submitted (via COMTC) for the communications group, a single COMTC LOAD macro will load all the change requests into the online communication table. See "COMTC ALLOCATE – Allocate a new communications group" on page 54 for details of the COMTC ALLOCATE macro.

- ADD

  Add a new communication resource. When a new communications resource is being added, full details of the resource must be provided, including its CRN, the system data and user data. Details of the resource are provided on COMTC ADD macro parameters and via the Communication Resource Definition DSECT, CT1TM. See "COMTC ADD – Add a new communication resource" on page 57 and "CT1TM – Communication Resource Definition DSECT" on page 96. If a group of new resources are being added, the name of the communications group must also be provided. Details of the new resource provided by COMTC ADD are written to the OCTM database.

- REPLACE

  Change a communication resource. The current system and user data for a communication resource can be changed by COMTC REPLACE. The CRN or CRI address is used to identify the resource being changed. Details of the changes are provided in a storage block formatted by the Communication Resource Definition DSECT, CT1TM. See "COMTC REPLACE – Change a communication resource" on page 64 and "CT1TM – Communication Resource Definition DSECT" on page 96. If a group of resources are being changed, the name of the communications group must also be provided. The information provided by COMTC REPLACE is written to the OCTM database.

- DELETE

  Delete a communication resource. The communication resource being deleted must currently exist on the OCTM database. The CRN or CRI address is used to identify the resource being deleted. If a group of resources are being deleted, the name of the communications group must also be provided. The communication resource being deleted is flagged as deleted in the OCTM database. See "COMTC DELETE – Delete a communication resource" on page 69 for details of the COMTC DELETE macro.

- CANCEL

  Cancel a communication change request. Communication change requests are submitted by COMTC ADD, REPLACE and DELETE macros. After submitting the change requests, if one is incorrect (for example, the wrong CRN is used in a COMTC DELETE), the COMTC CANCEL can be used to cancel the change request. If the change request being cancelled belongs to a communications group, the name of the communications group must be provided. Only those change requests that have not yet been loaded via COMTC LOAD can be cancelled. The cancel deletes the change request from the OCTM database. See "COMTC CANCEL – Cancel a communication change request" on page 73 for details of the COMTC CANCEL macro.

- LOAD

  Load communications change requests in the online communication table. Change requests submitted via COMTC ADD, REPLACE and DELETE macros are loaded into the online communication table. For example, if six new communication resources had been added in a group, those six resources are now added to the online communication table. If two communication resources had been deleted, those resources are now removed from the online communication table. A single communication change request can be loaded or a group of change requests can be loaded. See "COMTC LOAD – Load communication change requests" on page 76 for details of the COMTC LOAD macro.

- BACKOUT

  Back out communications change requests from the online communication table. A COMTC BACKOUT can be used after a COMTC LOAD or a COMTC CONFIRM. If a COMTC LOAD was used to load a group of change requests, the COMTC BACKOUT will back out (remove) that group of change requests. For example, if a group of change requests included some COMTC DELETE requests, the COMTC BACKOUT reinstates the deleted resources in the online communication table. The change requests that have been backed out still reside on the OCTM database. They can be subsequently removed from the OCTM database (if they are no longer required) by a COMTC CANCEL. See "COMTC BACKOUT – Back out communication change requests" on page 81 for details of the COMTC BACKOUT macro.

- CONFIRM

  Confirm communications change requests. Communication change requests should be confirmed after they have been successfully loaded. When communication change requests have been confirmed, they are retained in the online communication table over an ALCS restart. The logical records are flagged as confirmed in the OCTM database. A single communication change request can be confirmed or a group of change requests can be confirmed. See "COMTC CONFIRM – Confirm communication change requests" on page 85 for details of the COMTC CONFIRM macro.

- COMMIT

  Commit communications change requests. Communication change requests should be committed after they have been successfully confirmed. When communication change requests have been committed, they reside permanently in the OCTM database and the online communication table. When the change requests belong to a communications group, the communications resources that were added, updated or deleted by those change requests are no longer locked to the communications group. See "COMTC COMMIT – Commit communication change requests" on page 88 for details of the COMTC COMMIT macro.

- UNALLOCATE

  Unallocate a communications group. When all change requests for a communications group have been committed, the communications group name should be deleted. The

COMTC UNALLOCATE deletes the group name from the OCTM database. See “COMTC UNALLOCATE – Unallocate a communications group” on page 91 for details of the COMTC UNALLOCATE macro.

Migrating to OCTM

Specific tasks must be performed for the migration of the terminal and X.25 PVC resources (types 1/6/7) from the offline communication generation process to the OCTM facility. The offline communication generation process will continue to be used for the management of non-terminal communication resources. None of the facilities provided in ALCS for managing the communications resources have been made obsolete by OCTM. For example, if the extended CDS support (CDS2) is being used for managing the communication load list, usage of this can continue.

The primary tasks that must be performed for the OCTM migration are:

- Update the ALCS communications generation
- Initialize the OCTM database
- Build the OCTM database
- Activate OCTM

Before commencing the migration process, determine if your ALCS DASD generation has been updated (and loaded via ZDASD LOAD) since the cutover to ALCS V2.3.1. If the ALCS DASD generation has not been updated since the cutover, determine if it was updated on your ALCS V2.2.1 system after ALCS maintenance tape 6 was applied (more specifically, after ALCS V2.2.1 PTF UQ69454 (APAR AQ57272) was applied). If there have been no updates to the DASD generation, ensure that it is updated before you initialize the ALCS database for the OCTM migration. You can update your ALCS DASD generation by adding the pool record ID of AC10 to the L3ST pool, and by using ZDASD LOAD, CONFIRM and COMMIT to fully activate that update. If you are unsure about the need for this DASD generation update, we would recommend that you do it. If this update is required and you fail to do it, ALCS will abend with a CTL-EF5 dump when it is restarted for the OCTM database initialization.

Update the ALCS communications generation

The ALCS communications generation COMGEN macro must be updated. Code the **OCTM=YES** parameter on the COMGEN macro to specify that the ALCS system will be using the OCTM facility. Apply this update to the base communication generation deck and restart the ALCS system with this updated communications generation.

Perform a consolidation of your communications generation COMDFLT and COMDEF macros. Many ALCS users have large terminal networks and therefore maintain their base COMDFLT and COMDEF macros in multiple generation decks. Those ALCS users may also have many small generation decks which have been loaded online via the ZACOM LOAD command. The COMDFLT and COMDEF macros in those small generation decks must be consolidated into the base generation decks before the OCTM database is initialized. Build a new communication configuration load list that contains a list of the new consolidated generation decks. If the communications CDS (CDS2) is being used, load this new load list via the ZACOM LOAD LIST command (then confirm and commit this load list). Also, update the name of the communication configuration load list in the PARM field on the EXEC statement that starts the ALCS job. Do this even if you are using CDS2. When the OCTM database is built (see “Build the OCTM database” on page 15), ALCS uses the communication configuration load list defined in the EXEC statement PARM field, it does not use the load list on CDS2.

There are two further COMGEN generation parameters that should be coded *after* the OCTM migration is complete. These are the **CRIRANGE=** and **ORDRANGE=** parameters. These parameters ensure that CRI addresses and ordinal numbers used by the communications generation process do not conflict with those used by the OCTM facility. Conflicts could occur when new communication resources are being added to ALCS via the offline communications generation and via OCTM at the same time. Refer to ALCS Installation and Customization for a description of these COMGEN parameters.
If these parameters are coded *before* the OCTM migration, they will be used by the OCTM database build function (see "Build the OCTM database" on page 15) to exclude all communication resources whose CRI address or ordinal number is within the ranges defined by these parameters.

Soon after the OCTM migration is complete, the offline communication generation deck(s) should be condensed by removing all generation macros (COMDFLT and COMDEF) that define resources which are now being managed by the OCTM facility. During this condense process, the COMGEN **CRIRANGE=** and **ORDRANGE=** parameters should be coded to specify the ranges of CRI's and ordinal's that will be used by the non-terminal resources in the communication generation deck. See "Condensing the Communication Generation Decks" on page 17 for further information on this.

Initialize the OCTM database

The initialization of the OCTM database is performed automatically by ALCS. When the ALCS system is restarted with a base communications generation load module that includes an updated **OCTM=** parameter (updated from OCTM=NO to OCTM=YES on the COMGEN macro) it checks for the existence of an OCTM database. If none exists, it performs the OCTM database initialization function. During this ALCS restart, the programs that comprise the CEUS application could be loaded. Before ALCS initializes the OCTM database, it will build the online communication table using the current ALCS communications load list (on the EXEC statement PARM parameter or on the communications CDS2). The following describes the OCTM database initialization process.

- **Allocate the OCTM system fixed files**

  The OCTM database is comprised of two system fixed file areas, each containing L3 sized records (4000-byte records). The first system fixed file contains the ***base communications area*** and the second contains the ***update communications area***. ALCS allocates the initial records required for these two system fixed files. These initial records are taken from the L3 allocatable pool, and ALCS verifies that there are at least 500 available records remaining in the L3 allocatable pool before it performs the allocation process (if there are less than 500, the OCTM database initialization function will fail). For this initial allocation, only eleven records are obtained for the ***base communications area*** and one record obtained for the ***update communications area***. Further records are obtained when the OCTM database build function is performed (see below).

- **Initialize eleven records in base communications area**

  The first record in the base communications area is initialized as the OCTM control record. The following nine records are initialized as communications group name records. The eleventh record is initialized as the first base communication record.

In the ***base communications area***, apart from the first ten records, every record is a base communication record. ALCS fits a maximum of 15 logical records into each physical record (each logical record is a minimum of 264 bytes). If the communications user data area is used, the size of the logical record is increased to accommodate this. This may reduce the number of logical records that fit within a physical record, for example, if the user data area is 40 bytes long, ALCS will fit 13 logical records within a physical record.

In the ***update communications area***, ALCS fits 6 logical records into each physical record (each logical record is a minimum of 648 bytes).

Each communication resource in the ***base communications area*** is held in a condensed CO0IN macro format. Each logical record not only contains details of a communication resource, but also critical information that is required by the OCTM facility. For example, when a COMTC macro is issued for the resource, information about that COMTC (the COMTC action, the time/date when the COMTC was issued, etc.) is stored in the logical record in the ***base communications area***. If a COMTC macro is updating a current communication resource, the logical record in the ***base communications area*** contains a pointer to a logical record in the ***update communications area*** (which contains details of the update).

When the OCTM database build occurs (see below) additional records will be required for the ***base communications area***. These will be dynamically obtained from the L3 allocatable pool in batches of 50 records. When usage of the OCTM facility commences and additional records are required for the ***update communications area***, these will also be dynamically obtained from the L3 allocatable pool. ALCS will obtain additional records, as required, in batches of 50 physical records (each batch providing 300 logical records). Each time ALCS requires an additional batch of 50 records from the L3 allocatable pool, it verifies that there are at least 500 available records remaining in the L3 allocatable pool.

Build the OCTM database

After the OCTM database initialization is complete, the ALCS system can be brought up to NORM system state. The operator command ZOCTM BUILD can now be used to activate the OCTM database build function. The OCTM database is built from the definitions of the terminal and X.25 PVC resources (types 1/6/7) in the communications generation load module(s). OCTM accesses the communications load list, reads each communications generation load module, and populates the ***base communications area*** in the OCTM database with all the required definitions (except Prime and RO Cras). The ZOCTM STATUS command can be used to verify the status of OCTM both before and after the OCTM database build.

Each ALCS user must ensure that they have sufficient L3 allocatable pool records to accommodate the requirements of the OCTM database before activating the OCTM database build. When the OCTM database has been built, a restart of the ALCS system is required to enable the OCTM facility to become fully operational (as described below).

Before restarting ALCS, a verification of the OCTM database could be performed to ensure that it is fully ready for operational use. To do this, run the OCTM database backup function (ZOCTM BACKUP) to create an OCTM sequential file. Run the ALCS Communications Report program (DXCCOMOL) to obtain a report of all the communication resources, including those on the OCTM database. Review the communications report and verify that it includes all the communication resources.

Activate OCTM

To fully activate OCTM, two further actions are required. Firstly, another restart of ALCS is required to build all the core resident tables that are required for operational usage of OCTM. Secondly, when the ALCS system has reached either IDLE or NORM system state, the ZOCTM START command must be entered to fully activate OCTM, therefore allowing the system administrators to start using the CEUS (enabling them to utilize the functionality provided by the OCTM facility).

During system restart, ALCS builds the online communication table from the communications generation load module(s) and the OCTM database. It also builds many core resident tables which are required by the operational OCTM facility. Ensure that the same communications generation load modules (same communications load list) are used by the ALCS system when the OCTM database initialization function is performed and when the OCTM activation is performed. When ALCS builds the online communication table during restart, it first reads the communications generation load module(s) and then reads all the logical records in the ***base communications area*** of the OCTM database.

During this ALCS restart, when ALCS builds the online communication table, it will find identical resources in the communications generation load module(s) and the OCTM database. These resources are the terminals (except Prime and RO Cras) and the X.25 PVC's (types 1/6/7). When ALCS restart finds these resources in the generation load module(s), it will treat them as *dummy* resources and not load them. It will though check in the generation load module(s) to see if these *dummy* resources have explicitly defined CRI addresses and ordinal numbers, and if they do not, will allocate CRI's and ordinal's for them in the same way as it did previously. This is required so that the CRI's and ordinal's that are allocated to the non-OCTM communication resources remain the same as they were prior to the OCTM migration.

Initial Usage of OCTM

After the OCTM migration is complete, the CEUS can be used to start adding and updating communication resources. Select some specific resources that require updating so that you can verify that OCTM is functioning correctly on your ALCS production system. Within a week after the OCTM migration has completed, begin the process of condensing the communication generation decks (see “Condensing the Communication Generation Decks” on page 17). There are important reasons for commencing this condensing process as soon as possible. Until the condensing function has been completed, communication resources must **not be deleted** from either the OCTM database or the communications generation. Also, non-terminal resources in the communication generation must not be added or modified.

Condensing the Communication Generation Decks

The terminal and X.25 PVC resources that are defined in the offline communications generation but which are now managed by OCTM should be removed from the communication generation base and update decks. This is not required immediately after OCTM has become fully operational, but it should be done within one week after that. Most ALCS systems have a large terminal network connected to them, therefore they require multiple communication generation decks to accommodate all the COMDFLT/COMDEF macros for the terminal resources. When the communication resources managed by OCTM are removed from the generation decks, the remaining communication resources could be combined together into a single generation deck (these are the X.25 PVC types 2 to 5, the APPC connections, the ALCI LU's, etc.).

Although the OCTM facility provides significant benefits in the management of the communication resources, it does though split the communication resources between the communications generation and the OCTM database. This split required the introduction of additional parameters on the communications generation COMGEN macro for managing the CRI addresses and the resource ordinal numbers. These are the **CRIRANGE=** and **ORDRANGE=** parameters. Immediately after the OCTM migration is complete, there will be no conflict in the usage of CRI's and ordinals because the OCTM database had been built from the communication generation load module(s). When the CEUS starts adding new terminal resources it can request OCTM to give specific CRI addresses and ordinal numbers to the new resources. In parallel with this, new COMDEF's could be added to the communication generation (for example, a TCP/IP server resource) and use the same CRI addresses and ordinals as those used by OCTM. The COMGEN **CRIRANGE=** and **ORDRANGE=** parameters therefore ensure that this double usage of CRI's or ordinals does not occur. The OCTM facility will not allow any new terminal or X.25 PVC resources that are being added to the OCTM database to use the CRI's or ordinals that are in these ranges.

When the condensing of the communication generation occurs, special attention must be given to the CRI addresses and resource ordinal numbers of those resources still in the generation deck. The current CRI's and ordinal numbers can be retained, or new CRI's and ordinals can be allocated.

- **Retaining the current CRI's and ordinals**

  In the communications generation you can either explicitly define CRI's and ordinals or you can allow ALCS to allocate them. If you have allowed ALCS to allocate them, you must now change your COMDFLT/COMDEF macroinstructions and explicitly define the CRI's and ordinals. You can run the OCTM communications report program to find out which CRI's and ordinals are allocated to each resource.

  You can now optionally use the new parameters on the COMGEN macro (**CRIRANGE=** and **ORDRANGE=**) to reserve these CRI's and ordinals for the exclusive use of the communications generation and to also reserve additional CRI's and ordinals for future communication resources in the communication generation. These new COMGEN parameters allow 10 ranges of CRI's and ordinal numbers to be reserved. You could, for example, use up to 8 of these ranges to reserve current CRI's and ordinals and up to 2 of these ranges to reserve new groups of CRI's and ordinals (for future resources). Ensure that all the CRI's and ordinals used in the communication generation are defined in one of these ranges, otherwise you will receive warning messages when you run the communications generation.

If you find it difficult to accommodate the current CRI's and ordinals within just 8 ranges, then we would recommend implementation of the alternative option (allocate new CRI's and ordinals) as described below.

- **Allocate new CRI's and ordinals to current resources**

  Identify a range of CRI's and ordinals that are **not used** by any communication resources (in the communication generation or the OCTM database). Define those ranges in the COMGEN **CRIRANGE=** and **ORDRANGE=** parameters. Allocate new CRI's and ordinals within those ranges to the communication resources in the condensed communication generation. Do this by coding the ICRI= and IORD= parameters on the COMDFLT macro (alternatively, code the CRI= and ORD= parameters on the COMDEF macro).

When you have chosen the appropriate solution for managing the CRI's and ordinals, implement the required changes in the communication generation deck and complete the condensing of the COMDFLT/COMDEF's. Run the communications generation stages 1 and 2, create the load module(s) and prepare a new communications load list. Run the OCTM communications report program to validate the contents of the load module(s).

Managing the OCTM Operation

When the OCTM facility is fully operational, consideration must be given to the ongoing monitoring and maintenance of OCTM. The OCTM facility includes a range of functions that allow this to be done. This section of the OCTM user guide describes these functions and how they should be used.

Displaying OCTM status information

The OCTM database holds various information that is not directly related to the communications resources. This information can be displayed using ZOCTM commands. Three display functions are provided.

1. The **ZOCTM STATUS** command will display miscellaneous information such as the current OCTM status, number of groups currently allocated, number of physical records in use, etc.

2. The **ZOCTM GROUP=group_name** command will display detailed information about a currently allocated communications group. For example, the status of the group, date/time of last change (last COMTC for group), number of resources in group, the CRN of each resource that belongs to the group, etc. The information provided in this display can also be obtained by the CEUS via the COMTC QUERY macro.

3. The **ZOCTM GROUPS** command will display information about each group that is currently allocated. For example, the group name, number of resources in the group, date/time when group was allocated, date/time of last change to group, etc. The information provided in this display can also be obtained by the CEUS via the COMTC GROUPS macro.

Inhibiting access to the OCTM database

There may be occasions when the CEUS should be inhibited from accessing the OCTM database. Operator commands are therefore provided that can inhibit the usage of OCTM by the CEUS for short periods of time. When maintenance functions are performed on the OCTM database (for example, the OCTM database restore function, see “Running the OCTM database restore function” on page 20), usage of the OCTM facility by the CEUS is automatically stopped while the restore is running. Some ALCS users may wish to inhibit usage of the OCTM database when some of their own maintenance functions are being run. When the OCTM facility is stopped, the OCTM database can not be updated by the CEUS (via the COMTC macro) but the database will still be used for building the communications table during ALCS restart.

The operator command **ZOCTM STOP** inhibits access to the OCTM database and the command **ZOCTM START** enables access to the OCTM database.

Running the OCTM database backup function

The OCTM database resides in system fixed file, therefore it will automatically be included in the daily backup of the ALCS database. OCTM will though allow a specific backup to be taken of the OCTM database. All the resource definitions in the base and update communication areas of the OCTM database are written by the backup function to a sequential file (called the OCTM sequential file). The operator command **ZOCTM BACKUP** performs this function. There are various reasons why a backup may be required.

Firstly, when an offline report is required of the communication resources currently defined in the communications generation and the OCTM database. The OCTM sequential file created by the backup function is used as input to the offline ALCS Communications Report program (DXCCOMOL). See "ALCS Communications Report Program" on page 5 for a description of this offline program.

Secondly, an OCTM database may need to be specifically created for a regression test system. If a regression test is to be run using a copy of the production system database, the OCTM database on the regression test system may need to be modified. For example, if an input message file containing messages captured from the production system is to be used by the ALCS STV facility, some of the terminal definitions in the OCTM database must be changed to ***test*** resources (in the communications generation, this is defined by the TEST=YES option on the COMDEF macro). The OCTM sequential file created by the backup function could be used as input to the OCTM Offline Support program (DXCCTMOL). See "Running the OCTM Offline Support Program" on page 22 for a description of this program. This offline program enables changes to be applied to multiple terminal definitions and then writes the updated definitions to an output OCTM sequential file. This OCTM sequential file can be used as input to the OCTM database restore function (ZOCTM RESTORE) to rebuild the OCTM database on the regression test system.

Thirdly, if changes are required to multiple communication resources on the production system OCTM database, the OCTM sequential file created by the backup function could be used as input to the OCTM Offline Support program (DXCCTMOL) which can apply the required changes. For example, if the format of the communications user data area requires changing for all the terminal resources, the offline program can perform all the required reformatting. The OCTM sequential file output by DXCCTMOL contains the updated communication definitions, and this can be loaded onto the production system via the OCTM database restore function.

The following is an example of the SEQGEN definition (in the sequential file generation) that could be used to define the output OCTM sequential file that is required by the database backup function.

```
SEQGEN NAME=CMB,
      TYPE=GEN,
      UNIT=(3380,1),
      DISP=(NEW,CATLG,CATLG),
      DSNAME=................CMB,
      LABEL=(,,,OUT,RETPD=0),
      RECFM=VB,
      VOLCNT=1,
      BUFNO=2,
      BLKSIZE=13000,
      SPACE=(1000,1000),
      LRECL=12900
```

Running the OCTM database restore function

The OCTM database restore function reads an OCTM sequential file and replaces the resource definitions in the OCTM database with those on the sequential file. The OCTM sequential file could be created by the OCTM database backup function (see "Running the OCTM database backup function" on page 19) or by the OCTM Offline Support program (see "Running the OCTM Offline Support Program" on page 22). The OCTM database restore function must be performed in IDLE system state and is activated by the operator command **ZOCTM RESTORE**.

The following is an example of the SEQGEN definition (in the sequential file generation) that could be used to define the input OCTM sequential file that is required by the database restore function.

```
SEQGEN NAME=CMR,
      TYPE=GEN,
      UNIT=(3380,1),
      DISP=(OLD,KEEP,KEEP),
      DSNAME=................CMB,
      LABEL=(,,,IN,RETPD=0),
      BUFNO=4
```

Expanding the communications table user data area

The OCTM database backup and restore functions can also be used to rebuild the OCTM database when an expansion of the communications user data area is required. The restore function builds new logical records in the base communications area on the OCTM database to accommodate an enlarged user data area. The steps required to perform the user data area expansion are as follows.

1. The size of the user data area is defined in the USERLEN= parameter on the communications generation COMGEN macro. Update the USERLEN= parameter on the COMGEN macro with the new size and run the offline generation process to create a new communications load module.
2. Create an OCTM sequential file by running the OCTM database backup function.
3. Restart the ALCS system to load the new communications load module (created by the offline generation).
4. When the ALCS system is in IDLE state, activate the OCTM database restore function to restore the OCTM sequential file. When the OCTM database restore function completes, the logical records on the OCTM database will contain the expanded user data area.
5. Restart the ALCS system again to rebuild the online communication table from the restored OCTM database. Each resource definition in the online communication table will now contain an expanded user data area.

Monitoring the output from OCTM Policing

An OCTM policing function monitors the contents of the base and update communications areas on the OCTM database. It checks each communications resource in the update communications area for periods of inactivity since the last COMTC macro was issued. If a period of inactivity is detected, OCTM policing sends a warning message to the RO Cras, indicating that further action is required. The first warning message is sent after 48 hours of inactivity, and further warning messages will be sent every 8 hours after that. These warning messages should be monitored as they indicate that a CEUS user (a system administrator) has submitted change requests to OCTM, but has not taken any action with them for the past 2 days (or longer).

The OCTM policing function does though activate an ECB-controlled exit program (AOCM) which can be used to issue a COMTC macro to automatically activate the next action that is required by the communication resource or group (for example, if the communications resource is waiting to be confirmed, the exit would issue the COMTC CONFIRM macro for that resource). There are four different COMTC macros that can be issued by the AOCM exit program:

- COMTC LOAD
- COMTC CONFIRM
- COMTC COMMIT
- COMTC UNALLOCATE

Implementating the AOCM exit program, with the automatic activation of the next COMTC action, can reduce the need to monitor the RO Cras for the warning messages output by OCTM policing.

Running the Offline Communications Report Program

The offline Communications Report File Generator program (DXCCOMOL) provides a report of the communication resources defined in the offline communications generation. If the OCTM facility is being used, DXCCOMOL can also include in its report, details of the communication resources that are managed by OCTM. The input for DXCCOMOL is the communications generation load module(s), plus the OCTM sequential file (created by the OCTM database backup function). The report provides a useful summary of all the communication resources in the ALCS online communication table and therefore should be run on a weekly basis.

DXCCOMOL writes details of each communication resource to a sequential data set called the ALCS communication report file. It is in a format that allows the IEBPTPCH utility to print selective details of each resource. Different types of report can be obtained by using a utility to sort the contents of the communication report file prior to printing it with IEBPTPCH (for example, sort the resources into resource ordinal number sequence).

The name of OCTM sequential file data set name is defined to DXCCOMOL in a parameter on the JCL EXEC statement. The data set name can be any length (and can therefore overflow onto a second line), for example:

```
//        EXEC PGM=DXCCOMOL,PARM='COMSANJW,ALCSPROD.OCTM.SEQFILE.BACKU-
//            P.AUGUST'
```

The DXCCOMOL JCL does not require any DD statement for this OCTM sequential file because the offline program uses dynamic allocation to open the data set.

Running the OCTM Offline Support Program

The OCTM offline support program DXCCTMOL provides additional functionality for the management of the OCTM database. It can be used to apply a large number of changes to the communication resources managed by OCTM, it can be used to validate and implement new ranges of CRI addresses and resource ordinals (for the exclusive use of the offline communications generation), and it can also be used to create a dataset that contains an ALCS generation COMDEF macro for every communication resource on the OCTM database.

The following describes the functions provided by the OCTM offline support program DXCCTMOL and how they can be used.

- **Modifying and deleting communication resources**

Use this offline program to apply a large number of changes to the communication resources managed by OCTM. The communication resource definitions on the OCTM database are normally modified and deleted via the COMTC macro, but when a large number of changes are required, they can be implemented more efficiently by using this OCTM offline support program. These changes can include modifications to multiple communication resources and they can also include the deletion of multiple communication resources.

The primary input is an OCTM sequential file, created by the OCTM database backup function (activated by the ZOCTM BACKUP command). The primary output from this offline program is a new OCTM sequential file. This new sequential file contains all the communication resources that have **not** been deleted, and for those that have been modified, the updated communications definitions. The OCTM database restore function (activated by the ZOCTM RESTORE command) can be used to rebuild the OCTM database on the ALCS system from this new OCTM sequential file.

This offline program activates an installation-wide exit program DXCUTMOL which identifies the modifications that are required to specific communication resources and also identifies the resources to be deleted. This exit is activated by the offline program for every communications resource on the input OCTM sequential file (which also includes recent in-progress change requests). See ALCS Installation and Customization for details of this installation-wide exit program and guidance on how it can be used to apply changes to the communication resources on the OCTM database. When this exit program is assembled and link-edited, the load module must be placed in one of the DXCCTMOL offline program STEPLIB load libraries. This is required because DXCCTMOL invokes the exit program dynamically during program execution.

Use the DXCUTMOL exit program to apply the following types of change to the communication resources managed by OCTM.

- Set on the "test" status for communications resources that will be used by the ALCS STV (ZTEST) function
- Change the "initial" status from active to inactive
- Modify the content and format of the communications user data
- Delete all the X.25 PVC and ALC communication resources
- and so on

The DXCCTMOL offline program outputs a report to the PRINT dataset giving the names of the communication resources that have been modified or deleted. When this program has been run, check the report to verify that the list of resources that have been modified or deleted is correct. If not correct, review the code in the DXCUTMOL exit program, and if changes are required, apply those changes and reassemble the exit. Link-edit the exit and re-run the offline program. Check the report and if it is correct, use the output OCTM sequential file created by this offline program to rebuild the OCTM database on the ALCS system.

- **Validating and implementing new CRIRANGE and ORDRANGE parameters**

  Use this offline program to validate and implement new ranges of CRI addresses and resource ordinal numbers (for the exclusive use of the offline communications generation). If the current ranges of CRI addresses and resource ordinals that are defined in the CRIRANGE and ORDRANGE parameters (on the communications generation COMGEN macro) require modification, use this offline program to validate those changes against the communication resources on the OCTM database. Define the new ranges to this offline program in the SYSIN dataset. The offline program identifies the communication resources on the OCTM database that are using any of the CRI addresses or resource ordinals within the new ranges. It reads an OCTM sequential file (created by the OCTM database backup function) and outputs a report to the PRINT dataset giving the name of each communication resource that has a CRI or ordinal within the new ranges. The report outputs the following message for each resource that is within the specified ranges:

```
Resource  crn  deleted by CRI/ORD ranges
```

When you define the new ranges to the offline program (in the SYSIN dataset), code them in one or more records, as follows:

```
CRIRANGE=(080001,08006F,08009F,0800FF)
CRIRANGE=(090001,0900FF)
ORDRANGE=(100,150,250,280)
```

Information on the format of these CRIRANGE and ORDRANGE parameters can be found in the description of the communication generation COMGEN macro in ALCS Installation and Customization.

When you have run this program, check the report (in the PRINT dataset) to verify the list of resources that will be deleted from the OCTM database. If necessary, adjust the ranges of CRI addresses or ordinal numbers and re-run the program. A new OCTM sequential file is created by the program containing an updated OCTM control anchor record (with the NEW ranges of CRI addresses and ordinal numbers defined) plus all the communications resources that have NOT been deleted. Use this new OCTM sequential file to rebuild the OCTM database on the ALCS system (using the OCTM database restore function). Finally, update the CRIRANGE and ORDRANGE parameters on the communications generation COMGEN macro with these new ranges of CRI addresses and ordinal numbers.

- **Creating communications generation COMDEF macros**

  Use this offline program to create a sequential dataset which contains an ALCS generation COMDEF macro for every communication resource on the OCTM database. This enables communication generation decks to be created for all the resources on the OCTM database. Use this feature of the offline program when you are building a new ALCS system that will not use the OCTM facility (and when you require all the communication resources to be defined in the offline communications generation). Code PARM=LIST on the JCL EXEC statement to request this offline program to create the communication generation COMDEF macros. The program reads the input OCTM sequential file (created by the OCTM database backup function) and outputs the COMDEF macros to the LIST dataset.

The following is an example of the JCL that should be used for running the DXCCTMOL offline program.

```
//S1      EXEC PGM=DXCCTMOL,PARM=LIST
//STEPLIB DD  DSN=DXC.V2R3M1.DXCLMD1,DISP=SHR
//        DD  DSN=DXC.V2R3M1.DXCLMD2,DISP=SHR
//INPUT   DD  DSN=old_octm_backup_sequential_file,DISP=OLD
//OUTPUT  DD  DSN=new_octm_backup_sequential_file,
//            DISP=(NEW,CATLG),SPACE=(CYL,(50,50)),
//            DCB=(LRECL=12900,BLKSIZE=13000,RECFM=VB)
//PRINT   DD  SYSOUT=*,DCB=(LRECL=133,BLKSIZE=1330)
//LIST    DD  DSN=octm_list_sequential_file,
//            DISP=(NEW,CATLG),SPACE=(CYL,(50,50)),
//            DCB=(LRECL=80,BLKSIZE=8000,RECFM=FB)
//SYSIN   DD  *
//*
```

The following describes each input and output dataset (in the above DD statements) required by the offline program.

- INPUT dataset
  This is the input OCTM sequential file created by the online OCTM database backup function.

- OUTPUT dataset
  This is the output OCTM sequential file created by the offline program (which can be used as input to the online OCTM database restore function).

- PRINT dataset
  This is the report created by the offline program containing the names of the communication resources that have been modified or deleted.

- LIST dataset
  This is a sequential dataset containing an ALCS generation COMDEF macro for every communication resource on the OCTM database. This dataset is produced when PARM=LIST is specified on the JCL EXEC statement.

- SYSIN dataset
  This contains the new ranges of CRI addresses and ordinal numbers when this offline program is used for validating and implementing new CRI and ordinal ranges. Specify either CRIRANGE or ORDRANGE, or specify both. They should be specified on separate input lines, in the format:

  ```
  CRIRANGE=(nnnnnn,nnnnnn,nnnnnn,nnnnnn,.....)
  CRIRANGE=(nnnnnn,nnnnnn,nnnnnn,nnnnnn,.....)
  ORDRANGE=(nnnnn,nnnnn,nnnnn,nnnnn,....)
  ORDRANGE=(nnnnn,nnnnn,....)
  ```

This offline program terminates with register R15 set to:

R15 = 00	OK
R15 = 08	Invalid CRIRANGE and/or invalid ORDRANGE
R15 = 12	Invalid parameter on EXEC statement
R15 = 16	Unable to load or link DXCUTMOL

For a summary of the functions provided by this offline program, see "OCTM Offline Support Program" on page 6.

Appendix A - Communications End User System (CEUS)

The CEUS provides the user interface for the OCTM facility. This section of the user guide contains an overview of the CEUS, provides detailed information on the sample CEUS application, and also provides guidance on the type of functionality that you may wish to include in your own CEUS application.

CEUS Overview

The CEUS is comprised of ECB-controlled programs which receive communication change requests from end users and issue COMTC macros for those change requests. At most ALCS installations, the OCTM end users will be system administrators who are responsible for accepting communications change requests, checking them and then entering them into the OCTM facility via the CEUS.

The CEUS will be unique for each ALCS system because it must provide functionality that meets the specific operational and network requirements of the ALCS system. Each ALCS system will have its own unique requirements regarding the procedures which should be followed when applying changes via OCTM to the online communication table. Each ALCS system will also have its own unique layout and format for the communication table user data area.

Although the CEUS must include some ALCS ECB-controlled programs (for issuing the COMTC macro) the complete CEUS software could be split over multiple platforms. In particular, the software that provides the end user interface (for example, the screen formats used for entering details of new terminals) does not need to reside in ALCS and could be developed on another platform. It could be a GUI interface on a workstation, using a TCP/IP link to ALCS. It could also be developed using the ALCS 3270 screen mapping package.

Sample CEUS Application

A sample CEUS application is provided by IBM on the ALCS web site. A further sample CEUS application can be obtained from one of the ALCS users (contact IBM for more details on this). Both of these sample CEUS applications are available to the ALCS users on an ***as is*** basis. Each ALCS user can develop a CEUS using one of the supplied CEUS applications as the primary building block, or alternatively they can develop their own.

The sample CEUS developed by IBM is a 3270 map application which provides the basic functions that are necessary for interaction with the ALCS OCTM facility. Those basic functions include:

- Utilizing ALCS 3270 screen mapping support, providing formatted screens for inputting communications data
- Displaying details of communication resources managed by OCTM
- Providing screen maps that allow every type of communication data to be input to OCTM
- Utilizing the COMTC macro to submit change requests to OCTM (add, replace and delete communication resources) and to activate those change requests

- Reporting error conditions

Although the sample CEUS application provides only the basic functions, it does exercise and demonstrate the use of the COMTC macro. Each entry in the sample application is atomic. Input is received, an action performed, and another map is displayed. Nothing is held by ALCS relating to a session and no information is stored in the end user's terminal related records. The sample CEUS operates as a stateless application.

When each ALCS user develops their own CEUS application, it may need be more sophisticated than the sample CEUS developed by IBM. The additional functionality that ALCS users might require in their CEUS could include the following:

- Help information for each of the screen maps
- A security interface for ensuring that only authorized system administrators use the CEUS application
- Status information on each communications group, maintained in database records
- Limitations on the usage of some COMTC macro parameters

The section in this user guide called "Limitations of Sample CEUS" on page 38 provides information on the limitations of the sample CEUS and includes suggestions on how you may wish to remove those limitations when you develop your own CEUS application. The section in this user guide called "Additional CEUS Functionality" on page 39 contains suggestions on the additional functionality that could be included in your CEUS application.

Components of Sample CEUS

The primary ECB-controlled program for the sample CEUS application (the input editor program) is called A0PP. This program determines if the input is from one of the sample CEUS screen maps, and if it is, may enter the ECB-controlled program A0SL or A0GL based on the input map name. Similarly, programs A0SL and A0GL may route to subsequent programs based on the map used in the input message. Each map has a displayed field (called XMAP) at the same position in the screen map, and this field contains the map name. The primary screen map for the sample CEUS is called A0P0. All other screen maps are called A0Gx or A0Sx (where the last digit of the map name is numeric).

If the input message contains one of the Communications Group screen maps (the A0Gx maps) then the message will be processed by one of the A0Gx programs. If the input message is for a single communication resource (and is not associated with a communications group) it will contain an A0Sx screen map and will be processed by one of the A0Sx programs. As most of the information handled is common for the A0Gx and the A0Sx programs, they share the same subroutine program A0Ax. The A0Ax subroutine program contains two transfer vectors, A0Bx and A0Cx.

There are programs and maps for each ALCS communication device type. For example, the VTAM 3270 device type is managed by programs A0GV, A0SV and A0AV, and by screen maps A0G1 and A0S1. The screen map members contain ALCS MAP3270 macroinstructions. These macroinstructions create map DSECTs which define the input and output fields on the 3270 screen. They specify attributes of the screen (size, and so on) and the start position and length of each map field and its initial contents.

When you are altering or extending the sample CEUS application, the map fields need to be maintained at the same offset (for example, maps A0S1 and A0G1 have all modifiable fields at the same offset). All the screen maps contain fields called XOPTION, XNAME, XCRI, XPOPUP, MCRI, MCRIBASE, MORD and MORDBASE at the same offset so that thay can share the same routines in the ECB-controlled programs.

The sample CEUS application is made up of 19 3270 screen maps and 28 ECB-controlled programs. The names used are illustrated in the following diagrams.

Names of ECB-controlled Programs

```
                              +--------+
                         A0PP |Primary |
                              |Program |       +--------+
                              |        |  A0PE | COMTC  |
                              +--------+       | Error  |
                                               | Decoder|
                                               +--------+
            +--------+                            +--------+
       A0SL |Single  |                       A0GL |Group   |
       A0SM |select  |                       A0GM |select  |
            |LDTYPE  |                            |LDTYPE  |
            +--------+                            +--------+

           MAP    PROG      Subroutines for       PROG   MAP
           NAME   NAME      Single and Group      NAME   NAME
           ----   ----     PROG  TRANV  TRANV     ----   ----
                           ----  -----  -----
VTAM3270   A0S1   A0SV     A0AV   A0BV   A0CV     A0GV   A0G1
MQTERM     A0S2   A0SQ     A0AQ   A0BQ   A0CQ     A0GQ   A0G2
NETVIEW    A0S3   A0SN     A0AN   A0BN   A0CN     A0GN   A0G3
VTAMALC    A0S4   A0SA     A0AA   A0BA   A0CA     A0GA   A0G4
TCPIPALC   A0S5   A0ST     A0AT   A0BT   A0CT     A0GT   A0G5
OSYS       A0S6   A0SO     A0AO   A0BO   A0CO     A0GO   A0G6
X25ALC     A0S7   A0SX     A0AX   A0BX   A0CX     A0GX   A0G7
X25PVC     A0S8   A0SP     A0AP   A0BP   A0CP     A0GP   A0G8
```

Names of 3270 Screen Maps

```
                         +--------+
                    A0P0 |Primary |
                         |Map     |
                         |        |
                         +--------+
       +--------+                           +--------+
   A0S0|Single  |                      A0G0 |Group   |
       |select  |                           |select  |
       |LDTYPE  |                           |LDTYPE  |
       +--------+                           +--------+

       +--------+   VTAM3270                +--------+   VTAM3270
       |        |+   MQTERM                 |        |+   MQTERM
       |        ||+   NETVIEW               |        ||+   NETVIEW
       |        |||+   VTAMALC              |        |||+   VTAMALC
  A0S1 +--------+|||+   TCPIPALC      A0G1 +--------+|||+   TCPIPALC
   A0S2 +--------+|||+   OSYS          A0G2 +--------+|||+   OSYS
    A0S3 +--------+|||+   X25ALC        A0G3 +--------+|||+   X25ALC
     A0S4 +--------+|||+   X25PVC        A0G4 +--------+|||+   X25PVC
      A0S5 +--------+|||                  A0G5 +--------+|||
       A0S6 +--------+||                   A0G6 +--------+||
        A0S7 +--------+|                    A0G7 +--------+|
         A0S8 +--------+                     A0G8 +--------+
```

Installing the Sample CEUS

The following describes the primary tasks required to install the sample CEUS application on your ALCS system.

- **Install the sample CEUS 3270 screen map members**

  Install the 19 3270 screen map members in a source statement library. The names of these members are A0S0 to A0S8 (9 members), A0G0 to A0G8 (9 members) and A0P0.

- **Install and customize the sample CEUS A01PP member**

  Install the A01PP copy member in a source statement library. The A01PP copy member contains two items that you may wish to alter before assembling the sample CEUS programs.
  The first item in A01PP is the name of the sample CEUS application. That is currently defined as OCTM so that it matches the communications generation COMDEF macro described below. You can select any 4-character name for the sample CEUS application and define that name in the A01PP member and the COMDEF macro.
  The second item in A01PP is the name of the application that the sample CEUS will return to when exiting. That is currently defined as RES0 because most ALCS customers use the application name of RES0 for the primary ALCS application. If you require the sample CEUS to return to a different application when you exit from the sample CEUS, then define the name of that application in the A01PP member.

- **Install and assemble the sample CEUS ECB-controlled programs**

  Install the 28 ECB-controlled programs in a source statement library. The names of these programs are A0Ax (8 members), A0Gx (9 members), A0Sx (9 members), A0PP and A0PE. The last character of each program name is alphabetic. These programs should be assembled against your ALCS macro libraries etc. Ensure that you include in your SYSLIB the source statement library that contains the screen map members and the A01PP member.

- **Create load module for the sample CEUS programs**

  Verify that each ECB-controlled program has assembled without error. Create an application program load module for the 28 programs in the sample CEUS (to be loaded via the ZPCTL command).

- **Create sequential file for the screen maps**

  Assemble each screen map member with the assembler parameter SYSPARM(GENERATE), using the MVS JCL parameter PARM. Copy the SYSPUNCH output from this assembly to a sequential data set using the IEBGENER utility program. The ALCS ISPF panels can be used for running these two jobs. Alternatively, extract the sample JCL from the ISPF panels, update it and then run these two jobs.

- **Load the sample CEUS programs and maps onto your ALCS system**

  Verify that your ALCS sequential file configuration table contains a definition for the screen map sequential file. Load the screen map sequential file using the ZCMSP command. Load the application program load module using the ZPCTL command.

- **Define sample CEUS application in ALCS communications generation**

In order to invoke the sample CEUS application, it must be defined to ALCS in a communication generation COMDEF macro. The following is an example of the COMDEF macro that should be coded when the application name is OCTM.

```
COMDEF LDTYPE=ALCSAPPL,NAME=OCTM,PROG=A0PP,SYSSTATE=IDLE,ISTATUS=ACTIVE
```

When the ALCS communications table contains this definition, the ZROUT OCTM command can be used to route the 3270 terminal to the sample CEUS application.

Using the Sample CEUS

The first step that is required when you are ready to start using the sample CEUS is to route your 3270 terminal to the application. Use the ZROUT command to do this. If you have installed the sample CEUS with the application name defined as OCTM, use the ZROUT OCTM command to enable your 3270 to access the sample CEUS. After you receive the response to ZROUT, just hit the enter key, and this will display the primary screen map.

The following sections describe how the different screen maps are used.

A0P0 - The Primary Screen Map

There are five options provided on this primary map.

- **Option 1 - List Groups**

  Enter **1** in the option field of the map and hit enter. This will display a list of the currently allocated communications groups. The sample CEUS uses the COMTC GROUPS macro to obtain this list. When a system administrator needs to allocate a new communications group, they may need to determine if the communications group name they are planning to use for the new group is already in use. They may also wish to verify how many groups are currently active. The COMTC GROUPS macro returns to the sample CEUS information about each communications group in the Communications Groups Information DSECT (CT3TM). Although COMTC provides status information about each group, the sample CEUS only displays the name of the first 86 allocated groups. If more information is required, use the ZOCTM GROUPS command.

- **Option 2 - Work with a Group**

  Enter **2** in the option field of the map and enter the name of the group in the **Groupname** field. This will display the A0G0 screen map which enables you to start working with this communications group. The A0G0 screen map is described in “A0G0 - Screen Map for Managing a Group” on page 34.

- **Option 3 - Allocate Group**

  Enter **3** in the option field of the map and enter the name of the group to be allocated in the **Groupname** field. The sample CEUS uses the COMTC ALLOCATE macro to allocate the group name. You should allocate a communications group name when you need to work with a batch of communication change requests. If you use a communications group when submitting multiple change requests to OCTM, those change requests can be activated in the ALCS communications table as a single group. A unique name must be given for the communications group (it can be a maximum of 7 characters) and it must be allocated before any change requests are submitted for the group.

- **Option 4 - Unallocate Group**

Enter **4** in the option field of the map and enter the name of the group to be unallocated in the **Groupname** field. The sample CEUS uses the COMTC UNALLOCATE macro to unallocate the communications group. You should normally unallocate a communications group after the change requests belonging to that group have been committed. When a communications group has been committed, there will normally be no further requirement to retain the group (or the group name), therefore you should unallocate it soon after the commit.

There is a feature of COMTC UNALLOCATE which is not used by the sample CEUS. If a number of change requests have been submitted for a group and the group is not yet loaded (or has been backed out), a COMTC UNALLOCATE can be used to delete the group and cancel all the change requests in that group. This may be required if a batch of new terminals have been incorrectly added and all the change requests for that batch require cancelling. If you wish to use this feature, you could enhance this A0P0 screen map to include an additional option that will provide this feature.

- **Option 5 - Work on single resource**

  Enter **5** in the option field of the map (and hit the enter key) when you do not wish to work with a communications group and need to issue a change request for a single communication resource. This will display the A0S0 screen map which enables you to start working with a single communication resource. The A0S0 screen map is described below.

A0S0 - Screen Map for Single Resource Selection

There are eight options provided on this screen map. Each option is for a specific communications resource type. For example, option 1 is for the VTAM 3270 terminal type, option 2 is for the MQ terminal type, etc. Select the resource type that you need to work with by entering the appropriate number in the option field and hitting the enter key. This will display one of the screen maps, A0S1 to A0S8. The A0S1 map is for VTAM 3270 terminals, the A0S2 map is for MQ terminals, and so on. The A0S1 to A0S8 screen maps are described below.

A0S1 to A0S8 - Screen Maps for Working with a Single Resource

There are nine options provided on each one of these screen maps.

- **Option 1 - Show a Resource**

  Enter **1** in the option field of the map and also enter either the name of the resource (the CRN) or the CRI address of the resource (in fields at the top of the screen). This will provide a detailed display of the communication resource. When a communication resource is to be changed or deleted, the system administrator should display and check the current definition and status of the resource in the ALCS online communication table. The sample CEUS uses the COMTC QUERY macro to obtain details of the communication resource and its current OCTM status. The COMTC QUERY macro returns the information in the Communications Resource and Group Information DSECT (CT2TM). When a new resource is being displayed, the sample CEUS displays the definition of the new resource plus status information about it. When a changed (replaced) resource is being displayed, the sample CEUS displays the definition of the modified resource.
  Before submitting a change (replace) request, use option **1** to display a screen map that contains a complete definition of the resource. Modify those parts of the definition that require changing, and then use option **3** to submit the change.

- **Options 2 to 5 - Submit Change Request**

These four options are used for submitting different types of change request for the communication resource. You can use option **2** to add a new resource, option **3** to change an existing resource, or option **4** to delete an obsolete resource. The sample CEUS uses the COMTC ADD, COMTC REPLACE and COMTC DELETE macros to perform these functions. If you have incorrectly submitted a change request (for example, you have used the wrong resource name) you can cancel that change request by using option **5**. The sample CEUS uses the COMTC CANCEL macro to cancel a previous COMTC ADD, REPLACE or DELETE. Refer to "Submitting communications change requests" on page 35 for more details on this.

- **Options 6 to 9 - Activate Change Request**

  These four options are used for activating the change request in the ALCS online communication table. When you have submitted the change request, you must activate it in the online communication table via a three stage process. Those three stages are called load, confirm and commit. Firstly, use option **6** to load the change request, secondly, use option **7** to confirm the change request, and lastly, use option **8** to commit the change request. The sample CEUS uses the COMTC LOAD, COMTC CONFIRM and COMTC COMMIT macros to do this. If you decide that the change request should not have been loaded, use option **9** to back out the change request. A change request can be backed out of the online communication table after either a load (option **6**) or a confirm (option **7**). It can not be backed out after a commit (option **8**). The sample CEUS uses the COMTC BACKOUT macro to do this. Refer to "Activating communications change requests" on page 37 for more details on this.

A0G0 - Screen Map for Managing a Group

There are thirteen options provided on this screen map.

- **Options 1 to 8 - Select Resource Type**

  Each of these 8 options are for a specific communications resource type. For example, option **1** is for the VTAM 3270 terminal type, option **2** is for the MQ terminal type, etc. Select the resource type that you need to work with by entering the appropriate number in the option field and hitting the enter key. This will display one of the screen maps, A0G1 to A0G8. The A0G1 map is for VTAM 3270 terminals, the A0G2 map is for MQ terminals, and so on. The A0G1 to A0G8 screen maps are described in "A0G1 to A0G8 - Screen Maps for Working with each Resource" on page 35.

- **Options L C M and B - Activate Change Requests**

  These four options are used for activating a batch of change requests in the ALCS online communication table. When you have submitted all the change requests for this group, you must activate them in the online communication table via a three stage process. Those three stages are called load, confirm and commit. Firstly, use option **L** to load the change requests, secondly, use option **C** to confirm the change requests, and lastly, use option **M** to commit the change requests. The sample CEUS uses the COMTC LOAD, COMTC CONFIRM and COMTC COMMIT macros to do this. If you decide that this batch of change requests should not have been loaded, use option **B** to back out all the change requests. The change requests can be backed out of the online communication table after either a load (option **L**) or a confirm (option **C**). They can not be backed out after a commit (option **M**). The sample CEUS uses the COMTC BACKOUT macro to do this. Refer to "Activating communications change requests" on page 37 for more details on this.

- **Option S - Show Group Information**

Enter **S** in the option field of the map and hit enter. This will display a list of the communication resources that currently belong to this communications group. The sample CEUS uses the COMTC QUERY macro to do this. When a system administrator is submitting a large number of change requests, it can be advisable to occasionally display the list of resources for which change requests have been submitted. The COMTC QUERY macro not only provides the list of resources for which change requests have been submitted, but also provides information about each change request. The response output by the sample CEUS contains only the names of the resources.

A0G1 to A0G8 - Screen Maps for Working with each Resource

There are five options provided on this screen map.

- **Option 1 - Show a Resource**

  Enter **1** in the option field of the map and also enter either the name of the resource (the CRN) or the CRI address of the resource (in fields at the top of the screen). This will provide a detailed display of the communication resource. When a communication resource is to be changed or deleted, the system administrator should display and check the current definition and status of the resource in the ALCS online communication table. The sample CEUS uses the COMTC QUERY macro to obtain details of the communication resource and its current OCTM status. The COMTC QUERY macro returns the information in the Communications Resource and Group Information DSECT (CT2TM). When a new resource is being displayed, the sample CEUS displays the definition of the new resource plus status information about it. When a changed (replaced) resource is being displayed, the sample CEUS displays the definition of the modified resource.
  Before submitting a change (replace) request, use option **1** to display a screen map that contains a complete definition of the resource. Modify those parts of the definition that require changing, and then use option **3** to submit the change.

- **Options 2 to 5 - Submit Change Request**

  These four options are used for submitting different types of change request for this communications group. You can use option **2** to add a new resource, option **3** to change an existing resource, or option **4** to delete an obsolete resource. The sample CEUS uses the COMTC ADD, COMTC REPLACE and COMTC DELETE macros to perform these functions. If you have incorrectly submitted a change request (for example, you have used the wrong resource name) you can cancel that change request by using option **5**. The sample CEUS uses the COMTC CANCEL macro to cancel a previous COMTC ADD, REPLACE or DELETE. Further information on submitting communications change requests is provided below.

Submitting communications change requests

Options **2** to **4** are provided for submitting different types of change request. These change requests enable new communication resources to be added, current resources to be changed, and obsolete resources to be deleted. Option **5** is provided for cancelling an incorrect change request. The following describes each of these four options.

- **Option 2 - Add a communication resource**

  You can request OCTM to add a terminal resource (except Prime or RO Cras) or an X.25 PVC resource (types 1, 6 and 7). Different screen maps are required for each resource type because the type of communications data that must be provided is different for each type of resource. The screen maps contain fields for all the required communications data that is needed for new terminals and new X.25 PVC's. You do not need to complete every field in the screen map. The sample CEUS provides default values for many fields, although some fields must be completed. The sample CEUS forwards to OCTM the data you have entered via parameters in the COMTC ADD macro and via a Communication Resource Definition DSECT (CT1TM).

Adding X.25 PVC and ALC resources require additional considerations. An X.25 ALC terminal can not be added if the definition for the owning X.25 PVC does not already exist on the OCTM database. If you need to add a new X.25 PVC and its associated X.25 ALC terminals, you can do this in either of the following two ways.

1. Add the X.25 PVC resource on its own (do not use a communications group) and then load, confirm and commit that resource. Now add the associated X.25 ALC terminals (use a communications group for adding these terminals).
2. Allocate a communications group and add the X.25 PVC and its associated X.25 ALC terminals in the same group. The X.25 PVC must be added before any of the terminals.

When adding an X.25 ALC terminal resource, the resource name must be comprised of a base CRN plus one or two other elements depending on the type of X.25 PVC that the ALC terminal is associated with. There are three types of X.25 PVC that can have associated ALC terminals. The length of the base CRN for these ALC terminals is dependent on the type of X.25 PVC that they are associated with. When these terminals are associated with a ***Type 1*** X.25 PVC, the base CRN (a string of 1 to 6 characters) must have a two-character terminal address appended to it. When these terminals are associated with a ***Type 6*** or ***Type 7*** X.25 PVC, the base CRN (a string of 1 to 4 characters) must have a two-character interchange address followed by a two-character terminal address appended to it. Note that the interchange address (if required) and the terminal address must also be defined in the **TA=** and **IA=** fields on the screen map.

In addition to the X.25 ALC terminal resource type, there are other terminal resource types that have owning resources. They are the MQ terminal, the WAS terminal, the TCPIPALC terminal, and the VTAMALC terminal. When a terminal belonging to one of these four resource types is added, the owning resource (MQ queue, WAS connection, TCPIP server connection or ALCI LU) must already be defined in the ALCS offline communications generation (and loaded in the online communication table).

- **Option 3 - Change a communication resource**

  You can request OCTM to change a terminal resource (except Prime or RO Cras) or an X.25 PVC resource (types 1, 6 and 7). You can not change the resource name (CRN), the CRI address or the resource ordinal number. If you need to change any of these, the resource must be deleted and added again. Different screen maps are required for each resource type because the type of communications data that must be provided is different for each type of resource. The screen maps contain fields for all the required communications data that can be changed. Display the resource via option **1** before making any change, and then alter those fields on the screen where the data requires changing. The sample CEUS forwards to OCTM the complete definition for the resource (OCTM replaces the current definition with this updated one) via COMTC REPLACE macro parameters and a Communication Resource Definition DSECT (CT1TM).

- **Option 4 - Delete a communication resource**

  You can request OCTM to delete a terminal resource (except Prime and RO Cras) or an X.25 PVC resource (types 1, 6 and 7). If an X.25 PVC resource is being deleted, all its associated X.25 ALC terminal resources must also be deleted. If a batch of X.25 terminals must be moved from one PVC to another (and the CRI and ordinal number for each terminal is be retained), this must be achieved by deleting the terminals and adding them again. The sample CEUS forwards to OCTM the delete request via the COMTC DELETE macro.

- **Option 5 - Cancel a communication change request**

  You can request OCTM to cancel a change request that you have recently submitted The change request being cancelled must have been submitted via options **2**, **3** or **4** (add, replace or delete). A change request can not be cancelled if it is currently loaded in the online communication table. If the change request has already been loaded (or has been confirmed) in the online communication table, you can request the sample CEUS to back it out. When the back out is complete, you can then cancel the change request. The sample CEUS forwards to OCTM the cancel request via the COMTC CANCEL macro.

Activating communications change requests

There are four options for activating change requests in the online communication table. The first three must all be performed to fully activate change requests in the online communication table. Those three are called load, confirm and commit. The last option enables a change request to be backed out. After a load or confirm, any change request can be backed out (removed) from the online communication table. The following describes each of these four options.

- **Option 6 or L - Load communication change requests**

  When option **6** is used (on screen maps A0S1 to A0S8) the sample CEUS requests OCTM to load a single change request. When option **L** is used (on screen map A0G0) the sample CEUS requests OCTM to load a group of change requests. The change requests are loaded into the online communication table. Change requests that have been loaded (but not confirmed) are not retained in the online communication table over an ALCS restart (although they are still retained in the OCTM database). If an ALCS restart occurs, you must request OCTM to load the change request(s) again.

- **Option 7 or C - Confirm communication change requests**

  After a successful load, you should request OCTM to confirm the change request (or group of change requests). When option **7** is used (on screen maps A0S1 to A0S8) the sample CEUS requests OCTM to confirm a single change request. When option **C** is used (on screen map A0G0) the sample CEUS requests OCTM to confirm a group of change requests. The change requests are confirmed in the online communication table. After a confirm, the change request(s) are retained in the online communication table over an ALCS restart.

- **Option 8 or M - Commit communication change requests**

  After a successful confirm, you should request OCTM to commit the change request (or group of change requests). When option **8** is used (on screen maps A0S1 to A0S8) the sample CEUS requests OCTM to commit a single change request. When option **M** is used (on screen map A0G0) the sample CEUS requests OCTM to commit a group of change requests. The change requests are committed in the online communication table. When change request(s) have been committed, they can not be backed out.

- **Option 9 or B - Back out communication change requests**

When change request(s) that have been loaded or confirmed are causing problems on your ALCS system, you should request OCTM to back them out from the online communication table. When option **9** is used (on screen maps A0S1 to A0S8) the sample CEUS requests OCTM to back out a single change request. When option **B** is used (on screen map A0G0) the sample CEUS requests OCTM to back out a group of change requests. The change requests are removed from the online communication table. A back out does not delete the change requests from the OCTM database, therefore when a group of change requests have been backed out, any change request(s) that were incorrect can be corrected (or cancelled) and the group loaded again.

Limitations of Sample CEUS

Each ALCS user will need to develop their own CEUS application. If the sample CEUS provided by IBM is to be used as the primary building block for the development of your own CEUS, then the limitations that currently exist in the sample CEUS may need to be removed as part of that development effort. The following describes some of the limitations.

- **Communications User Data**

  The processing of the communications user data will always be unique for each ALCS customer. The sample CEUS provides a single field in the screen maps for inputting user data. That field is USERDAT=, and it allows a maximum of 20 characters of data to be input. This field allows only alphanumeric characters to be input and does not support hex data.
  When you customize the sample CEUS for your ALCS system, you may wish to provide a field in the screen maps for each field in your communications user data area. You are recommended to validate your communications user data before submitting it to OCTM (via the CT1TM DSECT) because OCTM will not validate it for you. You may already use the offline communications generation DXCZCUSR macro for validating your user data, therefore you should replicate the DXCZCUSR functionality in your CEUS application.

- **Error and information messages**

  Most of the screen maps in the sample CEUS required many fields to be defined in them. This therefore resulted in only limited space being available in each screen map for the error and information messages that can be output by the sample CEUS. Some actions, such as loading a large number of change requests in a communications group, may generate many error messages. Although OCTM returns to the sample CEUS (via the COMTC macro) a storage block containing all the error messages, only the first error message is displayed in the screen map. On all of the screen maps, only one line is provided for displaying the error and information messages. A single error message will normally fit on a line, but some messages may be too long and therefore may be truncated. When errors occur during the loading or backing out of communication change requests, you may need to refer to other sources (RO CRAS, system log or diagnostic file) to understand fully the errors that occurred. In summary, COMTC ADD and COMTC REPLACE can return multiple error messages in a storage block, and so can COMTC LOAD and COMTC BACKOUT.
  When you customize the sample CEUS for your ALCS system, you may wish to provide additional lines on your screen maps for the error and information messages.

- **Displaying a communications group**

The OCTM facility provides a COMTC QUERY macro and a ZOCTM GROUP=groupname command for obtaining a list of the communication resources that belong to a communications group. OCTM allows a maximum of 400 resources in a group, but the sample CEUS displays the names of only the first 64. The COMTC macro and the ZOCTM command both indicate the type of change request that has been issued for each resource in the group, but the sample CEUS does not display that information.

When you customize the sample CEUS for your ALCS system, you may wish to enhance the display of group information. Alternatively, you may wish the CEUS user to start a second 3270 session with ALCS and use that session for ZOCTM commands.

- **Other limitations in sample CEUS**

  There are a number of minor limitations which you may wish to review when customizing the sample CEUS for your ALCS system. The function for displaying a list of the allocated communications groups displays the first 86, although ALCS allows a maximum of 400 (the COMTC macro and the ZOCTM GROUPS command both provide all the group names). The function for displaying (showing) a communications resource will display only those resources managed by OCTM, although COMTC QUERY will provide information on any communication resource (the ZDCOM command can be used to display any resource). The function for unallocating a communications group will not allow a group that contains uncommitted change requests to be unallocated (the COMTC UNALLOCATE macro provides the DELETE=YES option for allowing a group to be unallocated and all its associated change requests to be cancelled).

Additional CEUS Functionality

There are additional functions that could be included in the CEUS implemented by ALCS users. The following provides some suggestions on this.

Maintaining CEUS control records

The CEUS could maintain control records that would enable information to be stored on the ALCS database and retained over a group of CEUS transactions. Information about COMTC macros that have been issued for a communication resource or group could be saved, enabling additional validation checks to be performed. For example. when a system administrator allocates a communications group, if they had been required to logon with a user-ID (and password), the CEUS could associate the group with their user-ID and save this information in a CEUS control record. The CEUS could then verify that each change request that is submitted for that group is input by the same person (same user-ID). CEUS control records could also be used for maintaining an audit trail of the change requests submitted for a communications group (type of change request, time/date of change request, etc.).

Restricting usage of CEUS functions

The ALCS access control function could be used to restrict the usage of the CEUS to authorized system administrators. There may be some CEUS functions (for example, displaying information about a communication resource) that do not need to be restricted, but other functions may need to be restricted. A RACF ALCSAUTH profile could be created for the CEUS and system administrators whose user-IDs have access to that profile could be allowed to perform the restricted functions. The system administrator would be required to logon to the system with a user-ID and password before being given the initial screen map by the CEUS. The CEUS would then use the AUTHC macro to check a user's authority before performing specific functions.

Managing communications groups

The OCTM facility places some restrictions on the usage of communications groups, but additional restrictions may be required. For example:

- The total number of groups that can be active at any time could be restricted to 50 (OCTM allows 400).
- The total number of groups that each system administrator can have active at any one time could be restricted to 10.
- The total number of change requests that can be submitted for a group could be restricted to 100 (OCTM allows 400). If terminals are being deleted in a group, you may wish to restrict the number of deletes to 50.

Some of these restrictions could be managed by providing a CEUS control record for each communications group that is allocated. That control record could hold the user-ID of the person responsible for the group and keep an audit trail of every COMTC issued for the group. A senior system administrator could display the information in the control record when monitoring the status of active groups.

Managing CRI's and ordinals

The CEUS can request the OCTM facility to allocate an available CRI and ordinal number for a new communication resource (it can also explicitly define the CRI and ordinal number for a new resource). For example, if a terminal is being added by a COMTC ADD macro, the CRIBASE= and ORDBASE= parameters can be used to request OCTM to allocate an available CRI and ordinal.

If the base CRI address is defined as CRIBASE=020001, OCTM will allocate the next available CRI starting from 020001. Different CRIBASE values could be used for allocating CRI addresses for different resource types. For example, if CRIBASE=250000 is used on a COMTC ADD for an "other system" terminal, OCTM would allocate the next available CRI starting from 250000. Each ALCS installation has its own rules on the allocation of CRI addresses, but the CRI= and CRIBASE= parameters on COMTC ADD should provide ample flexibility on the selection of CRI's for new communication resources.

In a similar way, you can control the ordinal numbers allocated to new communication resources. The ORDBASE parameter requests OCTM to allocate the next available ordinal number. If ORDBASE=500 is used, OCTM will allocate the next available ordinal number starting at ordinal 500.

Using the OCTM policing exit

The OCTM policing function is activated once each hour for monitoring the contents of the base and update communications areas on the OCTM database. The primary policing function is to monitor the status of communication resources and groups that currently have change requests in progress for them. If the policing function identifies a period of inactivity (since the last COMTC macro was issued for the communication resource or group), it sends a warning message to the RO Cras indicating that further action is required. The policing function also activates the ECB-controlled installation-wide exit AOCM. The AOCM exit program can be used to perform additional functions. For example, it can suppress the transmission of warning messages and can issue a COMTC macro to activate the next COMTC function for the communication resource or group.

For a detailed description of the AOCM installation-wide exit, see ALCS Installation and Customization.

The AOCM exit program is activated when there has been a period of inactivity for either a communications resource or a communications group. For example, if a communications group with five active change requests has been loaded (but not confirmed), and 48 hours has elapsed since the COMTC LOAD was issued, the policing exit program is activated. The policing function activates the exit program whenever there has been a period of inactivity for a communications resource or group that is waiting for either the COMTC LOAD, COMTC CONFIRM, COMTC COMMIT or COMTC UNALLOCATE macro to be issued. The exit is initially activated after a 48 hour period of inactivity, but it will be activated again after 56 hours, 64 hours, and every 8 hours after that until the required COMTC macro is issued. After each period of inactivity, the policing function sends a warning message to RO Cras stating which COMTC macro (load, confirm, etc.) is outstanding for the resource or group. The policing exit can suppress the transmission of the warning message, and/or issue the COMTC macro that the resource or group is waiting for. For example, if a COMTC COMMIT had been issued for a communications group, but the group had not yet been unallocated, after 48 hours the policing exit would be activated for this communications group. On this activation, the exit could just return to ALCS and allow the warning message to be sent to RO Cras. If the CEUS has still not unallocated the communications group after 56 hours, the exit will be activated again, and this time the exit could suppress transmission of the warning message and issue a COMTC UNALLOCATE to request OCTM to perform the unallocate function for that group.

Adding and changing communication resources

The ALCS communications generation function provides the COMDFLT macro for defining default values for groups of communication resources. Some default values are suitable for all the resources that belong to a specific communication type (for example, VTAM 3270). Some ALCS installations do not use the IBM defaults provided on the COMDEF macros and have their own set of defaults which they define on COMDFLT. When new resources are being added, the CEUS could display a screen map that included default values for specific fields, defaults which match those provided in the COMDFLT generation macro. When the CEUS is used for adding new communication resources or changing current resources, you may want to restrict the values that can be input in specific fields. You could therefore validate these fields in your CEUS application and reject any value which does not conform to your installation standards.

COMTC ADD and COMTC REPLACE allow alternate CRAS status to be defined for a terminal resource. This can be high CRAS status (AT1 to AT16 and AP1 to AP16) or low CRAS status. The CEUS could inhibit the allocation of CRAS status to those CEUS users who have appropriate CRAS authority. For example, if a CEUS user has high CRAS authority, they could be allowed to allocate any CRAS status, but if they have low CRAS authority they could allocate only low CRAS status.

Activating communications change requests

Communication change requests are activated using the COMTC LOAD, COMTC CONFIRM and COMTC COMMIT macros. A three stage process has been provided so that if incorrect change requests are submitted to the OCTM facility by the CEUS, there is an opportunity to detect this and back out the changes (via COMTC BACKOUT). The sample CEUS application has kept these three stages separate, requiring the system administrator to specifically request the load, confirm and then commit functions (via separate options). When a communications group is used, a COMTC UNALLOCATE is also required after the commit, making it a four stage process.

ALCS users may prefer to combine some of these functions together, therefore reducing the number of actions to be taken by the system administrator. For example, the load and confirm functions could be combined, the confirm and commit functions could be combined, the commit and unallocate functions combined, etc. The following provides some suggestions on this.

- **Combined load and confirm functions**

  The CEUS can request OCTM to load a single change request or a group of change requests. The COMTC LOAD macro is used to load the change request(s) into the online communication table. After a successful load, the CEUS could immediately use the COMTC CONFIRM macro to request OCTM to confirm the change request(s). After a confirm, the change request(s) are retained over an ALCS restart. A back out of the change request(s) is still possible after the confirm. The commit (and unallocate, if required) could be issued a few hours later when the change request(s) have been verified.

- **Combined confirm and commit functions**

  With this option, the COMTC LOAD is issued on its own. What are the benefits and disadvantages of issuing the COMTC LOAD on its own? The primary benefit is that if a major problem occurred because of the load, it would have no effect on the system during the next ALCS restart. The disadvantage is that if the ALCS system fails, the change request(s) are not re-loaded on ALCS restart and another load request (COMTC LOAD) is required after the restart.

- **Combined commit and unallocate functions**

  If the load and confirm functions have been combined together (for a communications group) you may wish to also combine the commit and unallocate functions. When a communications group has been committed, there will normally be no further requirement to retain the group, therefore the CEUS could issue the COMTC COMMIT and follow it immediately with a COMTC UNALLOCATE.

Inactivating and Activating Communication Resources

The COMTC LOAD and COMTC BACKUP macros update resources in the online communication table. If the communication resources being loaded or backed out are in the active (logged on) status in the online communication table, then the load (or back out) will fail. The CEUS user could use the ZACOM command to inactivate these resources, but you may prefer to perform the inactivation within the CEUS via the COMCC macro. You may also wish to consider the requirement to activate new or modified resources that have been loaded into the online communication table. The CEUS user could use ZACOM to activate the resources, but you may prefer to perform the activation within the CEUS via the COMCC macro. The following suggests how the CEUS could perform the inactivation and activation of resources.

- **Inactivation before a COMTC LOAD**

  When communication resources are being **replaced** or **deleted**, immediately prior to issuing the COMTC LOAD macro, use the COMIC macro to determine if the resource is active (test field ICESCST after the COMIC). If the resource is active, use the COMCC macro with FIELD=REC1IFR to inactivate (logoff) the resource.

- **Inactivation before a COMTC BACKOUT**

When communication resources have been **added** or **replaced**, immediately prior to issuing the COMTC BACKOUT macro, use the COMIC macro to determine if the resource is active (test field ICESCST after the COMIC). If the resource is active, use the COMCC macro with FIELD=REC1IFR to inactivate (logoff) the resource.

- **Activation after a COMTC LOAD**

  When communication resources are being **added** or **replaced**, immediately after a successful return from the COMTC LOAD macro, use the COMCC macro with FIELD=REC1ISR to activate (logon) the resource. If your ALCS system provides a terminal control record (for example, a AAA record) for each terminal, and a new terminal is being added, the CEUS could also perform initialization of the terminal control record for the terminal (or activate a utility program that performs the initialization).

- **Activation after a COMTC BACKOUT**

  When communication resources have been **replaced** or **deleted**, immediately after a successful return from the COMTC BACKOUT macro, use the COMCC macro with FIELD=REC1ISR to activate (logon) the resource.

ALCS places restrictions on the usage of COMCC when the macro is used to inactivate or activate a resource. The terminal must have CRAS status or the CEUS user must have CRAS authority (any CRAS, AT1 to AT255). If the COMCC macro is to be used for inactivation and activation, then include a COMIC macro in the CEUS input message editor program to verify CRAS status or authority before processing any requests that submit or activate OCTM change requests.

Managing impact of the UCOMCHG Callable Service

ALCS provides a callable service called UCOMCHG that can be used by monitor exits (for example, USRCOM6 monitor exit) to change resource names (CRN's) in the online communication table. When this callable service is used, the CRN is changed only in the online communication table, therefore the resource continues to use the previous CRN in the OCTM database.

If your ALCS system uses the UCOMCHG service, then it should also contain a batch of resources (normally terminal resources) that are for the specific use of UCOMCHG. These may be resources that are activated (logged on to) only after the UCOMCHG service has changed the resource name. Because these resources have a special purpose in your ALCS system, the CEUS could perform specific checks on them. For example, If the CEUS user attempts to change the definition of one of these resources, and the name of the resource has already been changed by the UCOMCHG service, then the CEUS could reject that change. Alternatively, you may wish to enhance the resource name validation in the CEUS prior to issuing the COMTC QUERY, REPLACE and DELETE macros to determine if the resource belongs to the group dedicated to UCOMCHG usage. If it is, the CEUS could reject the action or limit the actions that are allowed.

Appendix B - COMTC Communication Table Update Macro

The COMTC macro allows the Communications End User System (CEUS) to obtain information about communication resources, submit change requests for those resources and to permanently update them in the OCTM database and the online communication table.

The COMTC macro provides twelve different actions that can be requested by the CEUS. This section provides a detailed description of each action that is provided by the COMTC macro.

COMTC QUERY – Query status of communication resource or group

Format

```
[label]  COMTC ACTION=QUERY,{CRN={field|(reg1)}|CRI={field|(reg2)}|GROUP={field|(reg3)}}
               ,LEVEL={Dn|(reg4)}
```

label
Any valid assembler label.

CRN={*field*|(*reg1*)}
The Communication Resource Name (CRN) of the communication resource for which status information is required. The CRN must be an alphanumeric string of 1 to 8 characters. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of an 8-byte field that contains the CRN.
(*reg1*) Register containing the address of an 8-byte field that contains the CRN. Use general registers 0 through 7.

CRI={*field*|(*reg2*)}
The Communication Resource Identifier (CRI) of the communication resource for which status information is required. The CRI must be 6 hexadecimal digits. Specify one of:

field Assembler label of a 3-byte field that contains the CRI.
(*reg2*) Register containing the address of a 3-byte field that contains the CRI. Use general registers 0 through 7.

GROUP={*field*|(*reg3*)}
The name of a communications group for which status information is required. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of a 7-byte field that contains the name of the communications group.
(*reg3*) Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

LEVEL={D*n*|(*reg4*)}
An available storage level in the Entry Control Block (ECB). A 4000-byte storage block (size L3) containing status information on a communication resource or a communications group will be provided at this storage level. Specify one of:

D*n* Level symbol: **D0**, **D1**, and so on up to **DF** for level 15.
(*reg4*) Register containing the level value (use LA *reg*,**D***n*). Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=QUERY** option to query the status and obtain information about a communication resource or a communications group.

Use the **CRN=** or **CRI=** parameter to obtain information about a specific communication resource. When the definition of a communication resource is to be changed (via **COMTC REPLACE**), use COMTC QUERY to obtain the current definition for the communication resource. COMTC QUERY can also be used to obtain details of the owning LU for a specific communication resource before changes are applied via **COMTC REPLACE** to the resource itself. ALCS extracts information about the communication resource from the Online Communication Table Maintenance (OCTM) database or from the online communication table.

Alternatively, use the **GROUP=** parameter to obtain information about a communications group that is in use on the OCTM database. ALCS provides status information and a list of the communication resources that belong to the communications group.

When COMTC QUERY is used with the **CRN=** or **CRI=** parameters, a symbolic CRAS address must not be used (for example, you can not use CRN=ATnnn). On return from COMTC QUERY, called with the **CRN=** or **CRI=** parameter, a 4000-byte storage block will reside at the ECB storage level specified by the **LEVEL=** parameter. This storage block will contain detailed information about the communication resource. If a communication change request is currently being processed for the resource, details of the change request and its status are also provided. Use the **Communications Resource and Group Information DSECT** (**CT2TM**) to reference the contents of this storage block.

On return from COMTC QUERY, called with the **GROUP=** parameter, a 4000-byte storage block will reside at the ECB storage level specified by the **LEVEL=** parameter. This storage block will contain a list of the communication resources that belong to this communications group and its current status on the OCTM database. Use the **Communications Resource and Group Information DSECT** (**CT2TM**) to reference the contents of this storage block.

Register use

On return from COMTC QUERY, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK	Processing completed without error
COMTC_R_RETRY	Unable to obtain lock for communication resource or group - retry the COMTC QUERY macro immediately as this is only a temporary condition
COMTC_R_ERROR	An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.
COMTC_R_SYSERR	Unable to process the COMTC QUERY macro because of system error - check RO CRAS for a system error dump

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN The communication resource name on the **CRN=** parameter is invalid

COMTC_S_NCRN The communication resource name on the **CRN=** parameter is unknown

COMTC_S_ACRN The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_NCRI The communication resource identifier on the **CRI=** parameter is unknown

COMTC_S_ACRI The address of the field that the **CRI=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_EGRP No communication resources are associated with the communications group defined on the **GROUP=** parameter

COMTC QUERY may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC QUERY will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=QUERY** to obtain all the information about a specific communication resource in a storage block on ECB storage level 3 (D3).

```
         COMTC ACTION=QUERY,       ISSUE QUERY REQUEST
               CRN=EBW008,         CRN IS IN EBW008
               LEVEL=D3            RETURN COMMS RESOURCE DATA ON LEV 3
```

The following example shows how you could use the COMTC macro with **ACTION=QUERY** to obtain a list of communication resources (in a storage block on ECB storage level 4) that belong to a specific communications group on the OCTM database.

```
         CT2TM REG=R05             DEFINE BASE FOR CT2TM BLOCK
QUERYREQ EQU   *
         COMTC ACTION=QUERY,       ISSUE QUERY REQUEST
               GROUP=(R06),        GROUP NAME POINTED TO BY REGISTER 6
               LEVEL=D4            RETURN COMMS RESOURCE LIST ON LEV 4
         LR    R14,R15             LOAD RETURN CODE / REASON CODE
         SLL   R14,16              ISOLATE RETURN CODE
         SRL   R14,16              SHIFT TO LOW-ORDER BYTES
         B     *+4(R14)            BRANCH ON RETURN CODE
         B     QRC0                0 - OK
         B     QRC4                4 - RETRY IMMEDIATELY
         B     QRC8                8 - USER ERROR
         B     QRC12               12 - SYSTEM ERROR
QRC4     EQU   *
         DEFRC ,                   WAIT A MOMENT
         B     QUERYREQ            RETRY THE COMTC REQUEST
QRC0     EQU   *
         L     R05,CE1CR4          LOAD BASE ADDRESS OF CT2TM BLOCK
         L     R04,CT2CNT          LOAD COUNT OF RESOURCES IN GROUP
         LTR   R04,R04             IS COUNT ZERO?
         BZ    GRPEMPTY            IF YES - BRANCH
         LR    R03,R05             CT2TM BASE IN REGISTER 3
         A     R03,CT2LST          ADD DISPLACEMENT TO CRN'S
         LA    R02,MSGCRN          POINT TO CRN MSG DISPLAY FIELD
RESRLOOP EQU   *
         MVC   0(8,R02),0(R03)     MOVE CRN TO MSG DISPLAY AREA
         .....
         LA    R02,16(R02)         INCREMENT BASE OF MSG DISPLAY AREA
         LA    R03,9(R03)          INCREMENT BASE OF CRN LIST IN CT2TM
         BCT   R04,RESRLOOP        GO BACK TO TOP OF LOOP
         .....
         .....
GRPEMPTY EQU   *
```

Related information

"CT2TM – Communications Resource and Group Information DSECT" on page 123.

COMTC GROUPS – Obtain status of communications groups

Format

```
[label]  COMTC ACTION=GROUPS,LEVEL={Dn|(reg)}
               [,SORT={NAME|DATE}]
```

label
Any valid assembler label.

LEVEL={D*n*|(*reg*)}
An available storage level in the Entry Control Block (ECB). A 4000-byte storage block (size L3) containing information about the allocated communications groups will be provided at this storage level. Specify one of:

D*n*	Level symbol: **D0**, **D1**, and so on up to **DF** for level 15.
(*reg*)	Register containing the level value (use LA *reg*,**D***n*). Use general registers 0 through 7.

SORT={NAME|DATE}
Specifies how the communications group information should be sorted. Specify one of:

NAME	Sort the communications group information in alphabetic sequence based on group name. This is the default.
DATE	Sort the communications group information in date and time sequence based on the date and time the group was allocated, with the oldest group first.

Description

Use the COMTC macro with the **ACTION=GROUPS** option to obtain information about all the allocated communications groups.

On return from COMTC GROUPS, a 4000-byte storage block will reside at the ECB storage level specified by the **LEVEL=** parameter. This storage block provides information on a maximum of 100 communications groups. The information includes the group name, the time/date when the group was allocated, the latest action performed for that group, etc. If there are more than 100 allocated groups, L3ST pool records will be chained to this storage block. Each pool record will hold information on a further 100 allocated groups. The number of L3ST pool records forward chained from this storage block will depend on the number of groups currently allocated. Use the **Communications Groups Information DSECT** (**CT3TM**) to reference the contents of this storage block and the overflow pool records. You should ensure that the L3ST pool records are released before the entry terminates processing.

Register use

On return from COMTC GROUPS, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK	Processing completed without error

COMTC_R_RETRY Unable to obtain lock for communication resource or group - retry the COMTC GROUPS macro immediately as this is only a temporary condition

COMTC_R_ERROR An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC GROUPS macro because of system error - check RO CRAS for a system error dump

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_EGRS No communications groups exist

COMTC GROUPS may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC GROUPS will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=GROUPS** to obtain information about all the communications groups in a storage block on ECB storage level 3 (D3).

```
         CT3TM REG=R07             DEFINE BASE FOR CT3TM BLOCK
GRPQUERY EQU   *
         COMTC ACTION=GROUPS,      ISSUE GROUPS REQUEST
               LEVEL=D3,           RETURN COMMS GROUP DATA ON LEV 3
               SORT=DATE           SORT GROUPS INTO DATE SEQUENCE
         LR    R14,R15             LOAD RETURN CODE / REASON CODE
         SLL   R14,16              ISOLATE RETURN CODE
         SRL   R14,16              SHIFT TO LOW-ORDER BYTES
         B     *+4(R14)            BRANCH ON RETURN CODE
         B     GRC0                0  - OK
         B     GRC4                4  - RETRY IMMEDIATELY
         B     GRC8                8  - RETRY LATER
         B     GRC12               12 - USER ERROR
         B     GRC16               16 - SYSTEM ERROR
GRC4     EQU   *
         DEFRC ,                   WAIT A MOMENT
         B     GRPQUERY            RETRY THE COMTC REQUEST
GRC0     EQU   *
         L     R07,CE1CR3          LOAD BASE ADDRESS OF CT3TM
         L     R03,CT3CNT          LOAD COUNT OF GROUPS IN BLOCK
         LTR   R03,R03             IS COUNT ZERO?
         BZ    NOGROUP             IF YES - BRANCH
ITEMLOOP EQU   *
         TM    CT3STA,CT3CHNG      CHANGE REQUESTS NOT YET LOADED?
         BO    CHANGE              IF YES - BRANCH
         TM    CT3STA,CT3LOAD      CHANGE REQUESTS LOADED?
         BO    LOADED              IF YES - BRANCH
         TM    CT3PRG,CT3LDPR      IS LOAD IN PROGRESS?
         BO    LOADED              IF YES - BRANCH
         ...
         ...
BOTLOOP  EQU   *
         LA    R06,L'CT3ITM        LOAD LENGTH OF ITEM
         AR    R07,R06             INCREMENT TO NEXT ITEM
         BCT   R03,ITEMLOOP        GO BACK TO TOP OF LOOP
         B     FCHCHECK            GO CHECK FOR FORWARD CHAIN
```

Related information

"CT3TM – Communications Groups Information DSECT" on page 128.

COMTC ALLOCATE – Allocate a new communications group

Format

```
[label]  COMTC ACTION=ALLOCATE,GROUP={field|(reg1)}
```

label
Any valid assembler label.

GROUP={*field*|(*reg1*)}
The name of a new communications group that is to be allocated on the Online Communication Table Maintenance (OCTM) database. The name of the communications group must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field	Assembler label of a 7-byte field that contains the name of the new communications group.
(*reg1*)	Register containing the address of a 7-byte field that contains the name of the new communications group. Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=ALLOCATE** option to allocate the name of a new communications group.

The COMTC macro can be used to submit a wide range of communications change requests to ALCS. For example, you can modify a current communication resource using **COMTC REPLACE**, you can add new communication resources using **COMTC ADD**, and you can delete obsolete communication resources using **COMTC DELETE**. Use COMTC with **ACTION=ALLOCATE** to assign a communications group name to a batch of communication change requests that will be submitted to ALCS via the COMTC macro. When you have submitted all the communication change requests for your communications group, you can use a single COMTC macro to *load* all those change requests (using the communications group name in the **COMTC LOAD** macro). You can subsequently use COMTC macros to *confirm* and *commit* the change requests for that communications group (or use COMTC to *backout* the change requests).

Register use

On return from COMTC ALLOCATE, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK	Processing completed without error
COMTC_R_RETRY	Unable to obtain lock for communication resource or group - retry the COMTC ALLOCATE macro immediately as this is only a temporary condition
COMTC_R_ERROR	An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC ALLOCATE macro because of system error - check RO CRAS for a system error dump

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_TGRP The maximum number of permitted communications groups have already been allocated. Use COMTC GROUPS to obtain a list of the communications groups and unallocate those groups that have been *committed*.

COMTC_S_DGRP The communications group name that the **GROUP=** parameter references is already in use

COMTC ALLOCATE may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC ALLOCATE will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=ALLOCATE** to allocate the name of a new communications group.

```
         MVC   EBW016(7),=C'GRPFIVE' LOAD NAME OF COMMS GROUP
         COMTC ACTION=ALLOCATE,      ISSUE ALLOCATE REQUEST
               GROUP=EBW016          GROUP NAME IS IN EBW016
```

Related information

"COMTC UNALLOCATE – Unallocate a communications group" on page 91.

COMTC ADD – Add a new communication resource

Format

```
[label]  COMTC ACTION=ADD,CRN={field|(reg1)}
               ,{CRI={field|(reg2)}|CRIBASE={field|(reg3)}}
               ,{ORD={field|(reg4)}|ORDBASE={field|(reg5)}}
               ,COMSDAT={field|(reg6)}
               ,LEVEL={Dn|(reg7)}
               [,GROUP={field|(reg8)}]
```

label
Any valid assembler label.

CRN={*field***|(***reg1***)}**
The Communication Resource Name (CRN) of the new communication resource. The CRN must be an alphanumeric string of 1 to 8 characters, it must be unique within ALCS and the first character must be alphabetic. If the CRN is less than 8 characters, it should be left justified and padded with spaces. For VTAM 3270 terminal resources and X.25 PVC resources, the CRN must be the LU name as defined to VTAM.

The CRN for an X.25 ALC terminal resource is comprised of a base CRN plus one or two other elements depending on the type of X.25 PVC that the X.25 ALC terminal is associated with. There are three types of X.25 PVC that can have associated X.25 ALC terminals. The length of the base CRN for these terminals is dependent on the type of X.25 PVC that they are associated with.
For an X.25 ALC terminal that is associated with a ***Type 1*** X.25 PVC, the base CRN must be an alphanumeric string of 1 to 6 characters. You must append a two-character terminal address to the 6-character base CRN to create the full CRN.
For an X.25 ALC terminal that is associated with a ***Type 6*** or ***Type 7*** X.25 PVC, the base CRN must be an alphanumeric string of 1 to 4 characters. You must append a two-character interchange address followed by a two-character terminal address to the 4-character base CRN to create the full CRN.
The X.25 ALC terminal and interchange addresses that are used to create the full CRN must also be supplied in the **Communication Resource Definition DSECT, CT1TM**. The terminal address must be provided in field **CT1CTA** and the interchange address must be provided in field **CT1CIA**. Use the **COMSDAT=** parameter to specify the storage area that contains the CT1TM DSECT.
Specify one of:

field	Assembler label of an 8-byte field that contains the CRN of the new communication resource.
(*reg1*)	Register containing the address of an 8-byte field that contains the CRN of the new communication resource. Use general registers 0 through 7.

CRI={*field***|(***reg2***)}**
The Communication Resource Identifier (CRI) of the new communication resource. The CRI address must be 6 hexadecimal digits and it must be unique within ALCS. Specify one of:

field	Assembler label of a 3-byte field that contains the CRI address.

(*reg2*) Register containing the address of a 3-byte field that contains the CRI address. Use general registers 0 through 7.

CRIBASE={*field***|(***reg3***)}**
Base Communication Resource Identifier. ALCS will select an available CRI for the new communication resource. Specify one of:

field Assembler label of a 3-byte field that contains the base CRI address. It must be 6 hexadecimal digits and must be in the range **020000** to **FFFFFE**. The first available CRI whose address is equal to or higher than the base CRI is allocated to the new communication resource.

(*reg3*) Register containing the address of a 3-byte field that contains the base CRI address. Use general registers 0 through 7. It must be 6 hexadecimal digits and must be in the range **020000** to **FFFFFE**. The first available CRI whose address is higher than the base CRI is allocated by ALCS to the new communication resource.

ORD={*field***|(***reg4***)}**
The communication resource ordinal number for the new communication resource. The resource ordinal number is a hexadecimal number. Specify one of:

field Assembler label of a fullword field that contains the resource ordinal number in hexadecimal.

(*reg4*) Register containing the address of a fullword field that contains the resource ordinal number in hexadecimal. Use general registers 0 through 7.

ORDBASE={*field***|(***reg5***)}**
Base communication resource ordinal number for the new communication resource. The base resource ordinal number is a hexadecimal number. ALCS will select an available resource ordinal number for the new communication resource. Specify one of:

field Assembler label of a fullword field that contains the base resource ordinal number in hexadecimal. The first available ordinal number that is equal to or higher than the base resource ordinal number is allocated by ALCS to the new communication resource.

(*reg5*) Register containing the address of a fullword field that contains the base resource ordinal number in hexadecimal. Use general registers 0 through 7. The first available ordinal number that is higher than the base resource ordinal number is allocated by ALCS to the new communication resource.

COMSDAT={*field***|(***reg6***)}**
The communications data that defines the characteristics of the new communication resource. This data must be provided in a storage area that has been formatted using the **Communication Resource Definition DSECT, CT1TM**. Specify one of:

field Assembler label of a storage area that contains the communications data formatted by the **CT1TM** DSECT.

(*reg6*) Register containing the address of a storage area that contains the communications data formatted by the **CT1TM** DSECT. Use general registers 0 through 7.

LEVEL={D*n***|(***reg7***)}**
An available storage level in the Entry Control Block (ECB). An L3 size storage block is attached by ALCS to this storage level for recording any errors that are detected in the communication resource data that is provided (via the **COMSDAT=**

parameter) for this new communications resource. The storage block can hold a maximum of 49 error messages. Specify one of:

D*n*	Level symbol: **D0**, **D1**, and so on up to **DF** for level 15.
(*reg7*)	Register containing the level value (use LA *reg*,**D***n*). Use general registers 0 through 7.

GROUP={*field*|(*reg8*)}

The name of the communications group that this new communication resource belongs to. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field	Assembler label of a 7-byte field that contains the name of the communications group.
(*reg8*)	Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=ADD** option to add a new communication resource to the Online Communication Table Maintenance (OCTM) database. A single COMTC ADD should be used to provide all the details of the new communication resource. The new communication resource can be a terminal (except Prime and RO CRAS) or an X.25 PVC (types 1, 6 or 7).

Use the **CRN=** parameter to specify the Communications Resource Name that will be used to identify this new communication resource.

Use the **CRI=** parameter to specify the Communications Resource Identifier that will be used to identify this new communication resource. Alternatively, use the **CRIBASE=** parameter to request ALCS to select an available CRI for this new communication resource.

Use the **ORD=** parameter to specify the communication resource ordinal number for this new communication resource. Alternatively, use the **ORDBASE=** parameter to request ALCS to select an available communication resource ordinal number for this new communication resource.

Use the **COMSDAT=** parameter to point to a storage area containing detailed information about the new communication resource. The **Communication Resource Definition DSECT** (**CT1TM**) defines the layout of this storage area and the individual fields within it. Initialize this storage area to hexadecimal zeroes and then place the detailed information about the new communication resource in the relevant fields. ALCS systems that use the communications user data area can provide the required user data for the new communication resource in the CT1TM DSECT. The description of the **CT1TM** DSECT provides guidance on the fields that require data for each type of communication resource.

On return from COMTC ADD, an L3 size storage block resides at the ECB storage level specified by the **LEVEL=** parameter. During validation of the communication resource data provided in the **CT1TM** DSECT, if errors are detected, appropriate error messages are placed in this storage block. Test the return code in register 15 to determine if any errors were found. The storage block will contain a maximum of 49 error messages. If more than 49 errors are found, ALCS will report only the first 49. Each error message will be a maximum of 80 bytes, therefore the storage block is formatted into 80-byte items. A 4-byte header resides at the beginning of the block containing a count of the error messages that have been placed in the block. Immediately following the header is the first 80-byte error message item. The error message text is in EBCDIC and the message contains a prefix **DXC***nnnn*, where the message number *nnnn* is in the range 9100 to 9139. Details of these error messages can be found in ALCS Messages and Codes.

When a batch of new communication resources are being added, allocate a ***communications group name*** to that batch. The **COMTC ALLOCATE** macro allows you to do this. Although a separate COMTC ADD is required for adding each new communication resource, when you use a communications group to add a batch of new resources, it enables you to use a single **COMTC LOAD** macro for loading the batch of new resources on to the ALCS online communication table. Use the **GROUP=** parameter on COMTC ADD to identify the name of the communications group that the new communication resource belongs to.

X.25 PVC and ALC resources: ALCS supports different types of X.25 PVC communication resources, including those for Type-A traffic, Type-B traffic and host-to-host traffic. Use COMTC ADD for adding X.25 PVC communication resources for Type-A traffic. Other types of X.25 PVC resources must be defined to ALCS in the communication generation COMDEF generation macros.

An X.25 ALC terminal resource will always be associated with an X.25 PVC resource that manages Type-A traffic. The X.25 PVC resource must therefore always be defined to ALCS via a COMTC ADD before its associated X.25 ALC terminals. An X.25 PVC and its associated ALC terminals could be added in a single X.25 *communications group*. Use **COMTC ALLOCATE** to allocate a communications group name for the X.25 PVC and its associated X.25 ALC terminals. Use COMTC ADD to add each individual X.25 resource, and then issue COMTC macros to *load, confirm* and *commit* this X.25 communications group.

Register use

On return from COMTC ADD, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK Processing completed without error

COMTC_R_RETRY Unable to obtain lock for communication resource or group - retry the COMTC ADD macro immediately as this is only a temporary condition

COMTC_R_ERROR An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC ADD macro because of system error - check RO CRAS for a system error dump

COMTC_R_DATAERR The communications data provided in the **CT1TM** DSECT contained errors. A storage block at the ECB storage level specified by the **LEVEL=** parameter contains error messages describing up to 49 different errors in the communications data.

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN The communication resource name on the **CRN=** parameter is invalid

COMTC_S_ACRN The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_ACRI The address of the field that the **CRI=** or **CRIBASE=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_IPGS A previous COMTC macro for this communication resource or group failed to complete. Use COMTC QUERY to determine the status of the communication resource or group.

COMTC_S_ESEQ There are change requests for this communication resource or group that are in the *confirmed* status

COMTC_S_TGRP The maximum number of permitted non-group change requests have already been submitted. Identify non-group change requests that are waiting to be committed and use COMTC COMMIT to commit them.

COMTC_S_TCHG The maximum number of permitted change requests have already been submitted for the communications group defined in the **GROUP=** parameter

COMTC_S_ADAT The address of the field that the **COMSDAT=** parameter references is invalid

COMTC_S_DCRN The communication resource name on the **CRN=** parameter is already in use

COMTC_S_DCRI The communication resource identifier on the **CRI=** parameter is already in use

COMTC_S_DORD The communication resource ordinal number that the **ORD=** parameter references is already in use

COMTC_S_AORD The address of the field that the **ORD=** or **ORDBASE=** parameter references is invalid

COMTC_S_RCRI The communication resource identifier on the **CRI=** parameter is within the range of CRI's that are for the exclusive use of the ALCS communications generation (as defined in the **COMGEN CRIRANGE=** parameter)

COMTC_S_ICRI The **CRIBASE=** parameter was used to obtain the next available CRI, but there were no available CRI addresses found between this base CRI address and the maximum CRI address

COMTC_S_IORD The communication resource ordinal number is invalid.
If a base ordinal number has been provided in the **ORDBASE=** parameter, then no available ordinal numbers have been found between this base ordinal number and the maximum ordinal number.
If an ordinal number has been provided in the **ORD=** parameter, then that ordinal number is higher than the maximum ordinal number specified in the communications generation (as defined in the **COMGEN MAXORD=** parameter).

COMTC_S_RORD The communication resource ordinal number that the **ORD=** parameter references is within the range of ordinals that are for the exclusive use of the ALCS communications generation (as defined in the **COMGEN ORDRANGE=** parameter)

COMTC_S_FORD The communication resource ordinal number that the **ORD=** parameter references is invalid because it contains hexadecimal zeroes.

When the return code **COMTC_R_DATAERR** is reported in general register 15, ALCS will have detected errors in the communication data provided in the **CT1TM** DSECT. To obtain a list of the errors that have been detected, examine the contents of the storage block that resides at the data level specified by the **LEVEL=** parameter. The storage block contains error messages describing the error conditions that have been detected. There can be a maximum of 49 error messages in the storage block. The previous section provides information on the format of this storage block.

COMTC ADD may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC ADD will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=ADD** to define a new communication resource that is being added as part of a communications group.

```
COMTC ACTION=ADD,         ISSUE ADD REQUEST
      CRN=EBW008,         CRN IS IN EBW008
      CRIBASE=EBW016,     SELECT AN AVAILABLE CRI
      ORDBASE=EBW020,     SELECT AN AVAILABLE ORDINAL
      COMSDAT=EBX000,     COMMS DATA IN EBX000
      LEVEL=D4,           ECB LEVEL FOR ERROR MSG BLOCK
      GROUP=EBW024        GROUP NAME IN EBW024
```

Related information

“CT1TM – Communication Resource Definition DSECT” on page 96.

COMTC REPLACE – Change a communication resource

Format

```
[label]  COMTC ACTION=REPLACE,{CRN={field|(reg1)}|CRI={field|(reg2)}}
               ,COMSDAT={field|(reg3)}
               ,LEVEL={Dn|(reg4)}
               [,GROUP={field|(reg5)}]
```

label
Any valid assembler label.

CRN={*field*|(*reg1*)}
The Communication Resource Name (CRN) of the communication resource that is being changed. The CRN must be an alphanumeric string of 1 to 8 characters. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of an 8-byte field that contains the CRN.
(*reg1*) Register containing the address of an 8-byte field that contains the CRN. Use general registers 0 through 7.

CRI={*field*|(*reg2*)}
The Communication Resource Identifier (CRI) of the communication resource that is being changed. The CRI address must be 6 hexadecimal digits. Specify one of:

field Assembler label of a 3-byte field that contains the CRI address.
(*reg2*) Register containing the address of a 3-byte field that contains the CRI address. Use general registers 0 through 7.

COMSDAT={*field*|(*reg3*)}
A storage area that contains the changes to be applied to the communication resource. This storage area must be formatted using the **Communication Resource Definition DSECT**, **CT1TM**. Specify one of:

field Assembler label of a storage area formatted by the **CT1TM** DSECT.
(*reg3*) Register containing the address of a storage area formatted by the **CT1TM** DSECT. Use general registers 0 through 7.

LEVEL={D*n*|(*reg4*)}
An available storage level in the Entry Control Block (ECB). An L3 size storage block is attached by ALCS to this storage level for recording any errors that are detected in the communication resource data that is provided (via the **COMSDAT=** parameter) for this communications resource. The storage block can hold a maximum of 49 error messages. Specify one of:

D*n* Level symbol: **D0**, **D1**, and so on up to **DF** for level 15.
(*reg4*) Register containing the level value (use LA *reg*,**D***n*). Use general registers 0 through 7.

GROUP={*field*|(*reg5*)}
The name of the communications group that this communication resource belongs to. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field	Assembler label of a 7-byte field that contains the name of the communications group.
(*reg5*)	Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=REPLACE** option to replace the definition of a communication resource in the Online Communication Table Maintenance (OCTM) database. The definition for the communication resource must be provided in a storage area that is formatted by the **Communication Resource Definition DSECT** (**CT1TM**). Use the COMTC macro with the **ACTION=QUERY** option to obtain the current definition for the resource in a **CT1TM** DSECT. Apply the required changes for the communications resource to the appropriate fields within the **CT1TM** DSECT and issue the COMTC REPLACE macro.

Use the **CRN=** or **CRI=** parameter to identify the specific communication resource that is being changed. ALCS verifies that the communication resource being changed currently exists on the OCTM database, and if it does, alters its status to *changed*.

Use the **COMSDAT=** parameter to point to the storage area containing the modified **CT1TM** DSECT.

On return from COMTC REPLACE, an L3 size storage block resides at the ECB storage level specified by the **LEVEL=** parameter. During validation of each change to the communication resource, if errors are detected, appropriate error messages are placed in this storage block. Test the return code in register 15 to determine if any errors were found. The storage block will contain a maximum of 49 error messages. If more than 49 errors are found, ALCS will report only the first 49. Each error message will be a maximum of 80 bytes, therefore the storage block is formatted into 80-byte items. A 4-byte header resides at the beginning of the block containing a count of the error messages that have been placed in the block. Immediately following the header is the first 80-byte error message item. The error message text is in EBCDIC and the message contains a prefix **DXC***nnnn*, where the message number *nnnn* is in the range 9100 to 9139. Details of these error messages can be found in ALCS Messages and Codes.

When a batch of communication resources are being changed, allocate a ***communications group name*** to that batch. The **COMTC ALLOCATE** macro allows you to do this. Although a separate COMTC REPLACE is required for each communication resource being changed, when you use a communications group for a batch of resources, it enables you to use a single **COMTC LOAD** macro for loading all the changes on to the ALCS online communication table. Use the **GROUP=** parameter on COMTC REPLACE to identify the name of the communications group that the communication resource belongs to.

Register use

On return from COMTC REPLACE, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK	Processing completed without error
COMTC_R_RETRY	Unable to obtain lock for communication resource or group - retry the COMTC REPLACE macro immediately as this is only a temporary condition

COMTC_R_ERROR An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC REPLACE macro because of system error - check RO CRAS for a system error dump

COMTC_R_DATAERR The communications data provided in the **CT1TM** DSECT contained errors. A storage block at the ECB storage level specified by the **LEVEL=** parameter contains error messages describing up to 49 different errors in the communications data.

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN The communication resource name on the **CRN=** parameter is invalid

COMTC_S_NCRN The communication resource name on the **CRN=** parameter is unknown

COMTC_S_ACRN The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_NCRI The communication resource identifier on the **CRI=** parameter is unknown

COMTC_S_ACRI The address of the field that the **CRI=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP
Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_IPGS
A previous COMTC macro for this communication resource or group failed to complete. Use COMTC QUERY to determine the status of the communication resource or group.

COMTC_S_WGRP
The **GROUP=** parameter was incorrectly specified or contained the wrong group name. Other change requests have already been submitted for this communication resource, but those change requests do not belong to a communications group or belong to a different group.

COMTC_S_WCRX
A change request belonging to a communications group already exists for this communication resource, but the **GROUP=** parameter is missing from this COMTC macro

COMTC_S_ESEQ
There are change requests for this communication resource or group that are in the *confirmed* status

COMTC_S_TGRP
The maximum number of permitted non-group change requests have already been submitted. Identify non-group change requests that are waiting to be committed and use COMTC COMMIT to commit them.

COMTC_S_IDLR
The communication resource can not be changed because it is currently being deleted by a COMTC DELETE

COMTC_S_TCHG
The maximum number of permitted change requests have already been submitted for the communications group defined in the **GROUP=** parameter

COMTC_S_ADAT
The address of the field that the **COMSDAT=** parameter references is invalid

COMTC_S_ILDT
The communication resource type in the **CT1TM** DSECT for this communications resource does not match the communication resource type in the online communications table. An incorrect resource type may have been defined in the **CT1TM** DSECT for this resource. Alternatively, the wrong CRN or CRI may have been given in the COMTC macro parameters.

COMTC_S_ILDX
Modification of the PVC type for this X.25 PVC communication resource in the **CT1TM** DSECT is not allowed.

COMTC_S_ILDY
Modification of the interchange address, terminal address or owning PVC name in the **CT1TM** DSECT for this X.25 ALC terminal resource is not allowed.

COMTC_S_IUPD
ALCS can not change the X.25 ALC terminal resource because of a conflict with the ordinal numbers of the OCTM records that manage the X.25 ALC resource and its owning X.25 PVC. The X.25 ALC resource must be deleted and added again using the COMTC DELETE and COMTC ADD macros.

Alternatively, this reason code may indicate that modification of the associated device for this terminal (in the **CT1TM** DSECT) can not be performed because of a conflict with the ordinal number of the OCTM record that manages the associated device. This terminal resource must be deleted and added again using the COMTC DELETE and COMTC ADD macros.

When the return code **COMTC_R_DATAERR** is reported in general register 15, ALCS will have detected errors in the communication data provided in the **CT1TM** DSECT. To obtain a list of the errors that have been detected, examine the contents of the storage block that resides at the data level specified by the **LEVEL=** parameter. The storage block contains error messages describing the error conditions that have been detected. There can be a maximum of 49 error messages in the storage block. The previous section provides information on the format of this storage block.

COMTC REPLACE may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC REPLACE will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=REPLACE** to change a communication resource.

```
         COMTC ACTION=REPLACE,     ISSUE REPLACE REQUEST
               CRN=EBW008,         CRN IS IN EBW008
               COMSDAT=EBX000,     COMMS DATA IN EBX000
               LEVEL=D3            ECB LEVEL FOR ERROR MSG BLOCK
```

Related information

“CT1TM – Communication Resource Definition DSECT” on page 96.

COMTC DELETE – Delete a communication resource

Format

```
[label]  COMTC ACTION=DELETE,{CRN={field|(reg1)}|CRI={field|(reg2)}}
               [,GROUP={field|(reg3)}]
```

label
Any valid assembler label.

CRN={*field*|(*reg1*)}
The Communication Resource Name (CRN) of the communication resource that is being deleted. The CRN must be an alphanumeric string of 1 to 8 characters. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of an 8-byte field that contains the CRN.
(*reg1*) Register containing the address of an 8-byte field that contains the CRN. Use general registers 0 through 7.

CRI={*field*|(*reg2*)}
The Communication Resource Identifier (CRI) of the communication resource that is being deleted. The CRI must be 6 hexadecimal digits. Specify one of:

field Assembler label of a 3-byte field that contains the CRI.
(*reg2*) Register containing the address of a 3-byte field that contains the CRI. Use general registers 0 through 7.

GROUP={*field*|(*reg3*)}
The name of the communications group that this deletion request belongs to. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of a 7-byte field that contains the name of the communications group.
(*reg3*) Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=DELETE** option to delete a communication resource that is defined on the Online Communication Table Maintenance (OCTM) database.

Use the **CRN=** or **CRI=** parameter to identify the specific communication resource that is being deleted. ALCS verifies that the communication resource being deleted currently exists on the OCTM database and if it does, marks it for deletion. If a communications resource has a change request (**COMTC ADD** or **COMTC REPLACE**) currently in progress for it, the communication resource can not be deleted. Use the COMTC macro with the **ACTION=CANCEL** option to cancel the in-progress change request and then use COMTC DELETE to delete the communication resource.

When a batch of communication resources are being deleted, allocate a ***communications group name*** to that batch. The **COMTC ALLOCATE** macro allows you to do this. Although a separate COMTC DELETE is required for deleting each individual communication resource, when you use a communications group, it enables you to use a single **COMTC LOAD** macro to delete the batch of communication resources from the ALCS online communication table. Use the **GROUP=** parameter on COMTC DELETE to identify the name of the communications group that the deletion request belongs to.

Register use

On return from COMTC DELETE, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK
Processing completed without error

COMTC_R_RETRY
Unable to obtain lock for communication resource or group - retry the COMTC DELETE macro immediately as this is only a temporary condition

COMTC_R_ERROR
An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR
Unable to process the COMTC DELETE macro because of system error - check RO CRAS for a system error dump

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM
The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL
The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN
The communication resource name on the **CRN=** parameter is invalid

COMTC_S_NCRN
The communication resource name on the **CRN=** parameter is unknown

COMTC_S_ACRN
The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI
The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_NCRI
The communication resource identifier on the **CRI=** parameter is unknown

COMTC_S_ACRI The address of the field that the **CRI=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_IPGS A previous COMTC macro for this communication resource or group failed to complete. Use COMTC QUERY to determine the status of the communication resource or group.

COMTC_S_ESEQ There are change requests for this communication resource or group that are in the *confirmed* status

COMTC_S_TGRP The maximum number of permitted non-group change requests have already been submitted. Identify non-group change requests that are waiting to be committed and use COMTC COMMIT to commit them.

COMTC_S_TCHG The maximum number of permitted change requests have already been submitted for the communications group defined in the **GROUP=** parameter

COMTC_S_IDEL Another change request has already been submitted for this communication resource (a COMTC ADD, a COMTC REPLACE or a COMTC DELETE)

COMTC DELETE may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC DELETE will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=DELETE** to delete a communication resource.

```
         COMTC ACTION=DELETE,      ISSUE DELETE REQUEST
               CRN=EBW048          CRN IS IN EBW048
```

Related information

None.

COMTC CANCEL – Cancel a communication change request

Format

```
[label]  COMTC ACTION=CANCEL,{CRN={field|(reg1)}|CRI={field|(reg2)}}
               [,GROUP={field|(reg3)}]
```

label
Any valid assembler label.

CRN={*field***|(***reg1***)}**
The Communication Resource Name (CRN) of the communication resource whose change request on the Online Communication Table Maintenance (OCTM) database is being cancelled. The CRN must be an alphanumeric string of 1 to 8 characters. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of an 8-byte field that contains the CRN.
(*reg1*) Register containing the address of an 8-byte field that contains the CRN. Use general registers 0 through 7.

CRI={*field***|(***reg2***)}**
The Communication Resource Identifier (CRI) of the communication resource whose change request on the OCTM database is being cancelled. The CRI must be 6 hexadecimal digits. Specify one of:

field Assembler label of a 3-byte field that contains the CRI.
(*reg2*) Register containing the address of a 3-byte field that contains the CRI. Use general registers 0 through 7.

GROUP={*field***|(***reg3***)}**
The name of the communications group that the change request being cancelled belongs to. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of a 7-byte field that contains the name of the communications group.
(*reg3*) Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=CANCEL** option to cancel a communication change request that had been requested by **COMTC ADD**, **COMTC REPLACE** or by **COMTC DELETE**. If the communication change request belongs to a communications group, identify the name of that group in the **GROUP=** parameter.

ALCS verifies that the communication change request being cancelled currently exists on the OCTM database, that it belongs to the communications group that has been specified (in the **GROUP=** parameter) and that it does not have the *loaded*, *confirmed* or *committed* status. ALCS removes the change request from the OCTM database.

Register use

On return from COMTC CANCEL, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK Processing completed without error

COMTC_R_RETRY Unable to obtain lock for communication resource or group - retry the COMTC CANCEL macro immediately as this is only a temporary condition

COMTC_R_ERROR An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC CANCEL macro because of system error - check RO CRAS for a system error dump

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN The communication resource name on the **CRN=** parameter is invalid

COMTC_S_NCRN The communication resource name on the **CRN=** parameter is unknown

COMTC_S_ACRN The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_NCRI The communication resource identifier on the **CRI=** parameter is unknown

COMTC_S_ACRI The address of the field that the **CRI=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_IPGS A previous COMTC macro for this communication resource or group failed to complete. Use COMTC QUERY to determine the status of the communication resource or group.

COMTC_S_WGRP The **GROUP=** parameter was incorrectly specified or contained the wrong group name. The change request being cancelled does not belong to a communications group or belongs to a different group.

COMTC_S_WCRX The change request being cancelled belongs to a communications group, but the **GROUP=** parameter is missing from this COMTC macro

COMTC_S_ESEQ The change request for this communication resource is in the *loaded* or *confirmed* status

COMTC_S_NCRQ There are no change requests for this communication resource or the change request has already been cancelled

COMTC CANCEL may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC CANCEL will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=CANCEL** to cancel a change request that had been previously submitted via COMTC REPLACE for a communication resource.

```
         COMTC ACTION=CANCEL,      ISSUE CANCEL REQUEST
               CRN=EBW008          CRN IS IN EBW008
```

Related information

“COMTC ADD – Add a new communication resource” on page 57.
“COMTC REPLACE – Change a communication resource” on page 64.
“COMTC DELETE – Delete a communication resource” on page 69.

COMTC LOAD – Load communication change requests

Format

```
[label]  COMTC ACTION=LOAD,{CRN={field|(reg1)}|CRI={field|(reg2)}|GROUP={field|(reg3)}}
               ,LEVEL={Dn|(reg4)}
```

label
Any valid assembler label.

CRN={*field***|(***reg1***)}**
The Communication Resource Name (CRN) of the communication resource whose change request is being loaded. The CRN must be an alphanumeric string of 1 to 8 characters. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of an 8-byte field that contains the CRN.
(*reg1*) Register containing the address of an 8-byte field that contains the CRN. Use general registers 0 through 7.

CRI={*field***|(***reg2***)}**
The Communication Resource Identifier (CRI) of the communication resource whose change request is being loaded. The CRI must be 6 hexadecimal digits. Specify one of:

field Assembler label of a 3-byte field that contains the CRI.
(*reg2*) Register containing the address of a 3-byte field that contains the CRI. Use general registers 0 through 7.

GROUP={*field***|(***reg3***)}**
The name of the communications group whose change requests are being loaded. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of a 7-byte field that contains the name of the communications group.
(*reg3*) Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

LEVEL={D*n***|(***reg4***)}**
An available storage level in the Entry Control Block (ECB). An L3 size storage block is attached by ALCS to this storage level for recording errors that occur when the communication change requests are loaded in the online communication table. The storage block contains up to 49 error messages (with each message giving the name of the resource and a description of the load error). Specify one of:

D*n* Level symbol: **D0**, **D1**, and so on up to **DF** for level 15.
(*reg4*) Register containing the level value (use LA *reg*,**D***n*). Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=LOAD** option to load communication change requests that have been requested by **COMTC ADD**, **COMTC REPLACE** or by **COMTC DELETE** macros. Use the **CRN=** or **CRI=** parameter to load a change request for a specific communication resource. Use the **GROUP=** parameter to load a group of change requests for a communications group. When COMTC LOAD is used for a communications group, all the change requests that have been submitted via COMTC macros for individual communication resources that belong to the communications group are *loaded*.

The COMTC LOAD applies the change requests to the ALCS online communication table. For example, if you are loading a communications group that adds new 3270 terminals, when the COMTC LOAD has successfully completed, those 3270 terminals will be defined in the online communication table. The initial status of resources that are loaded in the communication table is *inactive* (except for OSYS *other system terminals* and resources defined as *test* terminals). If you require the status to be changed to *active*, use the COMCC macro to do this.

ALCS verifies that the communication change requests being loaded currently exist on the Online Communication Table Maintenance (OCTM) database. ALCS also verifies that they are not in the *confirmed* or *committed* status. ALCS changes the status of the change requests on the OCTM database to *loaded*.

On return from COMTC LOAD, an L3 size storage block resides at the ECB storage level specified by the **LEVEL=** parameter. When the communication change requests are loaded, if errors occur, appropriate error messages are placed in this storage block. Test the return code in register 15 to determine if any errors were found.
Communication load errors can also occur when ALCS processes the **ZACOM LOAD** command (this command is used for loading a communications load module created by ALCS communication generation). Communication load errors that occur during COMTC LOAD and ZACOM LOAD are reported on the MVS console. The load error messages include the prefix **DXC***nnn*, where the message number *nnn* is in the range 000 to 479. Details of these error messages can be found in ALCS Messages and Codes. COMTC LOAD therefore reports each load error on the MVS console and in the storage block. When more than 49 load errors occur during COMTC LOAD, the storage block contains only the first 49 load errors. Each error message will be a maximum of 80 bytes, therefore the storage block is formatted into 80-byte items. A 4-byte header resides at the beginning of the block giving the count of error messages in the block. Immediately following the header is the first error message item.

When you analyze the load errors that have been reported for a communications group, you may wish to cancel each change request that resulted in a load error. To do this, use **COMTC BACKOUT** to back out the communications group, use **COMTC CANCEL** to cancel each change request that caused a load error, and then use COMTC LOAD to reload the communications group.

Register use

On return from COMTC LOAD, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK	Processing completed without error

COMTC_R_RETRY Unable to obtain lock for communication resource or group - retry the COMTC LOAD macro immediately as this is only a temporary condition

COMTC_R_ERROR An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC LOAD macro because of system error - check RO CRAS for a system error dump

COMTC_R_LOADERR ALCS was unable to load one or more communication change requests in the online communications table. A storage block at the ECB storage level specified by the **LEVEL=** parameter contains error messages describing up to 49 different load errors. The storage block contains an error message count in the first fullword followed by 80-byte items that contain the individual error messages.

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN The communication resource name on the **CRN=** parameter is invalid

COMTC_S_ACRN The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_ACRI The address of the field that the **CRI=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP

Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_IPGS

A previous COMTC macro for this communication resource or group failed to complete. Use COMTC QUERY to determine the status of the communication resource or group.

COMTC_S_ESEQ

The change requests for this communication resource or group are in the *confirmed* or *committed* status

COMTC_S_NCHG

There are no change requests for this communication resource or communications group

COMTC_S_LOAD

ALCS experienced a serious error when loading the change requests into the online communications table. Some data in the online communications table may be incorrect and it may require an ALCS restart to clear this problem. When this error occurs, ALCS outputs a dump (dump number **000EEA**) for diagnostic purposes. If required, forward this dump to IBM for analysis.

COMTC_S_NOLD

ALCS has not loaded any change requests into the online communications table. This can occur when a COMTC LOAD has previously been issued for the communications resource or group and ALCS found no additional change requests that required loading. This can also occur when COMTC LOAD has been issued for a communications group, but ALCS was unable to load any of the change requests in the group because of multiple load errors. A storage block at the ECB storage level specified by the **LEVEL=** parameter contains error messages for up to 49 load errors.

COMTC LOAD may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC LOAD will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=LOAD** to load a change request that had been previously submitted via COMTC ADD for a communication resource.

```
         COMTC ACTION=LOAD,        ISSUE LOAD REQUEST
               CRN=EBW020,         CRN IS IN EBW020
               LEVEL=D5            ECB LEVEL FOR ERROR MSG BLOCK
```

Related information

"COMTC ADD – Add a new communication resource" on page 57.
"COMTC REPLACE – Change a communication resource" on page 64.
"COMTC DELETE – Delete a communication resource" on page 69.

COMTC BACKOUT – Back out communication change requests

Format

```
[label]  COMTC ACTION=BACKOUT,{CRN={field|(reg1)}|CRI={field|(reg2)}|GROUP={field|(reg3)}}
               ,LEVEL={Dn|(reg4)}
```

label
Any valid assembler label.

CRN={*field*|(*reg1*)}
The Communication Resource Name (CRN) of the communication resource whose change request is being backed out. The CRN must be an alphanumeric string of 1 to 8 characters. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of an 8-byte field that contains the CRN.
(*reg1*) Register containing the address of an 8-byte field that contains the CRN. Use general registers 0 through 7.

CRI={*field*|(*reg2*)}
The Communication Resource Identifier (CRI) of the communication resource whose change request is being backed out. The CRI must be 6 hexadecimal digits. Specify one of:

field Assembler label of a 3-byte field that contains the CRI.
(*reg2*) Register containing the address of a 3-byte field that contains the CRI. Use general registers 0 through 7.

GROUP={*field*|(*reg3*)}
The name of the communications group whose change requests are being backed out. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of a 7-byte field that contains the name of the communications group.
(*reg3*) Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

LEVEL={D*n*|(*reg4*)}
An available storage level in the Entry Control Block (ECB). An L3 size storage block is attached by ALCS to this storage level for recording errors that occur when the communication change requests are backed out in the online communication table. The storage block contains up to 49 error messages (with each message giving the name of the resource and a description of the error). Specify one of:

D*n* Level symbol: **D0**, **D1**, and so on up to **DF** for level 15.
(*reg4*) Register containing the level value (use LA *reg*,**D***n*). Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=BACKOUT** option to back out communication change requests that have been previously *loaded* in the ALCS online communication table (via **COMTC LOAD**). Communication change requests that have been *confirmed* (via **COMTC CONFIRM**) can also be backed out. The **COMTC ADD**, **COMTC REPLACE** and **COMTC DELETE** macros are used to submit change requests to ALCS. Use the **CRN=** or **CRI=** parameter to back out a change request for a specific communication resource. Use the **GROUP=** parameter to back out a group of change requests for a communications group. When COMTC BACKOUT is used for a communications group, all the change requests that have been submitted for individual communication resources that belong to the group are *backed out*.

The COMTC BACKOUT removes the change requests from the ALCS online communication table. For example, if you are backing out a communications group that added new 3270 terminals, after the *back out* those 3270 terminals will no longer have access to the ALCS system. If you are backing out a communications group that deleted a group of ALC terminals, after the *back out* those ALC terminals will have been reinstated in the ALCS online communication table.

ALCS verifies that the communication change requests being backed out currently exist on the Online Communication Table Maintenance (OCTM) database and that they are in the *loaded* or *confirmed* status.

On return from COMTC BACKOUT, an L3 size storage block resides at the ECB storage level specified by the **LEVEL=** parameter. When the communication change requests are backed out, if errors occur, appropriate error messages are placed in this storage block. Test the return code in register 15 to determine if any errors were found.
Similar error messages can be output when ALCS processes a COMTC LOAD. For example, when a COMTC DELETE is being backed out, the communications resource is reinstated (loaded) in the online communication table. Communication load errors can also occur when ALCS processes the **ZACOM LOAD** command (this command is used for loading a communications load module created by ALCS communication generation). Communication load errors that occur during COMTC BACKOUT and ZACOM LOAD are reported on the MVS console. The load error messages include the prefix **DXC***nnn*, where the message number *nnn* is in the range 000 to 479. Details of these error messages can be found in ALCS Messages and Codes. COMTC BACKOUT therefore reports each error on the MVS console and in the storage block. When more than 49 errors occur during COMTC BACKOUT, the storage block contains only the first 49. Each error message will be a maximum of 80 bytes, therefore the storage block is formatted into 80-byte items. A 4-byte header resides at the beginning of the block giving the count of error messages in the block. Immediately following the header is the first error message item.

Register use

On return from COMTC BACKOUT, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK	Processing completed without error
COMTC_R_RETRY	Unable to obtain lock for communication resource or group - retry the COMTC BACKOUT macro immediately as this is only a temporary condition

COMTC_R_ERROR An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC BACKOUT macro because of system error - check RO CRAS for a system error dump

COMTC_R_LOADERR ALCS was unable to backout one or more communication change requests from the online communications table. A storage block at the ECB storage level specified by the **LEVEL=** parameter contains error messages describing up to 49 different backout errors. The storage block contains an error message count in the first fullword followed by 80-byte items that contain the individual error messages.

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN The communication resource name on the **CRN=** parameter is invalid

COMTC_S_ACRN The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_ACRI The address of the field that the **CRI=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_IPGS A previous COMTC macro for this communication resource or group failed to complete. Use COMTC QUERY to determine the status of the communication resource or group.

COMTC_S_ESEQ The change requests for this communication resource or group must be in the *loaded* or *confirmed* status

COMTC_S_NCHG There are no change requests for this communication resource or communications group

COMTC_S_LOAD ALCS experienced a serious error when performing a backout of change requests from the ALCS online communications table. Some data in the online communications table may be incorrect and it may require an ALCS restart to clear this problem. When this error occurs, ALCS outputs a dump (dump number **000EEA**) for diagnostic purposes. If required, forward this dump to IBM for analysis.

COMTC_S_NOLD Multiple change requests in a communications group were being backed out from the online communications table, but ALCS was unable to backout any of them. A storage block at the ECB storage level specified by the **LEVEL=** parameter contains error messages for up to 49 backout errors.

COMTC_S_ABCK The communications group or resource has already been backed out.

COMTC BACKOUT may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC BACKOUT will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=BACKOUT** to back out change requests that had been previously loaded for a communications group.

```
         COMTC ACTION=BACKOUT,     ISSUE BACKOUT REQUEST
               GROUP=EBX040,       NAME OF COMMS GROUP IN EBX040
               LEVEL=D3            ECB LEVEL FOR ERROR MSG BLOCK
```

Related information

"COMTC LOAD – Load communication change requests" on page 76.
"COMTC CONFIRM – Confirm communication change requests" on page 85.

COMTC CONFIRM – Confirm communication change requests

Format

```
[label]  COMTC ACTION=CONFIRM,{CRN={field|(reg1)}|CRI={field|(reg2)}|GROUP={field|(reg3)}}
```

label
Any valid assembler label.

CRN={*field*|(*reg1*)}
The Communication Resource Name (CRN) of the communication resource whose change request is being confirmed. The CRN must be an alphanumeric string of 1 to 8 characters. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of an 8-byte field that contains the CRN.
(*reg1*) Register containing the address of an 8-byte field that contains the CRN. Use general registers 0 through 7.

CRI={*field*|(*reg2*)}
The Communication Resource Identifier (CRI) of the communication resource whose change request is being confirmed. The CRI must be 6 hexadecimal digits. Specify one of:

field Assembler label of a 3-byte field that contains the CRI.
(*reg2*) Register containing the address of a 3-byte field that contains the CRI. Use general registers 0 through 7.

GROUP={*field*|(*reg3*)}
The name of the communications group whose change requests are being confirmed. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of a 7-byte field that contains the name of the communications group.
(*reg3*) Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=CONFIRM** option to confirm communication change requests that have been previously *loaded* in the ALCS online communication table (via **COMTC LOAD**). The **COMTC ADD**, **COMTC REPLACE** and **COMTC DELETE** macros are used to submit communication change requests to ALCS. The COMTC CONFIRM enables those change requests to be reloaded in the ALCS online communication table during the next ALCS restart.

Use the **CRN=** or **CRI=** parameter to confirm a change request for a specific communication resource. Use the **GROUP=** parameter to confirm a group of change requests for a communications group. When COMTC CONFIRM is used for a communications group, all the change requests that have been submitted for individual communication resources that belong to the communications group are *confirmed*.

ALCS verifies that the communication change requests being confirmed currently exist on the Online Communication Table Maintenance (OCTM) database and that they are in the *loaded* status (but not *committed*). The COMTC CONFIRM changes the status of the communication change requests on the OCTM database to *confirmed*. After a communication resource or a communications group has been *confirmed*, no further change requests can be applied to it.

Register use

On return from COMTC CONFIRM, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK Processing completed without error

COMTC_R_RETRY Unable to obtain lock for communication resource or group - retry the COMTC CONFIRM macro immediately as this is only a temporary condition

COMTC_R_ERROR An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC CONFIRM macro because of system error - check RO CRAS for a system error dump

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN The communication resource name on the **CRN=** parameter is invalid

COMTC_S_ACRN The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_ACRI The address of the field that the **CRI=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_IPGS A previous COMTC macro for this communication resource or group failed to complete. Use COMTC QUERY to determine the status of the communication resource or group.

COMTC_S_ESEQ The change requests for this communication resource or group must be in the *loaded* status

COMTC_S_NCHG There are no change requests for this communication resource or communications group

COMTC CONFIRM may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC CONFIRM will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=CONFIRM** to confirm a change request that had been previously submitted via COMTC ADD for a communication resource.

```
         COMTC ACTION=CONFIRM,     ISSUE CONFIRM REQUEST
               CRN=EBW020          CRN IS IN EBW020
```

Related information

“COMTC LOAD – Load communication change requests” on page 76.

COMTC COMMIT – Commit communication change requests

Format

```
[label]  COMTC ACTION=COMMIT,{CRN={field|(reg1)}|CRI={field|(reg2)}|GROUP={field|(reg3)}}
```

label
Any valid assembler label.

CRN={*field*|(*reg1*)}
The Communication Resource Name (CRN) of the communication resource whose change request is being committed. The CRN must be an alphanumeric string of 1 to 8 characters. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of an 8-byte field that contains the CRN.

(*reg1*) Register containing the address of an 8-byte field that contains the CRN. Use general registers 0 through 7.

CRI={*field*|(*reg2*)}
The Communication Resource Identifier (CRI) of the communication resource whose change request is being committed. The CRI must be 6 hexadecimal digits. Specify one of:

field Assembler label of a 3-byte field that contains the CRI.

(*reg2*) Register containing the address of a 3-byte field that contains the CRI. Use general registers 0 through 7.

GROUP={*field*|(*reg3*)}
The name of the communications group whose change requests are being committed. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of a 7-byte field that contains the name of the communications group.

(*reg3*) Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

Description

Use the COMTC macro with the **ACTION=COMMIT** option to commit communication change requests on the Online Communication Table Maintenance (OCTM) database that have been previously *confirmed* (via **COMTC CONFIRM**). The **COMTC ADD**, **COMTC REPLACE** and **COMTC DELETE** macros are used to submit communication change requests to ALCS. The COMTC COMMIT makes those change requests permanent on the OCTM database.

Use the **CRN=** or **CRI=** parameter to commit a change request for a specific communication resource. Use the **GROUP=** parameter to commit a group of change requests for a communications group. When COMTC COMMIT is used for a communications group, all the change requests that have been submitted for individual communication resources that belong to the communications group are *committed*.

ALCS verifies that the communication change requests being committed currently exist on the OCTM database and that they are in the *confirmed* status. After change requests have been *committed* for one or more communication resources, new change requests can be submitted for those communication resources.

Register use

On return from COMTC COMMIT, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK Processing completed without error

COMTC_R_RETRY Unable to obtain lock for communication resource or group - retry the COMTC COMMIT macro immediately as this is only a temporary condition

COMTC_R_ERROR An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.

COMTC_R_SYSERR Unable to process the COMTC COMMIT macro because of system error - check RO CRAS for a system error dump

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again

COMTC_S_USRL The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed

COMTC_S_FCRN The communication resource name on the **CRN=** parameter is invalid

COMTC_S_ACRN The address of the field that the **CRN=** parameter references is invalid

COMTC_S_FCRI The communication resource identifier on the **CRI=** parameter is invalid

COMTC_S_ACRI The address of the field that the **CRI=** parameter references is invalid

COMTC_S_FGRP The communications group name on the **GROUP=** parameter is invalid

COMTC_S_NGRP The communications group name on the **GROUP=** parameter is unknown

COMTC_S_AGRP The address of the field that the **GROUP=** parameter references is invalid

COMTC_S_CGRP Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.

COMTC_S_IPGS A previous COMTC macro for this communication resource or group failed to complete. Use COMTC QUERY to determine the status of the communication resource or group.

COMTC_S_ESEQ The change requests for this communication resource or group must be in the *confirmed* status

COMTC_S_NCHG There are no change requests for this communication resource or communications group

COMTC COMMIT may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC COMMIT will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=COMMIT** to commit a change request that had been previously submitted via COMTC ADD for a communication resource.

```
         COMTC ACTION=COMMIT,      ISSUE COMMIT REQUEST
               CRN=EBW020          CRN IS IN EBW020
```

Related information

"COMTC CONFIRM – Confirm communication change requests" on page 85.

COMTC UNALLOCATE – Unallocate a communications group

Format

```
[label]  COMTC ACTION=UNALLOCATE,GROUP={field|(reg)}
               [,DELETE={NO|YES}]
```

label
Any valid assembler label.

GROUP={*field*|(*reg*)}
The name of the communications group that is being unallocated on the Online Communication Table Maintenance (OCTM) database. It must be an alphanumeric string of 4 to 7 characters, although the first character can not be numeric. If the group name is less than 7 characters, it should be left justified and padded with spaces. Specify one of:

field Assembler label of a 7-byte field that contains the name of the communications group.

(*reg*) Register containing the address of a 7-byte field that contains the name of the communications group. Use general registers 0 through 7.

DELETE={<u>NO</u>|YES}
Specifies how the unallocate request should be processed if the communications group is not *committed* and there are communication change requests that belong to it.

<u>NO</u> Do not unallocate the communications group if there are communication change requests belonging to it and the communications group has not been *committed* (via **COMTC COMMIT**).

YES Unallocate the communications group and delete (cancel) all the communications change requests that belong to it.

Description

Use the COMTC macro with the **ACTION=UNALLOCATE** option to unallocate (remove) a communications group on the OCTM database. When all the communications change requests belonging to a group have been *committed* (via **COMTC COMMIT**) the group name should be unallocated. After the communications group has been unallocated, the group name becomes available again for reuse. If the same group name is required again for applying new communications change requests, the group name must be unallocated and then allocated again (via **COMTC ALLOCATE**).

If the **DELETE=NO** option is used and the communications group has not yet been *committed*, ALCS will not allow the communications group to be unallocated.

If the communications group has communication change requests belonging to it that have not yet been *loaded* (via **COMTC LOAD**) or have been *backed out* (via **COMTC BACKOUT**), use the **DELETE=YES** option to unallocate this communications group and delete (cancel) from the OCTM database all the change requests that belong to it.

Register use

On return from COMTC UNALLOCATE, a return code will reside in the low-order two bytes (bits 16 to 31) of general register 15. Test the return code that COMTC has set by using the following symbols:

COMTC_R_OK	Processing completed without error
COMTC_R_RETRY	Unable to obtain lock for communication resource or group - retry the COMTC UNALLOCATE macro immediately as this is only a temporary condition
COMTC_R_ERROR	An error occurred during the processing of the COMTC macro. There are various types of error that might occur including errors in the COMTC macro parameters. Check the reason code in the high-order two bytes of register 15 (as described below) for details of the error.
COMTC_R_SYSERR	Unable to process the COMTC UNALLOCATE macro because of system error - check RO CRAS for a system error dump

When general register 15 contains the return code **COMTC_R_ERROR**, a reason code will reside in the high-order two bytes (bits 0 to 15) of the register. Test the reason code that COMTC has set by using the following symbols:

COMTC_S_ACTM	The OCTM database is currently unavailable because the OCTM facility has been stopped by the ZOCTM STOP command - retry the COMTC macro when OCTM is active again
COMTC_S_USRL	The OCTM database is currently unavailable because the length of the communications user data area has been changed in a communications generation load module (**COMGEN USERLEN=** parameter) - retry the COMTC macro when an OCTM backup and restore has been performed
COMTC_S_FGRP	The communications group name on the **GROUP=** parameter is invalid
COMTC_S_NGRP	The communications group name on the **GROUP=** parameter is unknown
COMTC_S_AGRP	The address of the field that the **GROUP=** parameter references is invalid
COMTC_S_CGRP	Another COMTC macro may currently be attempting to unallocate the communications group defined on the **GROUP=** parameter. Use COMTC QUERY to determine the status of the communications group.
COMTC_S_IPGS	A previous COMTC macro for this communications group failed to complete. Use COMTC QUERY to determine the status of the communications group.
COMTC_S_NUNL	The communications group can not be unallocated. The **DELETE=NO** option was used but there are communication change requests which belong to the communications group that have not yet been *committed*.

COMTC_S_YUNL The communications group can not be unallocated. The **DELETE=YES** option was used but there are communication change requests which belong to the communications group that are in the *loaded* or *confirmed* status. Use COMTC BACKOUT to back out the change requests from the online communication table and issue the COMTC UNALLOCATE macro again.

COMTC UNALLOCATE may also use general register 14. It does not corrupt any other registers.

Loss of control

COMTC UNALLOCATE will cause the entry to lose control.

Example

The following example shows how you could use the COMTC macro with **ACTION=UNALLOCATE** to unallocate a communications group. In this example, the communications group included some invalid change requests, therefore COMTC UNALLOCATE is used to not only unallocate the communications group but to delete every communications change request that belongs to the group.

```
UNALLOC  EQU   *
         COMTC ACTION=UNALLOCATE,  ISSUE UNALLOCATE REQUEST
               GROUP=(R06),        GROUP NAME POINTED TO BY REGISTER 6
               DELETE=YES          REQUEST DELETION OF CHANGE REQUESTS
         LR    R04,R14             LOAD COUNT OF COMMS RESOURCES
         LR    R14,R15             LOAD RETURN CODE / REASON CODE
         SLL   R14,16              ISOLATE RETURN CODE
         SRL   R14,16              SHIFT TO LOW-ORDER BYTES
         B     *+4(R14)            BRANCH ON RETURN CODE
         B     URC0                0 - OK
         B     URC4                4 - RETRY IMMEDIATELY
         B     URC8                8 - USER ERROR
         B     URC12               12 - SYSTEM ERROR
URC4     EQU   *
         DEFRC ,                   WAIT A MOMENT
         B     UNALLOC             RETRY THE COMTC REQUEST
URC0     EQU   *
```

Related information

"COMTC ALLOCATE – Allocate a new communications group" on page 54.

Appendix C - DSECT's used by COMTC macro

The OCTM interface provided for the CEUS not only includes the COMTC macro but also DSECT macros. This section provides a detailed description of the DSECT's that are used via the COMTC macro.

CT1TM – Communication Resource Definition DSECT

Format

```
CT1TM REG=reg[,SUFFIX=s]
```

REG=*reg*
Base register for DSECT addressability.

SUFFIX=*s*
Suffix for this DSECT, a single alphanumeric character.

Description

Use the CT1TM DSECT to reference fields in the Communication Resource Definition data block. This DSECT is used by the ALCS Online Communication Table Maintenance (OCTM) facility, and it contains information about a communication resource. There are eight types of communication resource that this DSECT can hold information for.

The OCTM **Communication Table Update** monitor-request macro COMTC uses this DSECT for three of its options.
The first option is **COMTC QUERY** which provides detailed information about a communication resource in a data block that is formatted by this DSECT. **COMTC QUERY** does not reference this DSECT directly, but via the **Communication Resource and Group Information DSECT (CT2TM)**.
The other two options are **COMTC ADD** and **COMTC REPLACE**. These two options require detailed information to be provided about a communication resource in a storage area that is formatted by this DSECT. When this DSECT is used for the **COMTC ADD** macro, the storage area should be initialized with hexadecimal zeros. When this DSECT is used for the **COMTC REPLACE** macro, the storage area should be initialized by obtaining the current definition of the resource via a **COMTC QUERY**.

This DSECT supports eight different communication resource types. The field CT1TYP in this DSECT contains the name of the communication resource type.

When a Communication Resource Definition block is being prepared for the COMTC ADD macro, many of the fields are optional but some are mandatory. Guidance is given below on the fields that are mandatory for the COMTC ADD macro.

The CT1TM DSECT defines the following symbols for fields within the Communication Resource Definition data block. The symbols are described below under various headings with each heading identifying the communication resource types that the symbols relate to.

Symbols for all resource types: The following symbols can be used for all types of communication resource.

CT1TYP	Communication resource type Eight byte field that identifies the communication resource type that this communication resource belongs to. For example, if this field contains the character string VTAM3270, the resource type is a 3270 terminal.

In the communications generation, this is provided in the COMDEF **LDTYPE=** parameter.

Resource Type	**Resource Type Description**
MQTERM	The MQSeries terminal resource type, provided for those terminals that connect to ALCS through an MQSeries queue.
WASTERM	The Websphere Application Server terminal resource type, provided for terminals that connect to ALCS through the ALCS WAS bridge.
NETVIEW	The NetView operator ID resource type, provided for NetView operator terminals.
OSYS	The Other System terminal resource type, provided for those terminals that access this ALCS system through another system (who owns and controls the terminal).
TCPIPALC	The TCP/IP ALC terminal resource type, provided for those terminals that connect to ALCS through a TCP/IP network.
VTAM3270	The VTAM 3270 terminal resource type, provided for 3270 terminals.
VTAMALC	The VTAM/ALCI ALC terminal resource type, provided for those ALC terminals that connect to ALCS through ALCI and VTAM.
X25PVC	The X.25 permanent virtual circuit (PVC) resource type, provided for the connectivity of ALC terminals through NPSI and VTAM.
X25ALC	The X.25 ALC terminal resource type, provided for those ALC terminals that connect to ALCS through NPSI, VTAM and an X.25 PVC.

COMTC ADD considerations:
This field is mandatory. The name of the communication resource type must be left justified and padded with spaces.

COMTC REPLACE considerations:
Although the communication resource type can not be changed, this field must though contain the name of the communication resource type that this communication resource belongs to.

CT1USD Communication resource user data
A 256 byte field that contains communications user data that is associated with this communication resource. If your ALCS installation provides user data for the communication resources, you should have a data macro that defines each field in the user data area. Define the user data that requires refreshing in the online communication table on an ALCS restart. In the communications generation, this is provided in the COMDEF **USERDAT=** or **USERnn=** parameters.

COMTC ADD considerations:
The user data that is to be placed in this field is normally less than 256 bytes. Therefore, before inserting the user data in this field, initialize it

by inserting a *pad character* in each byte. The CT1UPD symbol described below defines the hexadecimal value for the *pad character* (which can be hexadecimal zero). Insert the user data that you require in this field, overlaying the *pad character*. ALCS does not validate the user data, therefore any errors in this data will not be detected by ALCS. If user data is not being provided, each byte should either contain hexadecimal zero or the *pad character*.

COMTC REPLACE considerations:
If the communications user data is not being changed, do not alter the user data supplied by COMTC QUERY. If communications user data is being changed, initialize this field by placing a *pad character* in each byte. The CT1UPD symbol described below defines the hexadecimal value for the *pad character* (which can be hexadecimal zero). Insert the user data that requires changing in this field, overlaying the *pad character*.

CT1UPD

User data pad character
One byte field that contains the communications user data *pad character*.

COMTC ADD considerations:
If communications user data is not being provided for this communication resource (field CT1USD is hexadecimal zeros) this field should contain hexadecimal zero. If communications user data is being provided, define a *pad character* in this field. The hexadecimal *pad character* must reside in each byte of CT1USD that does not contain communications user data. The *pad character* can be any hexadecimal value.

COMTC REPLACE considerations:
If the communications user data is not being changed, leave the contents of this field as hexadecimal zero (this is the value that COMTC QUERY places in this field).
If communications user data is being changed, provide a *pad character* in this field. Initialize the 256-byte user data field CT1USD by placing this *pad character* in each byte, and then insert the user data that requires changing in CT1USD (overlaying the *pad character*).

CT1IST

Initial status
Status byte that defines the initial status of the communication resource when the ALCS system starts. In the communications generation, this is provided in the COMDEF **ISTATUS=** parameter. Initial status must not be defined for the resource type **NETVIEW**.

Symbol	**Meaning if on (1)**
CT1ACTV	The initial status is *active*. ALCS will initiate communication with this resource when the system starts.
CT1INAC	The initial status is *inactive*. ALCS will start this communication resource only in response to an external request (for example, a ZACOM command).
CT1SHAR	The initial status is *shared*. This symbol is valid only for VTAM 3270 printers. ALCS will share this 3270 printer with one or more other OS/390 subsystems.

COMTC ADD considerations:

If you do not specify the initial status for the communication resource, ALCS will allocate a status of *active*.

COMTC REPLACE considerations:
If the initial status is not being changed, do not modify the value provided by COMTC QUERY. If the initial status is being changed, set the contents of the status byte using the appropriate symbol.

CT1FLA

Status flag byte
Status byte that defines the *logon* and *test* status for the communication resource.

Symbol	**Meaning if on (1)**
CT1LOGO	The user of this communication resource must logon to the ALCS system with *userid* and *password*. The *logon* option is used for System Authorization Facility (SAF) authorization checking. In the communications generation, this is provided in the COMDEF **LOGON=** parameter. Logon is not allowed for the resource types **MQTERM, NETVIEW, OSYS,** and **X25PVC**. If the user of this communication resource is not required (or allowed) to logon, the resource requires a default user ID (provided in field CT1UID).
CT1TEST	The communication resource is a *test* resource. ALCS *test* resources receive their input messages from the ALCS System Test Vehicle (STV). In the communications generation, this is provided in the COMDEF **TEST=** parameter. The following resource types can be defined as *test* resources, **VTAM3270, VTAMALC, X25ALC,** and **X25PVC**.

COMTC ADD considerations:
If logon is required for this resource, or if it is a test resource, set the contents of this status flag byte using the appropriate symbol. Note that a communication resource can not have both of these options set on (it can be neither of them, or just one of them, but not both).

COMTC REPLACE considerations:
If the logon or test status for this resource is not being changed, do not modify the value provided by COMTC QUERY. Alternatively, set the contents of this status flag byte using the appropriate symbol. Note that only one of these options can be selected, not both.

CT1UID

Default user ID
Eight byte field that contains the default user ID for this communication resource. In the communications generation, this is provided in the COMDEF **LOGON=** parameter. A default user ID must not be defined for the resource type **NETVIEW**. When this field is not used, it should contain spaces or hexadecimal zeros. The default user ID is used by the OS/390 System Authorization Facility (SAF) for authorization checking and is associated with the logon option for this communications resource. The CT1LOGO symbol described above provides the logon option. For some communication resource types, logon is not allowed, but for others it is an optional feature.

COMTC ADD considerations:

The default user ID must be left justified and padded with spaces. If a default user ID is not being provided, set the contents of this field to spaces or hexadecimal zero.
If you specify that users of this communications resource must always logon to the ALCS system with *user ID* and *password* (the CT1LOGO symbol is set on), you do not need to provide a default user ID.
If you specify that logon is not required, you must provide a default user ID (ALCS does not use the default user ID provided in the **DEFAULTID=** parameter on the communication generation **COMGEN** macro).

COMTC REPLACE considerations:
If the default user ID for this resource is not being changed, do not modify the value provided by COMTC QUERY.
If you are deleting the default user ID, place spaces or hexadecimal zero in this field.
If you are providing a new or changed default user ID, but you have also specified that users of this communications resource must always logon to the ALCS system with *user ID* and *password* (the CT1LOGO symbol is set on) ALCS will not use this default user ID. If the CT1LOGO symbol is set off (logon is not required), ensure that this field contains a default user ID.

Symbols for the terminal resource types: The following symbols can be used for the terminal resource types. If the communication resource is an X.25 permanent virtual circuit (PVC), the symbols in the following section can be ignored.

CT1APL Application name
Four byte field that contains the name of the ALCS application to which ALCS routes messages from this communication resource. In the communications generation, this is provided in the COMDEF **APPL=** parameter. The application resource type definitions in the communication generation (**LDTYPE=ALCSAPPL**) define the valid application names for your ALCS system.

COMTC ADD considerations:
An application name ***must*** be provided for the resource. ALCS does not provide a default application name. The application name must be an alphanumeric string, and if it is less than 4 characters, it should be left justified and padded with spaces.

COMTC REPLACE considerations:
If the application name for the resource is not being changed, do not modify the value provided by COMTC QUERY. If the application name is being changed and it is less than 4 characters, it should be left justified and padded with spaces.

CT1ASD Associated resource CRN
Eight byte field that contains the CRN of the associated communication resource. In the communications generation, this is provided in the COMDEF **ASSDEV=** parameter. When this field is not used, it should contain spaces or hexadecimal zeros.

COMTC ADD considerations:
Ensure that the communications definition for the associated resource has already been defined in your ALCS system. If the CRN of the associated resource is less than 8 characters, it should be left justified and

padded with spaces. If this communication resource is a CRAS display, you are recommended to associate a CRAS printer with that display because some CRAS commands output response messages to an associated printer.

COMTC REPLACE considerations:
If the associated communication resource is not being changed, do not modify the value provided by COMTC QUERY. If the current associated resource is being removed (the terminal will no longer have an associated resource), set the contents of this field to hexadecimal zero. If the CRN of the associated resource is being changed, ensure that the communications definition for the associated resource has already been defined in your ALCS system. If the CRN of the associated resource is less than 8 characters, it should be left justified and padded with spaces.

CT1CRS

Alternate CRAS type
Flag byte that indicates if the communication resource is an alternate CRAS terminal. In the communications generation, this is provided in the COMDEF **CRAS=** parameter. All terminal resource types (displays and printers) can be an alternate CRAS type of **AT**, but only printer resource types can be an alternate CRAS type of **AP**. The alternate CRAS number is defined in field CT1ALT (described below).

Note: The terminals which have Prime CRAS status or RO CRAS status can not be defined in the OCTM database (these terminals must be defined in the ALCS communications generation).

Symbol	**Meaning if on (1)**
CT1CRSAT	This communication resource is an alternate (AT*nnn*) CRAS
CT1CRSAP	This communication resource is an alternate (AP*nnn*) printer CRAS

COMTC ADD considerations:
If the terminal is not being given alternate CRAS status, this flag byte must contain hexadecimal zero. If the terminal requires alternate CRAS status, set the contents of this flag byte to indicate the alternate CRAS type (AT or AP). Ensure that you also set the alternate CRAS number in field CT1ALT (described below). Terminal resource types **VTAM3270**, **NETVIEW**, **MQTERM (TERM=3270DSP and TERM=3270PRT)**, and **WASTERM (TERM=3270DSP and TERM=3270PRT)** can have any alternate CRAS number, but all other terminal resource types are restricted to the alternate CRAS number of AT17 to AT255 and AP17 to AP255.

COMTC REPLACE considerations:
If the CRAS status for this communication resource is not being changed, do not modify the value provided by COMTC QUERY. If you are removing alternate CRAS status from this terminal, set the contents of this flag byte to hexadecimal zero. If you are changing the CRAS status (for example, from an **AT** CRAS to an **AP** CRAS) set the contents of this flag byte using the appropriate symbol.

CT1ALT

Alternate CRAS number
One byte field that contains the alternate CRAS number *nnn* (AT*nnn* or AP*nnn*). In the communications generation, this is provided in the

COMDEF **CRAS=** parameter. If the terminal has an alternate CRAS number, the flag byte CT1CRS, as described above, indicates the alternate CRAS type (**AT** or **AP**). If the terminal is not an alternate CRAS status, this field will contain hexadecimal zero. For terminal resource types **VTAM3270**, **NETVIEW**, **MQTERM (TERM=3270DSP and TERM=3270PRT)**, and **WASTERM (TERM=3270DSP and TERM=3270PRT)** the alternate CRAS number is between 1 and 255. For all other terminals, the alternate CRAS number is between 17 and 255.

COMTC ADD considerations:
For **AT** CRAS terminals, the alternate CRAS number must not be allocated to any other **AT** CRAS terminal. Similarly, for **AP** CRAS printers, the alternate CRAS number must not be allocated to any other **AP** CRAS printer. Terminal resource types **VTAM3270**, **NETVIEW**, **MQTERM (TERM=3270DSP and TERM=3270PRT)**, and **WASTERM (TERM=3270DSP and TERM=3270PRT)** can have any alternate CRAS number, but all other terminal resource types are restricted to the alternate CRAS number of 17 to 255. If the terminal is not an alternate CRAS, set the contents of this field to hexadecimal zero.

COMTC REPLACE considerations:
If the alternate CRAS number for this terminal resource is not being changed, do not modify the contents of this field (as provided by COMTC QUERY). If you are removing the alternate CRAS status from this terminal (field CT1CRS has been set to hexadecimal zero), set the contents of this field to hexadecimal zero.
If you are changing the alternate CRAS number for this terminal, note that terminal resource types **VTAM3270**, **NETVIEW**, **MQTERM (TERM=3270DSP and TERM=3270PRT)**, and **WASTERM (TERM=3270DSP and TERM=3270PRT)** can have any alternate CRAS number, but all other terminal resource types are restricted to the alternate CRAS number of 17 to 255. For **AT** CRAS terminals, the alternate CRAS number must not be allocated to any other **AT** CRAS terminal. Similarly, for **AP** CRAS printers, the alternate CRAS number must not be allocated to any other **AP** CRAS printer.

CT1TRM Terminal device type
Flag byte that defines the device type of the terminal resource. In the communications generation, this is provided in the COMDEF **TERM=** parameter.

Symbol	**Meaning if on (1)**
CT11977	This communication resource emulates an ALC 1977 terminal (ALC keyboard printer)
CT11980	This communication resource emulates an ALC 1980 terminal (ALC receive-only printer)
CT12915	This communication resource emulates an ALC 2915 terminal (ALC 12 line display)
CT14505	This communication resource emulates an ALC 4505 terminal (ALC 30 line display)
CT1327P	This communication resource is a 3270 printer
CT1327D	This communication resource is a 3270 display

COMTC ADD considerations:
For most terminal resources, the terminal device type ***must*** be provided. The only exception is for VTAM 3270 terminals which are not system test vehicle (STV) *test* resources (for these terminals, ALCS accepts the device type you have specified until a VTAM session is established with the terminal). Any of the ALC terminal device types can be used for the **MQTERM, WASTERM, OSYS, TCPIPALC, VTAMALC** and **X25ALC** resource types. Either of the 3270 terminal device types can be used for the **MQTERM, WASTERM, NETVIEW, OSYS** and **VTAM3270** resource types.

COMTC REPLACE considerations:
If the terminal device type for this communication resource is not being changed, do not modify the value provided by COMTC QUERY. If you are changing the terminal device type, set the contents of this flag byte using the appropriate symbol. Any of the ALC terminal device types can be used for the **MQTERM, WASTERM, OSYS, TCPIPALC, VTAMALC** and **X25ALC** resource types. Either of the 3270 terminal device types can be used for the **MQTERM, WASTERM, NETVIEW, OSYS** and **VTAM3270** resource types.

CT1CSI — Cross system identifier
Three byte field that contains the cross system identifier for this terminal resource. In the communications generation, this is provided in the COMDEF **CSID=** parameter. For terminal resources that do not have a cross system identifier, this field contains hexadecimal zero. The cross system identifier, which is 6 hexadecimal digits, is required by terminal resources that are owned by another system. Resource types **MQTERM**, **WASTERM** and **NETVIEW** do not have a cross system identifier.

COMTC ADD considerations:
When a cross system identifier is not being provided for the communication resource, this field must contain hexadecimal zero. A cross system identifier ***must*** though be provided when the communications resource type is an Other System terminal (**OSYS**). When the communications resource type is **MQTERM**, **WASTERM** or **NETVIEW** a cross system identifier must not be provided.

COMTC REPLACE considerations:
If the cross system identifier for this communication resource is not being changed, do not modify the contents of this field (as provided by COMTC QUERY). If you are changing it, insert the new cross system identifier in this field. Do not though provide a cross system identifier when the communications resource type is **MQTERM**, **WASTERM** or **NETVIEW**. If you are deleting the cross system identifier, set the contents of this field to hexadecimal zero. If the communications resource type is an Other System terminal (**OSYS**), you must not delete the cross system identifier.

CT1FLB

Terminal characteristics
Flag byte that defines specific characteristics of the terminal resource.

Symbol	**Meaning if on (1)**
CT1FALL	CRAS fallback candidate The terminal resource is a fallback candidate for Prime CRAS or RO CRAS. In the communications generation, this is provided by the **FALLBACK** option on the COMDEF **CRAS=** parameter. The only communication resource types that can be CRAS fallback candidates are **NETVIEW** and **VTAM3270**. Terminal resources must be an alternate CRAS AT1 to AT16 or AP1 to AP16 to be eligible as a fallback candidate. COMTC ADD and REPLACE considerations: If this terminal resource is to be a CRAS fallback candidate, ensure that it is defined as an alternate CRAS in field CT1CRS. Also ensure that it has an alternate CRAS number between 1 and 16 defined in field CT1ALT and that it is a resource type of **NETVIEW** or **VTAM3270**. On most ALCS systems there is at least one alternate CRAS display (AT1 to AT16) defined as a fallback candidate for Prime CRAS, and at least one alternate CRAS printer (AT1 to AT16 or AP1 to AP16) defined as a fallback candidate for RO CRAS.
CT1SYSS	Suppress system generated messages The communication resource must not be sent system generated messages. In the communications generation, this is provided in the COMDEF **SYSSEND=** parameter. The ALCS system can generate *system* messages that it wants to send to the communication resource. This can include system error and unsolicited messages. COMTC ADD and COMTC REPLACE considerations: If the communication resource is not to receive system generated messages, ensure that this indicator is set on.
CT1TABC	Tabular skip feature The communication resource has the tabular skip feature. This feature is available only on the ALC 1977 printer terminal. In the communications generation, this is provided in the COMDEF **TAB=** parameter. The **VTAM3270** and **NETVIEW** resource types do not support ALC printers, therefore this feature is not relevant for those communication resource types. COMTC ADD and COMTC REPLACE considerations: The terminal resource must be an ALC 1977 printer if the tabular skip feature is to be used. Ensure that it is defined as an ALC 1977 printer in the field CT1TRM. Also ensure that it is not a **NETVIEW** or **VTAM3270** resource type.
CT1UNSL	Unsolicited light feature The communication resource has the unsolicited light feature. This feature is available only on the ALC 1977

printer terminal. In the communications generation, this is provided in the COMDEF **UNSLITE=** parameter. This feature is used by ALCS to inform the ALC 1977 terminal that an unsolicited message is waiting to be printed on the terminal. The **VTAM3270** and **NETVIEW** resource types do not support ALC printers, therefore this feature is not relevant for those communication resource types.

COMTC ADD and COMTC REPLACE considerations: The terminal resource must be an ALC 1977 printer if the unsolicited light feature is to be used. Ensure that it is defined as an ALC 1977 printer in the field CT1TRM. Also ensure that it is not a **NETVIEW** or **VTAM3270** resource type.

CT1DBCS

DBCS support
The communication resource supports the double byte character set. This feature is available only on the VTAM 3270 terminal. In the communications generation, this is provided in the COMDEF **DBCS=** parameter.

COMTC ADD and COMTC REPLACE considerations: The terminal resource must be a **VTAM3270** if the DBCS feature is to be used.

CT1LSM

Inhibit scrolling and send the complete unformatted response to this communication resource.
The response will be one or multiple blocks, (intermediate blocks will terminate with an EOMp character and the final block with an EOMc character) when connected to ALCS via a MATIP TYPE A TCP/IP connection.

Note that the application, which constructs the response may only use the DISPC ADD and SEND macros. The DISPC SEND DEST parameter is not allowed and the DISPC SEND START, NUM, SPACE, BOTTOM, LINE, CONFIRM, CLEAR parameters are ignored.

Extra note: Heap storage is used. To enable this, you must allocate type 3 storage units. Specify a type 3 storage unit size greater than or equal the largest response plus 50 bytes (to allow for message headers and trailers).

COMTC ADD and COMTC REPLACE considerations: The terminal resource must be a **TCPIPALC** or **WASTERM** if this DBCS feature is to be used.

CT1RETR

Command retrieve active
The ALCS message retrieve facility is active for the communication resource. In the communications generation, this is provided in the COMDEF **RETRV=** parameter. The ALCS ZRETR command controls usage of the retrieve facility (start, stop, display, etc.).

COMTC ADD and COMTC REPLACE considerations:
The message retrieve facility is not available to the MQSeries terminal (**MQTERM**) resource types. If the ALCS message retrieve facility is to be immediately available when the communication resource is activated, this indicator must be set on.

CT1SCR

Screen size
One byte field that contains the screen size. In the communications generation, this is provided in the COMDEF **SCRSZE=** parameter. For terminal resources that do not require a screen size, or use the default screen size, this field contains hexadecimal zero. The screen size is required primarily for ALC display terminals. The default screen size for ALC 2915 displays is 12 lines and for ALC 4505 displays is 30 lines. 3270 displays which are **NETVIEW** or **VTAM3270** resource types, do not require a screen size.

COMTC ADD considerations:
This field can be set to hexadecimal zero if the terminal resource does not require a screen size or is to use the system defaults (12 lines for the ALC 2915 display and 30 lines for the ALC 4505 display). The Other System (**OSYS**) resource type supports both ALC and 3270 printers, therefore if a screen size is not provided for an OSYS 3270 display, a default screen size of 24 lines is used.

COMTC REPLACE considerations:
If the screen size for this communication resource is not being changed, do not modify the value provided by COMTC QUERY. If you are changing the screen size, either modify this field with the required size or store hexadecimal zero to request ALCS to use the default screen size.

CT1BUF

Printer buffer size
Halfword field that contains the printer buffer size (in bytes). In the communications generation, this is provided in the COMDEF **BUFSZE=** parameter. The printer buffer size, which is between 0 and 4000 bytes, is required primarily for ALC printers. 3270 printers which are **NETVIEW** or **VTAM3270** resource types do not require a printer buffer size. If a terminal resource does not require a printer buffer size, this field contains hexadecimal zero.
If a terminal resource does require a printer buffer size, and this field contains hexadecimal zero, field CT1INI (the printer flag byte) will indicate if a zero buffer size is to be used. If the CT1ZERB symbol is set on in field CT1INI (described below), the buffer size is zero, otherwise ALCS uses the default printer buffer size.

COMTC ADD considerations:
This field can be set to hexadecimal zero if the terminal resource does not require a buffer size. If the terminal resource requires a buffer size of zero, set this field to hexadecimal zero and also set on the CT1ZERB symbol in the CT1INI printer flag byte (described below). If the terminal resource is to use the default buffer size, just set this field to hexadecimal zero. ALCS uses a default buffer size of 97 bytes for ALC printers and a default buffer size of 1920 bytes for 3270 printers (when they are an **OSYS** resource type). If a default buffer size is provided on

the **BUFSZE=** parameter on the communication generation **COMGEN** macro, that will be used as the default for ALC printers.

COMTC REPLACE considerations:
If the buffer size for this communication resource is not being changed, do not modify the value provided by COMTC QUERY. If you are changing the buffer size, modify this field with the required size. If you are changing the buffer size to zero, set on the CT1ZERB symbol in the CT1INI printer flag byte (described below). If you want ALCS to use the default buffer size (instead of the size currently being used), store hexadecimal zero in this field (and ensure that symbol CT1ZERB in CT1INI is set off).

CT1TMO

Printer timeout values
Six byte field that contains the printer timeout values. There are three timeout values, and each one occupies a halfword. In the communications generation, this is provided in the COMDEF **TIMEOUT=** parameter. The first halfword contains the answerback timeout in *seconds* (a value between 0 and 300). The second halfword contains the test message transmission interval in *seconds* (a value between 0 and 600). The third halfword contains the retry error count (a value between 0 and 30). Further information on the timeout values can be found in the description of the **COMGEN TIMEOUT=** parameter in the ALCS Installation and Customization manual. Printer timeout values are required only for ALC printers. Printers which are **NETVIEW, OSYS** or **VTAM3270** resource types do not require timeout values. If a terminal resource does not require timeout values, these three halfwords contain hexadecimal zeroes.
If a terminal resource requires timeout values, and one or more of the timeout values in this field is set to hexadecimal zero, field CT1INI (the printer flag byte) will indicate if a value of zero is to be used for any of the timeout values. When the printer flag byte CT1INI (described below) has not been used (and the timeout value in this field is zero), ALCS uses the default printer timeout value.

COMTC ADD considerations:
This field can be set to hexadecimal zero if the terminal resource does not require timeout values. If the terminal resource requires any of the three timeout values to be set to zero, set the required halfwords in this field to hexadecimal zero and also set on the appropriate symbol in the CT1INI printer flag byte (described below). If the terminal resource is to use any of the default timeout values, just set the appropriate halfword in this field to hexadecimal zero. ALCS uses the following default timeout values. A default of 50 seconds for the answerback timeout; a default of 300 seconds for the test message transmission interval; and a default of 3 for the retry error count. If a default timeout value is provided on the **TIMEOUT=** parameter on the communication generation **COMGEN** macro, that will be used as the default for the ALC printers.

COMTC REPLACE considerations:
If the timeout values for this communication resource are not being changed, do not modify the values provided by COMTC QUERY. If you are changing the timeout values, modify this field with the required values. If you are changing any timeout value to zero, set on the appropriate symbol in the CT1INI printer flag byte (described below). If you want ALCS to use the default timeout value instead of the value cur-

rently being used, store hexadecimal zero in the appropriate halfword in this field and ensure that the appropriate symbol in CT1INI is set off.

CT1INI Printer Flag Byte
Flag byte that is used in conjunction with the CT1BUF and CT1TMO fields (as described above) to provide printer buffer and timeout values.

Symbol	**Meaning if on (1)**
CT1ZERB	A buffer size of zero is to be used for the printer. If the CT1BUF field (the printer buffer size) contains hexadecimal zeros, ALCS will use a zero buffer size.
CT1ZER1	An answerback timeout of zero seconds is to be used for the printer. If the first halfword of the CT1TMO field (printer answerback timeout) contains hexadecimal zeros, ALCS will use a zero value for that timeout (ALCS will wait indefinitely for an answerback).
CT1ZER2	A test message transmission interval of zero seconds is to be used for the printer. If the second halfword of the CT1TMO field (test message transmission interval) contains hexadecimal zeros, ALCS will use a zero value for that interval (ALCS will therefore not send test messages).
CT1ZER3	A recovery retry count of zero is to be used for the printer. If the third halfword of the CT1TMO field (recovery retry count) contains hexadecimal zeros, ALCS will use a zero value for the retry count (ALCS will not retry transmission of the message).

COMTC ADD and REPLACE considerations:
This flag is used only when a zero value is required for the printer buffer size or for any of the printer timeout values. Refer to the descriptions of the CT1BUF and CT1TMO fields for more information on the usage of the printer flag byte.

Symbol for the MQSeries terminal resource type: The following symbol must be used for the MQSeries terminal resource type (**MQTERM**).

CT1MQL Owning MQ queue resource
Eight byte field that contains the CRN of the MQ queue resource through which ALCS accesses the MQSeries terminal. In the communications generation, this CRN is provided in the COMDEF **LINK=** parameter. The MQ queue resource is defined in the **MQ** communication resource type in the ALCS communications generation (**COMDEF LDTYPE=MQ**).

COMTC ADD considerations:
The CRN of the owning MQ queue resource ***must*** be provided for the MQSeries terminal. ALCS does not provide a default owning MQ queue resource name. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Ensure that the communications definition for the owning MQ queue resource has already been defined in your ALCS communications generation.

COMTC REPLACE considerations:

If the owning MQ queue resource is not being changed, do not modify the value provided by COMTC QUERY. If the CRN of the owning MQ queue resource is being changed, ensure that the communications definition for the owning resource has already been defined in your ALCS system. If the CRN is less than 8 characters, it should be left justified and padded with spaces.

Symbol for the Websphere Application Server terminal resource type: The following symbol must be used for the Websphere Application Server terminal resource type (**WASTERM**).

CT1WSL Owning WAS connection resource
Eight byte field that contains the CRN of the WAS connection resource through which ALCS accesses the Websphere Application Server terminal. In the communications generation, this CRN is provided in the COMDEF **LINK=** parameter. The WAS connection resource is defined in the **WAS** communication resource type in the ALCS communications generation (**COMDEF LDTYPE=WAS**).

COMTC ADD considerations:
The CRN of the owning WAS connection resource ***must*** be provided for the Websphere Application Server terminal. ALCS does not provide a default owning WAS connection resource name. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Ensure that the communications definition for the owning WAS connection resource has already been defined in your ALCS communications generation.

COMTC REPLACE considerations:
If the owning WAS connection resource is not being changed, do not modify the value provided by COMTC QUERY. If the CRN of the owning WAS connection resource is being changed, ensure that the communications definition for the owning resource has already been defined in your ALCS system. If the CRN is less than 8 characters, it should be left justified and padded with spaces.

Symbol for the Other System terminal resource type: The following symbol must be used for the Other System terminal resource type (**OSYS**).

CT1COM Other system communication ID
One byte field that contains the communication ID of the other system that owns this Other System terminal resource. In the communications generation, this communication ID is provided in the COMDEF **COMID=** parameter. The connection between this ALCS system and the other system will normally be an SNA LU6.1 link. The communication ID is defined for the LU6.1 link in the **ALCSLINK** communication resource type in the ALCS communications generation (**COMDEF LDTYPE=ALCSLINK**).

COMTC ADD considerations:
The communication ID of the other system ***must*** be provided for the Other System terminal resource. ALCS does not provide a default communication ID. The communication ID must be an alphabetic character A through Z. Ensure that the other system communication ID has already been defined in your ALCS communications generation.

COMTC REPLACE considerations:

If the other system communication ID is not being changed, do not modify the value provided by COMTC QUERY. If the communication ID of the other system is being changed, ensure it is already defined in your ALCS system. The communication ID must be an alphabetic character A through Z.

Symbols for the TCP/IP ALC terminal resource type: The following symbols can be used for the TCP/IP ALC terminal resource type (**TCPIPALC**).

CT1CDE Output message translation
Flag byte that defines how ALCS should translate output messages sent over TCP/IP to this terminal resource. In the communications generation, this is provided in the COMDEF **CODE=** parameter.

Symbol	**Meaning if on (1)**
CT1CALC	ALCS will translate the output message using the **ALC** transmission code.
CT1IAT7	ALCS will translate the output message using the **IATA7** transmission code.
CT1IAT5	ALCS will translate the output message using the **IATA5** transmission code.
CT1NONE	ALCS will not translate the output message.

COMTC ADD considerations:
If you do not specify the translation option for output messages, ALCS will use the default option of translating from EBCDIC to ALC.

COMTC REPLACE considerations:
If the translation option is not being changed, do not modify the value provided by COMTC QUERY. If the translation option is being changed, set the contents of this flag byte using the appropriate symbol. If you set the contents of this flag byte to hexadecimal zero, ALCS will use the default translation option of translating from EBCDIC to ALC.

CT1HEX H1/H2 address values
Halfword field that contains the H1 and H2 address values for this terminal's address in the SITA network. This field may be required when the terminal communicates with ALCS through a MATIP TCP/IP server resource. The H1 and H2 address values, when combined with the address values A1 and A2 (described below) constitute the address of the ASCU (Agent Set Control Unit or terminal cluster). In the communications generation, the H1 and H2 address values are provided in the COMDEF **HEX=** parameter.

COMTC ADD considerations:
If the H1 and H2 address values are not being provided, this field must contain hexadecimal zeroes.

COMTC REPLACE considerations:
If the H1 and H2 address values are not being changed, do not modify the value provided by COMTC QUERY. If the current H1 and H2 address values are being changed, place the new values in this field. If the current H1 and H2 address values are being deleted, set the contents of this field to hexadecimal zero.

CT1TCI

A1 address value
One byte field that contains the A1 address value for this terminal's address in the SITA network. The A1 address value, when combined with the address values H1/H2 (described above) and the A2 address value (described below) constitute the address of the ASCU (Agent Set Control Unit or terminal cluster). In the communications generation, the A1 address value is provided in the COMDEF **TCID=** parameter.

COMTC ADD considerations:
If the terminal is communicating with ALCS through a MATIP TCP/IP server connection, the A1 address value ***must*** be provided for this TCP/IP terminal resource. If the A1 address value is not being provided, this field must contain hexadecimal zero.

COMTC REPLACE considerations:
If the A1 address value is not being changed, do not modify the value provided by COMTC QUERY. If the current A1 address value is being changed, place the new value in this field. If the current A1 address value is being deleted, set the contents of this field to hexadecimal zero.

CT1CIA

A2 address value
One byte field that contains the A2 address value for this terminal's address in the SITA network. The A2 address value, when combined with the address values H1, H2 and A1 (described above) constitute the address of the ASCU (Agent Set Control Unit or terminal cluster). In the communications generation, the A2 address value is provided in the COMDEF **IA=** parameter.

COMTC ADD considerations:
If the terminal is communicating with ALCS through a MATIP TCP/IP server connection, the A2 address value ***must*** be provided for this TCP/IP terminal resource. If the A2 address value is not being provided, this field must contain hexadecimal zero.

COMTC REPLACE considerations:
If the A2 address value is not being changed, do not modify the value provided by COMTC QUERY. If the current A2 address value is being changed, place the new value in this field. If the current A2 address value is being deleted, set the contents of this field to hexadecimal zero.

CT1CTA

Terminal address
One byte field that contains the terminal address. If this terminal resource is attached to an ASCU (Agent Set Control Unit or terminal cluster), the address of the ASCU is provided in the H1, H2, A1 and A2 address values described above. In the communications generation, the terminal address is provided in the COMDEF **TA=** parameter.

COMTC ADD considerations:
If the terminal is communicating with ALCS through a MATIP TCP/IP server connection, the terminal address ***must*** be provided for this TCP/IP terminal resource. If the terminal address is not being provided, this field must contain hexadecimal zero.

COMTC REPLACE considerations:
If the terminal address is not being changed, do not modify the value provided by COMTC QUERY. If the current terminal address is being changed, place the new value in this field. If the current terminal

address is being deleted, set the contents of this field to hexadecimal zero.

CT1TCL

Owning TCP/IP server or client.
Eight byte field that contains the CRN of the TCP/IP server or client connection through which ALCS accesses this TCP/IP terminal resource. In the communications generation, this CRN is provided in the COMDEF **LINK=** parameter. The TCP/IP server or client connection is defined in the **TCPIP** communication resource type in the ALCS communications generation (**COMDEF LDTYPE=TCPIP**).

COMTC ADD considerations:
If the terminal is communicating with ALCS through a MATIP TCP/IP server or client connection, the CRN of the owning TCP/IP server connection ***must*** be provided for this TCP/IP terminal resource. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Ensure that the communications definition for the owning TCP/IP server or client connection has already been defined in the ALCS communications generation.

COMTC REPLACE considerations:
If the owning TCP/IP server or client connection is not being changed, do not modify the value provided by COMTC QUERY. If the CRN of the owning TCP/IP server or client connection is being changed, ensure that the communications definition for the owning resource has already been defined in your ALCS system. If the CRN is less than 8 characters, it should be left justified and padded with spaces.

CT1RHT

Gateway IP address
Sixteen byte field that contains the remote IP address of the gateway to which this TCP/IP terminal is attached. In the communications generation, this is provided in the COMDEF **RHOST=** parameter. The gateway IP address is a string of 1 to 15 characters with dotted decimal notation (for example, 9.190.314.167).

COMTC ADD considerations:
If the IP address is less than 15 characters, it should be left justified and padded with spaces. If the gateway IP address is not being provided, this field must contain hexadecimal zero.

COMTC REPLACE considerations:
If the gateway IP address is not being changed, do not modify the value provided by COMTC QUERY. If the IP address is being changed, it must be less than 15 characters and should be left justified and padded with spaces. If the gateway IP address is being deleted, set the contents of this field to hexadecimal zero.

CT2RH2

Gateway IP address
Sixteen byte field that contains an alternate IP address of the gateway to which this TCP/IP terminal is attached. In the communications generation, this is provided in the COMDEF **RHOST=** parameter. The gateway IP address is a string of 1 to 15 characters with dotted decimal notation (for example, 9.190.314.167).

COMTC ADD considerations:
If the IP address is less than 15 characters, it should be left justified and padded with spaces. If the gateway IP address is not being provided, this field must contain hexadecimal zero.

COMTC REPLACE considerations:
If the IP address is not being changed, do not modify the value provided by COMTC QUERY. If the IP address is being changed, it must be 15 characters or less and should be left justified and padded with spaces. If the IP address is being deleted, set the contents of this field to hexadecimal zero.

CT2RH3 Gateway IP address
Sixteen byte field that contains an alternate IP address of the gateway to which this TCP/IP terminal is attached. In the communications generation, this is provided in the COMDEF **RHOST=** parameter. The gateway IP address is a string of 1 to 15 characters with dotted decimal notation (for example, 9.190.314.167).

COMTC ADD considerations:
If the IP address is less than 15 characters, it should be left justified and padded with spaces. If the gateway IP address is not being provided, this field must contain hexadecimal zero.

COMTC REPLACE considerations:
If the IP address is not being changed, do not modify the value provided by COMTC QUERY. If the IP address is being changed, it must be 15 characters or less and should be left justified and padded with spaces. If the IP address is being deleted, set the contents of this field to hexadecimal zero.

CT2RH4 Gateway IP address
Sixteen byte field that contains an alternate IP address of the gateway to which this TCP/IP terminal is attached. In the communications generation, this is provided in the COMDEF **RHOST=** parameter. The gateway IP address is a string of 1 to 15 characters with dotted decimal notation (for example, 9.190.314.167).

COMTC ADD considerations:
If the IP address is less than 15 characters, it should be left justified and padded with spaces. If the gateway IP address is not being provided, this field must contain hexadecimal zero.

COMTC REPLACE considerations:
If the IP address is not being changed, do not modify the value provided by COMTC QUERY. If the IP address is being changed, it must be 15 characters or less and should be left justified and padded with spaces. If the IP address is being deleted, set the contents of this field to hexadecimal zero.

Symbol for the VTAM 3270 terminal resource type: The following symbol must be used for the VTAM 3270 terminal resource type (**VTAM3270**).

CT1PRI Resource priority
One byte field that defines the priority of this resource. In the communications generation, this is provided in the COMDEF **PRI=** parameter. During communication initialization, ALCS acquires the VTAM 3270 resources in priority order, starting with priority zero.

COMTC ADD considerations:
Set the resource priority at a value between zero and 255. If you require ALCS to use the default priority (which is zero) set the content of this field to hexadecimal zero.

COMTC REPLACE considerations:
If the resource priority is not being changed, do not modify the value provided by COMTC QUERY. If the resource priority is being changed, set it at a value between zero and 255.

Symbols for the VTAM/ALCI ALC terminal resource type: The following symbols can be used for the VTAM/ALCI ALC terminal resource type (**VTAMALC**).

CT1LEI

LEID address
Three byte field that contains the Logical End-point IDentifier (LEID) address for this ALC terminal. In the communications generation, this is provided in the COMDEF **LEID=** parameter. The LEID, which is 6 hexadecimal digits, is the address by which this terminal is known by ALCI.

COMTC ADD considerations:
The LEID address ***must*** be provided for the VTAM/ALCI terminal. ALCS does not provide a default LEID. The LEID address is 6 hexadecimal digits.

COMTC REPLACE considerations:
If the LEID address for the VTAM/ALCI terminal is not being changed, do not modify the address provided by COMTC QUERY. If the LEID address is being changed, this field must contain 6 hexadecimal digits.

CT1NEF

Owning ALCI resource
Eight byte field that contains the CRN of the ALCI logical unit (LU) through which ALCS accesses this ALC terminal. In the communications generation, this CRN is provided in the COMDEF **NEFLU=** parameter. The ALCI LU is defined in the ALCS communications generation by the **VTAMALC** communication resource type (when the **TERM=NEFLU** parameter is used).

COMTC ADD considerations:
If the CRN of the owning ALCI resource is less than 8 characters, it should be left justified and padded with spaces. If an owning ALCI resource is not being defined for the ALC terminal, this field should contain spaces or hexadecimal zero. ALCS does not provide a default owning ALCI resource name. Ensure that the communications definition for the owning ALCI resource has already been defined in your ALCS communications generation.

COMTC REPLACE considerations:
If the owning ALCI resource is not being changed, do not modify the value provided by COMTC QUERY. If the CRN of the owning ALCI resource is being changed, ensure that the communications definition for the owning resource has already been defined in your ALCS system. If the CRN is less than 8 characters, it should be left justified and padded with spaces.

Symbols for the X.25 PVC resource type: The following symbols can be used for the X.25 permanent virtual circuit (PVC) resource type (**X25PVC**).

CT1PRT

X.25 PVC type
Flag byte that defines the type of X.25 permanent virtual circuit. In the communications generation, this is provided in the COMDEF **PRTCOL=** parameter. ALCS supports seven types of PVC, of which

three types are used for conversational (Type A) traffic from ALC terminals.

Symbol	Meaning if on (1)
CT1TYP1	This is a Type 1 PVC. This PVC type does not have multiple interchanges addresses or multiple terminal circuit identities.
CT1TYP6	This is a Type 6 PVC. This PVC type includes multiple interchanges addresses (IA's). The interchange addresses are provided in the X.25 ALC terminal definition. The description of the CT1CIA symbol (see below) includes information about the interchange address.
CT1TYP7	This is a Type 7 PVC. This PVC type includes multiple terminal circuit identities (TCID's). The terminal circuit identities are provided in the X.25 ALC terminal definition. The description of the CT1TCI symbol (see below) includes information about the terminal circuit identifier.

COMTC ADD considerations:
You can specify only PVC types that are used for Type A traffic. These are types 1, 6 and 7. If you do not specify the PVC type for this communication resource, ALCS will use Type 1 PVC as the default.

COMTC REPLACE considerations:
The X.25 PVC type can not be changed, therefore do not modify the value provided by COMTC QUERY.

CT1CDE Output message translation
Flag byte that defines how ALCS should translate output messages sent over X.25 to this terminal resource. In the communications generation, this is provided in the COMDEF **CODE=** parameter.

Symbol	Meaning if on (1)
CT1CALC	ALCS will translate the output message using the **ALC** transmission code.
CT1IAT7	ALCS will translate the output message using the **IATA7** transmission code.
CT1IAT5	ALCS will translate the output message using the **IATA5** transmission code.
CT1NONE	ALCS will not translate the output message.

COMTC ADD considerations:
If you do not specify the translation option for output messages, ALCS will use the default option of translating from EBCDIC to ALC.

COMTC REPLACE considerations:
If the translation option is not being changed, do not modify the value provided by COMTC QUERY. If the translation option is being changed, set the contents of the this flag byte using the appropriate symbol. If you set the contents of this flag byte to hexadecimal zero, ALCS will use the default translation option (EBCDIC to ALC).

CT1PRI

Resource priority
One byte field that defines the priority of this resource. In the communications generation, this is provided in the COMDEF **PRI=** parameter. During communications initialization, ALCS acquires the X.25 PVC resources in priority order, starting with priority zero.

COMTC ADD considerations:
Set the resource priority at a value between zero and 255. If you require ALCS to use the default priority (which is zero) set the content of this field to hexadecimal zero.

COMTC REPLACE considerations:
If the resource priority is not being changed, do not modify the value provided by COMTC QUERY. If the resource priority is being changed, set it at a value between zero and 255.

Symbols for the X.25 ALC terminal resource type: The following symbols can be used for the X.25 ALC terminal resource type (**X25ALC**).

CT1PVC

Owning X.25 PVC
Eight byte field that contains the CRN of the X.25 permanent virtual circuit (PVC) through which ALCS accesses this ALC terminal. In the communications generation, this CRN is provided in the COMDEF **PVC=** parameter. The ALCS communication definition for the owning X.25 PVC will reside on the OCTM database.

COMTC ADD considerations:
The CRN of the X.25 PVC resource ***must*** be provided for the X.25 ALC terminal resource. ALCS does not provide a default name for the X.25 PVC. If the CRN is less than 8 characters, it should be left justified and padded with spaces. Ensure that the communications definition for the owning X.25 PVC resource has already been defined in the OCTM database.

COMTC REPLACE considerations:
The name of the owning X.25 PVC resource for this ALC terminal can not be changed, therefore do not modify the value provided by COMTC QUERY.

CT1TCI

Terminal circuit identity
One byte field that contains the terminal circuit identity for the terminal controller that this terminal is attached to. In the communications generation, the terminal circuit identity is provided in the COMDEF **TCID=** parameter. This field is required when the X.25 PVC through which ALCS accesses this terminal is defined as a Type 7 PVC. The terminal circuit identity is 2 hexadecimal digits.

COMTC ADD considerations:
This field must contain hexadecimal zero when the owning X.25 PVC is Type 1 or Type 6. When the owning X.25 PVC is Type 7, this field ***must*** contain a terminal circuit identity (ALCS does not provide a default).

COMTC REPLACE considerations:
If the terminal circuit identity is not being changed, do not modify the value provided by COMTC QUERY. If the owning X.25 PVC is Type 7, and the the current terminal circuit identity is being changed, place the new value (as 2 hexadecimal digits) in this field.

CT1CIA Interchange address
One byte field that contains the interchange address for the terminal controller that this terminal is attached to. In the communications generation, this interchange address is provided in the COMDEF **IA=** parameter. This field is used when the X.25 PVC through which ALCS accesses this terminal is defined as a Type 6 or Type 7 PVC. The interchange address is 2 hexadecimal digits.

COMTC ADD considerations:
This field must contain hexadecimal zero when the X.25 PVC is Type 1. When the X.25 PVC is Type 6 or 7, this field ***must*** contain an interchange address. The interchange address can not be hexadecimal zero and ALCS does not provide a default interchange address. The CRN for this ALC terminal is comprised of a 4-character base CRN followed by an interchange address and a terminal address. When the **CRN=** parameter on the COMTC ADD macro is coded for this ALC terminal, ensure that the full CRN includes this interchange address.

COMTC REPLACE considerations:
The interchange address for this ALC terminal can not be changed, therefore do not modify the value provided by COMTC QUERY.

CT1CTA Terminal address
One byte field that contains the terminal address of this ALC terminal. In the communications generation, the terminal address is provided in the COMDEF **TA=** parameter. The terminal address is 2 hexadecimal digits.

COMTC ADD considerations:
The terminal address ***must*** be provided for the resource. ALCS does not provide a default terminal address. The terminal address can not be hexadecimal zero.
If the owning X.25 PVC for this ALC terminal is Type 1, the CRN is comprised of a 6-character base CRN followed by this terminal address.
If the owning X.25 PVC for this ALC terminal is Type 6 or Type 7, the CRN is comprised of a 4-character base CRN followed by an interchange address (described above) plus this terminal address. When the **CRN=** parameter on the COMTC ADD macro is coded for this ALC terminal, ensure that the full CRN includes this terminal address.

COMTC REPLACE considerations:
The terminal address for this ALC terminal can not be changed, therefore do not modify the value provided by COMTC QUERY.

Register use

Not applicable. CT1TM does not generate executable code.

Loss of control

Not applicable. CT1TM does not generate executable code.

Example

See the section which describes the COMTC QUERY, COMTC ADD and COMTC REPLACE macros.

Summary of data required for each resource type

The CT1TM DSECT provides a large number of fields for accommodating the requirements of the eight resource types supported by OCTM. When this DSECT is used in conjunction with the COMTC macro (for example, COMTC ADD), data should be provided only in those fields that are relevant to the specific resource type that the COMTC macro is being issued for. The following provides a summary of the fields that are relevant to each resource type.

Fields for MQTERM resources

The following fields must contain data:

CT1TYP	Communication resource type (MQTERM)
CT1APL	Application name
CT1MQL	CRN of owning MQ queue resource
CT1TRM	Terminal device type

The following fields are optional:

CT1USD	Communication resource user data	
CT1UPD	User data pad character	
CT1IST	Initial status	
CT1UID	Default user ID	
CT1ASD	CRN of associated resource	
CT1CRS	Alternate CRAS type	
CT1ALT	Alternate CRAS number	
CT1FLB	Flag byte for terminal characteristics:	
	CT1SYSS	Suppress system generated messages
	CT1TABC	Tabular skip feature
CT1SCR	ALC screen size	
CT1BUF	Printer buffer size	
CT1TMO	Printer timeout values	
CT1INI	Flag byte for printer buffer and timeout values	

Fields for WASTERM resources

The following fields must contain data:

CT1TYP	Communication resource type (WASTERM)
CT1APL	Application name
CT1WSL	CRN of owning WAS connection resource
CT1TRM	Terminal device type

The following fields are optional:

CT1USD	Communication resource user data
CT1UPD	User data pad character
CT1IST	Initial status
CT1UID	Default user ID
CT1ASD	CRN of associated resource
CT1CRS	Alternate CRAS type

CT1ALT	Alternate CRAS number
CT1FLB	Flag byte for terminal characteristics:
	CT1SYSS Suppress system generated messages
	CT1TABC Tabular skip feature
CT1SCR	ALC screen size
CT1BUF	Printer buffer size
CT1TMO	Printer timeout values
CT1INI	Flag byte for printer buffer and timeout values
CT1LSM	Inhibit scrolling and send the complete response to this communication resource.

Fields for NETVIEW resources

The following fields must contain data:

CT1TYP	Communication resource type (NETVIEW)
CT1APL	Application name

The following fields are optional:

CT1USD	Communication resource user data
CT1UPD	User data pad character
CT1ASD	CRN of associated resource
CT1CRS	Alternate CRAS type
CT1ALT	Alternate CRAS number
CT1TRM	Terminal device type
CT1FLB	Flag byte for terminal characteristics:
	CT1FALL CRAS fallback candidate
	CT1SYSS Suppress system generated messages
	CT1RETR Command retrieve active

Fields for OSYS resources

The following fields must contain data:

CT1TYP	Communication resource type (OSYS)
CT1APL	Application name
CT1TRM	Terminal device type
CT1CSI	Cross system identifier
CT1COM	Other system communication ID

The following fields are optional:

CT1USD	Communication resource user data
CT1UPD	User data pad character
CT1IST	Initial status
CT1UID	Default user ID
CT1ASD	CRN of associated resource
CT1CRS	Alternate CRAS type
CT1ALT	Alternate CRAS number
CT1FLB	Flag byte for terminal characteristics:
	CT1SYSS Suppress system generated messages
	CT1TABC Tabular skip feature
	CT1UNSL Unsolicited light feature
	CT1RETR Command retrieve active
CT1SCR	ALC screen size

CT1BUF	Printer buffer size
CT1INI	Flag byte for printer buffer size

Fields for TCPIPALC resources

The following fields must contain data:

CT1TYP	Communication resource type (TCPIPALC)
CT1APL	Application name
CT1TRM	Terminal device type

The following fields are optional:

CT1USD	Communication resource user data
CT1UPD	User data pad character
CT1IST	Initial status
CT1FLA	Flag byte for the *logon* status:
	CT1LOGO Logon required
CT1UID	Default user ID
CT1ASD	CRN of associated resource
CT1CRS	Alternate CRAS type
CT1ALT	Alternate CRAS number
CT1CSI	Cross system identifier
CT1FLB	Flag byte for terminal characteristics:
	CT1SYSS Suppress system generated messages
	CT1TABC Tabular skip feature
	CT1UNSL Unsolicited light feature
	CT1RETR Command retrieve active
CT1SCR	ALC screen size
CT1BUF	Printer buffer size
CT1TMO	Printer timeout values
CT1INI	Flag byte for printer buffer and timeout values
CT1CDE	Output message translation code
CT1HEX	H1/H2 address values
CT1TCI	A1 address value
CT1CIA	A2 address value
CT1CTA	Terminal address
CT1TCL	CRN of owning TCP/IP server
CT1RHT	Gateway IP address
CT1LSM	Inhibit scrolling and send the complete response to this communication resource.

Fields for VTAM3270 resources

The following fields must contain data:

CT1TYP	Communication resource type (VTAM3270)
CT1APL	Application name

The following fields are optional:

CT1USD	Communication resource user data
CT1UPD	User data pad character
CT1IST	Initial status
CT1FLA	Flag byte for the *logon* and *test* status:
	CT1LOGO Logon required

	CT1TEST	Test resource
CT1UID	Default user ID	
CT1ASD	CRN of associated resource	
CT1CRS	Alternate CRAS type	
CT1ALT	Alternate CRAS number	
CT1TRM	Terminal device type	
CT1CSI	Cross system identifier	
CT1FLB	Flag byte for terminal characteristics:	
	CT1FALL	CRAS fallback candidate
	CT1SYSS	Suppress system generated messages
	CT1DBCS	DBCS support
	CT1RETR	Command retrieve active
CT1PRI	Resource priority	

Fields for VTAMALC resources

The following fields must contain data:

CT1TYP	Communication resource type (VTAMALC)
CT1APL	Application name
CT1TRM	Terminal device type
CT1LEI	LEID address

The following fields are optional:

CT1USD	Communication resource user data	
CT1UPD	User data pad character	
CT1IST	Initial status	
CT1FLA	Flag byte for the *logon* and *test* status:	
	CT1LOGO	Logon required
	CT1TEST	Test resource
CT1UID	Default user ID	
CT1ASD	CRN of associated resource	
CT1CRS	Alternate CRAS type	
CT1ALT	Alternate CRAS number	
CT1CSI	Cross system identifier	
CT1FLB	Flag byte for terminal characteristics:	
	CT1SYSS	Suppress system generated messages
	CT1TABC	Tabular skip feature
	CT1UNSL	Unsolicited light feature
	CT1RETR	Command retrieve active
CT1SCR	ALC screen size	
CT1BUF	Printer buffer size	
CT1TMO	Printer timeout values	
CT1INI	Flag byte for printer buffer and timeout values	
CT1NEF	CRN of owning ALCI resource	

Fields for X25PVC resources

The following fields must contain data:

CT1TYP	Communication resource type (X25PVC)
CT1APL	Application name

The following fields are optional:

CT1USD	Communication resource user data	
CT1UPD	User data pad character	
CT1IST	Initial status	
CT1FLA	Flag byte for the *test* status:	
	CT1TEST	Test resource
CT1UID	Default user ID	
CT1PRT	X.25 PVC type (1, 6 or 7)	
CT1CDE	Output message translation code	
CT1PRI	Resource priority	

Fields for X25ALC resources

The following fields must contain data:

CT1TYP	Communication resource type (X25ALC)
CT1APL	Application name
CT1TRM	Terminal device type
CT1PVC	CRN of owning X.25 PVC
CT1CTA	Terminal address

The following fields are optional:

CT1USD	Communication resource user data	
CT1UPD	User data pad character	
CT1IST	Initial status	
CT1FLA	Flag byte for the *logon* and *test* status:	
	CT1LOGO	Logon required
	CT1TEST	Test resource
CT1UID	Default user ID	
CT1ASD	CRN of associated resource	
CT1CRS	Alternate CRAS type	
CT1ALT	Alternate CRAS number	
CT1CSI	Cross system identifier	
CT1FLB	Flag byte for terminal characteristics:	
	CT1SYSS	Suppress system generated messages
	CT1TABC	Tabular skip feature
	CT1UNSL	Unsolicited light feature
	CT1RETR	Command retrieve active
CT1SCR	ALC screen size	
CT1BUF	Printer buffer size	
CT1TMO	Printer timeout values	
CT1INI	Flag byte for printer buffer and timeout values	
CT1TCI	Terminal circuit identity	
CT1CIA	Interchange address	

Related information

"COMTC QUERY – Query status of communication resource or group" on page 46.
"COMTC ADD – Add a new communication resource" on page 57.
"COMTC REPLACE – Change a communication resource" on page 64.
"CT2TM – Communications Resource and Group Information DSECT" on page 123.

CT2TM – Communications Resource and Group Information DSECT

Format

```
CT2TM REG=reg[,SUFFIX=s]
```

REG=*reg*
Base register for DSECT addressability.

SUFFIX=*s*
Suffix for this DSECT, a single alphanumeric character.

Description

Use the CT2TM DSECT to reference fields in the Communications Resource and Group Information data block. When the **COMTC QUERY** macro is used to obtain information from ALCS about a communication resource or a communications group, ALCS provides the information in a Communications Resource and Group Information data block. This data block is 4000-bytes long (an L3 size block) and contains information either for a communication resource or a communications group.

When communication resource information is provided in the CT2TM DSECT, the communication resource can belong to any communication resource type. For many communication resource types, the information is obtained from the Online Communication Table Maintenance (OCTM) database, and for other communication resource types the information is obtained from the ALCS online communication table.

The CT2TM DSECT defines the following symbols for fields within the Communications Resource and Group Information data block.

CT2FLG Flag byte that identifies the type of data in this block (communication resource data or communications group data) and where the data has originated from (the OCTM database or the online communication table).

Symbol	**Meaning if on (1)**
CT2FLGA	This data block contains communication resource information obtained from the OCTM database
CT2FLGB	This data block contains communication resource information obtained from the ALCS online communication table
CT2FLGG	This data block contains information about a communications group on the OCTM database

CT2NME Eight byte field that contains the communication resource name (CRN) of the communication resource or the name of the communications group. If the field contains a CRN, it will be an alphanumeric string of 1 to 8 characters, left justified and padded with spaces. If the field contains a communications group name, it will be an alphanumeric string of 4 to 7 characters, left justified and padded with spaces.

When the block provides information that has been obtained from the OCTM database (for a ***communications resource*** or a ***communications group***) the following fields in the CT2TM DSECT contain data.

CT2TME Eight byte field that contains the time, in HH.MM.SS character format, when the last status change occurred.

CT2DTE Eight byte field that contains the date, in YYYY.DDD format, when the last status change occurred.

CT2LTM Eight byte field that contains the time, in HH.MM.SS character format, when COMTC LOAD was issued (if COMTC LOAD has not been issued yet, this field contains binary zeroes).

CT2LDT Eight byte field that contains the date, in YYYY.DDD format, when COMTC LOAD was issued (if COMTC LOAD has not been issued yet, this field contains binary zeroes).

CT2BTM Eight byte field that contains the time, in HH.MM.SS character format, when COMTC BACKOUT was issued (if COMTC BACKOUT has not been issued, this field contains binary zeroes).

CT2BDT Eight byte field that contains the date, in YYYY.DDD format, when COMTC BACKOUT was issued (if COMTC BACKOUT has not been issued, this field contains binary zeroes).

CT2STA Flag byte that identifies the current status of the communication resource or communications group. If there are no change requests currently active, the flag byte contains binary zeroes.

Symbol	**Meaning if on (1)**
CT2CHNG	Change requests have been received for the communication resource or group, but they are not currently loaded
CT2LOAD	Change requests for the communication resource or group have been loaded by a COMTC LOAD macro
CT2BACK	Change requests for the communication resource or group have been backed out by a COMTC BACKOUT macro
CT2CONF	Change requests for the communication resource or group have been confirmed by a COMTC CONFIRM macro
CT2COMM	Change requests for the communication resource or group have been committed by a COMTC COMMIT macro

CT2PRG Flag byte that identifies the current processing status of the communication resource or communications group. If there is no COMTC macro currently being processed, the flag byte contains binary zeroes.

Symbol	**Meaning if on (1)**
CT2CGPR	Change requests are currently being processed for the communication resource or group
CT2LDPR	A COMTC LOAD macro is currently being processed for the communication resource or group

CT2BKPR A COMTC BACKOUT macro is currently being processed for the communication resource or group

CT2CFPR A COMTC CONFIRM macro is currently being processed for the communication resource or group

CT2CMPR A COMTC COMMIT macro is currently being processed for the communication resource or group

When the block provides information about a ***communications group***, the following fields in the CT2TM DSECT contain data.

CT2GTM Eight byte field that contains the time, in HH.MM.SS character format, when this communications group was allocated by a COMTC ALLOCATE macro.

CT2GDT Eight byte field that contains the date, in YYYY.DDD format, when this communications group was allocated by a COMTC ALLOCATE macro.

CT2CNT Fullword field that contains a count of communication resources that belong to this communications group. This count will be zero if no change requests have been submitted for the communications group, or the group has been committed via COMTC COMMIT.

CT2LST Fullword field that contains a displacement from the beginning of this block to an area that contains a list of communication resources that belong to this communications group. If there are no change requests that belong to this group, this field will contain binary zeroes. The resource list is comprised of 9-byte items each containing an 8-byte Communication Resource Name (CRN) preceded by a 1-byte change request type. The change request type will identify the type of change request that was submitted for this resource. The first CRN and its change request type are in the first 9 bytes, the second CRN and its change request type are in the next 9 bytes, and so on. The 1-byte change request type is "**A**" for COMTC ADD, "**R**" for COMTC REPLACE, and "**D**" for COMTC DELETE. There is a maximum of 400 CRN's in the resource list.

When the block provides information about a ***communication resource*** that is defined on the OCTM database, the following fields in the CT2TM DSECT contain data.

CT2CRI Three byte field that contains the Communication Resource Identifier (CRI) for this communication resource.

CT2ORD Fullword field that contains the communication resource ordinal number for this communication resource.

CT2GRP Seven byte field that contains the name of the communications group that this communication resource belongs to. If it does not belong to a communications group, this field contains spaces. It will be an alphanumeric string of 4 to 7 characters.

CT2CHG Flag byte that identifies the type of change request that is currently in progress for this communication resource. If there are no change requests currently active, the flag byte contains binary zeroes.

Symbol	Meaning if on (1)
CT2RADD	This is a new communication resource that is being *added* by the COMTC ADD macro
CT2RCHG	This is a current communication resource that is being *changed* by the COMTC REPLACE macro
CT2RDEL	This is a current communication resource that is being *deleted* by the COMTC DELETE macro

CT2BCA Fullword field that contains a displacement from the beginning of this block to an area that contains the base definition of this communication resource. For example, if this communication resource is currently being changed by COMTC macros, this base definition will not include those changes. The changes are provided in the field CT2UCA (see below). Use the **OCTM Communication Resource Definition DSECT, CT1TM** to reference the data in this area.

CT2UCA Fullword field that contains a displacement from the beginning of this block to an area that contains the changes that are currently in progress for this communication resource. The COMTC REPLACE macro is used to change a communication resource. If there are no changes currently being processed for this communication resource, this field contains hexadecimal zeroes. Use the **OCTM Communication Resource Definition DSECT, CT1TM** to reference the data in this area.

When the block provides information about a ***communications resource*** whose communication type is **not managed** on the OCTM database (for example, an APPC connection), the resource information will have been obtained from the ALCS online communication table. For these communication resources, the following fields in the CT2TM DSECT contain data.

CT2TYP Two flag bytes that identify the communication resource type that this communication resource belongs to.

The following symbols are for the first flag byte (CT2TYP+0).

Symbol	Meaning if on (1)
CT2TYPAL	An ALCS application resource type
CT2TYPLK	An SNA LU6.1 communication link resource type
CT2TYPAP	An APPC connection resource type
CT2TYPIP	A TCP/IP connection resource type (client or server)
CT2TYPMQ	An MQSeries queue resource type
CT2TYPSC	A BSC point to point resource type
CT2TYPPV	An X.25 PVC resource type (for PVC types 2 to 5)
CT2TYPNF	An ALCI LU resource type

CT2TYP+1 The following symbols are for the second flag byte (CT2TYP+1).

Symbol	Meaning if on (1)
CT2TYPSL	A Virtual SLC link resource type
CT2TYPTY	A WTTY link resource type

CT2TYPWS A Websphere Application Server connection resource type

CT2TYPVT A VTAM 3270 resource type (for Prime and RO CRAS only)

CT2CRI Three byte field that contains the Communication Resource Identifier (CRI) for this communication resource.

CT2ORD Fullword field that contains the communication resource ordinal number for this communication resource.

Register use

Not applicable. CT2TM does not generate executable code.

Loss of control

Not applicable. CT2TM does not generate executable code.

Example

See the section which describes the COMTC QUERY macro.

Related information

"COMTC QUERY – Query status of communication resource or group" on page 46.
"CT1TM – Communication Resource Definition DSECT" on page 96.

CT3TM – Communications Groups Information DSECT

Format

```
CT3TM REG=reg[,SUFFIX=s]
```

REG=*reg*
Base register for DSECT addressability.

SUFFIX=*s*
Suffix for this DSECT, a single alphanumeric character.

Description

Use the CT3TM DSECT to reference fields in the Communications Groups Information data block. When the **COMTC GROUPS** macro is used to obtain information from ALCS about all of the allocated communications groups, ALCS provides the information in Communications Groups Information data blocks. Each data block is 4000-bytes long (an L3 size block) and contains information on a maximum of 100 allocated communications groups. The first data block is attached to an ECB data level and the remaining blocks are held in L3ST pool records. The number of blocks required for holding information on all the communications groups depends on the number of groups that are currently allocated. ALCS obtains information about each communications group from the Online Communication Table Maintenance (OCTM) database.

The CT3TM DSECT defines the following symbols for fields within the Communications Groups Information data block.

The following are the fields which reside in the record header.

CT3BID
Two byte record identifier. The record ID is hexadecimal **AC10** (equate #RIDOCTM in DXCRID)

CT3FCH
Fullword field that contains the file address of a forward chained L3ST pool record. The forward chained record contains information on additional communications groups. When there is no forward chained pool record, this field contains hexadecimal zeroes.

CT3CNT
Fullword field that contains a count of the communications groups described in this storage block (or pool record).

CT3TOT
Fullword field that contains the total count of currently allocated communications groups.

The following are the fields which reside in each communications group item.

CT3ITM
This symbol identifies the beginning of a 46 byte communications group item.

CT3NME
Seven byte field that contains the name of the communications group. It will be an alphanumeric string of 4 to 7 characters, left justified and padded with spaces.

CT3RSC Fullword field that contains a count of communication resources that belong to this communications group. This count will be zero if no change requests have been submitted for the communications group, or the group has been committed via COMTC COMMIT.

CT3GTM Eight byte field that contains the time, in HH.MM.SS character format, when this communications group was allocated by a COMTC ALLOCATE macro.

CT3GDT Eight byte field that contains the date, in YYYY.DDD format, when this communications group was allocated by a COMTC ALLOCATE macro.

CT3TME Eight byte field that contains the time, in HH.MM.SS character format, when the last status change occurred.

CT3DTE Eight byte field that contains the date, in YYYY.DDD format, when the last status change occurred for this communications group.

CT3STA Flag byte that identifies the current status of the communications group. If there are no change requests currently active, the flag byte contains binary zeroes.

Symbol	**Meaning if on (1)**
CT3CHNG	Change requests have been received for the communications group, but they are not currently loaded
CT3LOAD	Change requests for the communications group have been loaded by a COMTC LOAD macro
CT3BACK	Change requests for the communications group have been backed out by a COMTC BACKOUT macro
CT3CONF	Change requests for the communications group have been confirmed by a COMTC CONFIRM macro
CT3COMM	Change requests for the communications group have been committed by a COMTC COMMIT macro

CT3PRG Flag byte that identifies the current processing status of the communications group. If there is no COMTC macro currently being processed, the flag byte contains binary zeroes.

Symbol	**Meaning if on (1)**
CT3LDPR	A COMTC LOAD macro is currently being processed for the communications group
CT3BKPR	A COMTC BACKOUT macro is currently being processed for the communications group
CT3CFPR	A COMTC CONFIRM macro is currently being processed for the communications group
CT3CMPR	A COMTC COMMIT macro is currently being processed for the communications group

Register use

Not applicable. CT3TM does not generate executable code.

Loss of control

Not applicable. CT3TM does not generate executable code.

Example

See the section which describes the COMTC GROUPS macro.

Related information

"COMTC GROUPS – Obtain status of communications groups" on page 51.