Advantage™ VISION:Transact® for CICS®

Getting Started

Release 7.8

Computer Associates®

MAN05193450E
B02569-1E

Contents

Chapter 1: Introduction to VISION:Transact

Appendix A: Using the New Enhancements

Chapter 1

Introduction to VISION:Transact

The Purpose of This Guide

This guide introduces you to Advantage™ VISION:Transact® (hereinafter referred to as VISION:Transact). By the time you have finished reading this guide, you will have an overview of the wide scope of the product and its usability will be familiar to you. It is important to us that you feel comfortable with Advantage VISION:Transact before you begin to use it.

CA Technology Services: Delivering Business Value On Your Terms

CA Technology Services is a global organization of highly trained, experienced professionals who are determined to provide you with the technical expertise you need, when and how you need it. From implementing a CA solution to helping you get the most out of the CA technology that you have, CA Technology Services is committed to delivering business value to you on **your** terms.

Our professionals understand your unique business needs and work closely with you to assess which technology is right for your business. Whether the assignment is large or small or you need a custom, stand-alone, or packaged solution, we tailor our efforts to meet your business demands.

By offering a broad range of flexible services, we help you maximize your investment in our technology, achieve more efficient IT performance, and better manage your infrastructure, security, storage, applications, and data. Such flexibility ensures that you reach your time-to-market goals while improving your business performance.

Why not ask your CA representative for more information about how a CA Technology Services professional can help your organization get the most out of your CA business solutions?

CA Education Services: Ready When You Are

The goal of CA Education Services is to help you realize the full potential of your CA software investment. To meet this goal, our high-quality instructors strive to understand your specific training requirements, and then deliver the right kind of training when, where, and how you need it.

All CA instructors are fully certified and offer a wealth of hands-on enterprise management experience gained in working with today's largest and most complex businesses. Whether your training is web-based, self-paced, or in the traditional classroom, you always receive the most up-to-date instruction and expertise that is available. The knowledge you gain through training prepares you to successfully leverage the capabilities of your CA software.

Why not ask your CA representative how our training and education programs can help you get more out of your CA business solutions?

Computer Associates: Commitment, Quality, Innovation

For more than a quarter century, CA has been developing and supporting software solutions that are currently used by more than 99 percent of the Fortune 500 companies in more than 100 countries. CA is committed to offering leading technologies in flexible partnerships to help you derive full value from your software investments.

At Computer Associates, we are committed to offering simple and meaningful solutions to your complex problems, and to delivering management solutions that offer security, reliability, availability, and performance. We work hard to achieve the highest levels of quality in our solutions to help you meet your changing business needs.

To meet these needs, CA's world-class solutions address all aspects of process management, information management, and infrastructure management with six focus areas:

- Enterprise management
- Security
- Storage
- Portal and business intelligence
- Database management
- Application life cycle management and application development

In addition, our innovative approach to technology is carried over into our innovative business solutions. From a revolutionary new business model to a dedicated customer relationship organization, CA is responding to your changing business needs.

We know what it takes to deliver and support valuable solutions 24 hours a day, 7 days a week, 365 days a year while maintaining the highest standards for quality and innovation:

- We are the first global enterprise software company to meet the exacting standards for worldwide ISO 9002 certification.
- We have earned over 150 patents for innovative software solutions.
- We have the highest caliber software developers and consultants in the industry.

We also know you expect us to stand by our commitments. And we do.

For More Information

After reading this *Getting Started*, you can refer to the numerous resources available to you for additional information. Your product CD contains instructional documents that showcase your software and provide detailed explanations about the product's comprehensive, feature-rich components. In addition, you can obtain procedural information and answers to any questions you may encounter by accessing the Computer Associates web site at ca.com.

VISION:Transact

VISION:Transact is a powerful generator of IMS Transaction Monitor (TM) and CICS Transaction Server (TS) applications. This easy-to-learn product provides the capability to build and thoroughly test complex application development systems using a simple, English-like language, thereby accelerating the development process. Its high-level, non-procedural language results in high productivity, consistent standards, and good programming practices. Advantage VISION:Transact allows developers to build, test, and maintain systems while minimizing procedural coding and maximizing the time and talent devoted to the business considerations of the application.

Advantage VISION:Transact is a software system that can be used to design, implement, and execute data processing applications. Novice users can be productive within a few hours, yet VISION:Transact produces complex applications efficiently. An experienced user can build an application in VISION:Transact in less than half the time it would take for a similar application in COBOL or other third-generation languages.

The philosophy in using VISION:Transact is to:

- Conceptualize how VISION:Transact functions during input, processing, and output
- Design applications that use as many of the automatic functions of VISION:Transact as possible

VISION:Transact provides extensive facilities for the creation and maintenance of files and databases, as well as report generation. VISION:Transact processes virtually any data source and database format available on an IBM® host (for example, flat files, IMS, and DB2) without requiring the programmer to be familiar with these formats.

The VISION:Transact system is made up of the following main components:

- VISION:Transact Application Generator
- VISION:Transact Runtime Library
- VISION:Workbench for ISPF
- Common Library Utilities.

See the *VISION:Transact Reference Guide* for more information about these components.

Features

This section describes important features of VISION:Transact.

CICS and IMS Applications and Web-Services

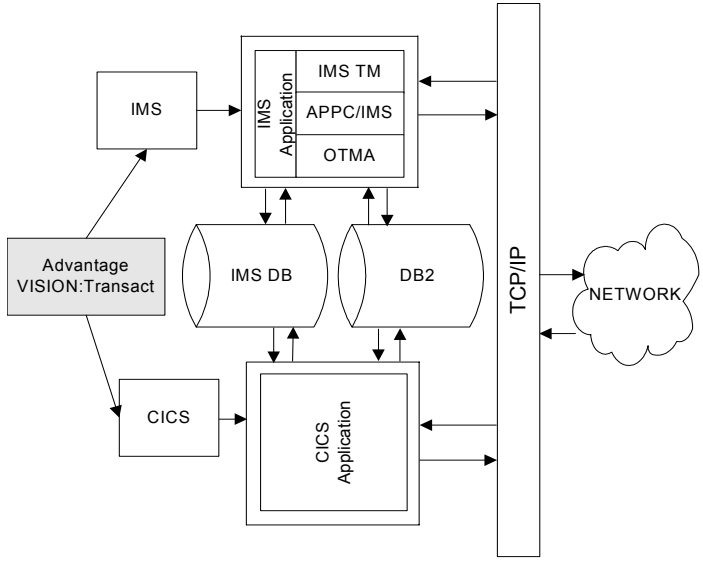
VISION:Transact provides rapid development and deployment of CICS TS and IMS TM applications. This is critical to help you meet ever-changing corporate requirements to provide information to both internal and external users of information. Along with deploying new systems, there is a requirement to make adjustments as the business environment changes. The use of the Internet to get this information to and from end users has also added to the complexity of using IMS and CICS. Mainframe legacy systems continue to be the backbone of many corporations and present the IT organization with new challenges.

Managing Your Legacy Systems

Legacy systems contain valuable information that must be accessible by new systems to deliver complex applications that meet rapidly changing business requirements. Instead of sifting through numerous lines of code, VISION:Transact generates code that reflects your data needs and business rules. Organizations are continuing to find that they need to leverage the business logic, data, and processing power residing on their legacy systems as they move ahead with new eBusiness initiatives. With VISION:Transact, they can accomplish this quickly and easily.

Building CICS and IMS Applications for Use on the Internet

The explosion of the Internet technologies and eBusiness initiatives has resulted in new challenges for IT organizations. Business partners need links to timely information across multiple platforms, and customers are demanding new services delivered in real time. With quick deployment of CICS and IMS applications, along with new technologies to move these applications into the Internet, businesses are able to leverage their current investment in their legacy systems and use the power of the Internet to better serve their customers.



Building Complex Real Time Applications

VISION:Transact uses 4GL statements to define 3270 screen or CICS COMMAREA interactions, processing actions for input/output fields, and database operations. These 4GL statements are compiled into object code that is link-edited into application load modules for execution within the CICS TS or IMS TM environments.

Benefits

VISION:Transact provides:

- A way to quickly build simple-to-complex CICS TS and IMS TM applications
- An easy to lean 4GL language
- A way to exploit the internet for data access
- Simple, transparent access to DB2, IMS, and VSAM

Hardware and Platform Requirements

The following hardware and platforms are required for VISION:Transact:

- All IBM and IBM-compatible enterprise servers running z/OS or OS/390
- Application execution under CICS TS or IMS TM
- Supports VSAM and IMS databases with option for supporting DB2
- Optional ISPF program development through Workbench for ISPF

Environment

This section describes the packaging and environmental considerations for VISION:Transact 7.8.

IBM Operating Systems Supported

VISION:Transact supports the z/OS and OS/390 operating systems.

Available Application Execution Environments

You can use the following application execution environments:

- CICS TS from IBM
- IMS TM from IBM

SMP/E Installation Specifications

The packaging of VISION:Transact Release 7.8 has been changed to conform to the IBM SMP/E standards. SMP/E modification control statements (MCS) along with the supporting JCL for the RECEIVE, APPLY, and ACCEPT processes have been developed to install the elements of the product. The Indirect File method is used to reference the unloaded product data sets during the install process. Problem fixes or post-delivery enhancements to the product are also packaged to conform to SMP/E standards and processes for PTFs and/or APARs.

The Function Modifier Identification (FMID) for VISION:Transact Release 7.8 is CDDDX078.

CA LMP Licensing Specifications

To verify proper authorization to use the product at a customer site, VISION:Transact Release 7.8 uses the CA License Management Program (LMP).

In addition to the base product noted above, VISION:Transact provides the following database interface as an optional feature:

- DB2 Database Interface

Thus, each customer will use one or two product codes depending on their licensed features. There will be one product code for the base system and one code for the optional DB2 Interface feature.

The assigned product codes are as follows:

3I	VISION:Transact
3J	VISION:Transact Interface for DB2

VISION:Transact Release 7.8 has been modified to invoke LMP to verify the authorized use of the VISION:Transact product. This verification occurs during product initialization. If the CAIRIM service is not operational, the product terminates with an appropriate error message.

Documentation

The documentation is delivered on a compact disc (CD). The books are in Adobe Acrobat Portable Document Format (PDF) and are designed for you to read online using the Acrobat Reader (also included on the CD).

Each online book contains a table of contents, index, and underlined colored hypertext links. To go directly to the book, chapter, section, or topic being referenced, click the hypertext link.

The following documentation is available for VISION:Transact Release 7.8 on the documentation compact disc:

Name	Description
Readme file	A file containing system and installation requirements, last minute known issues, and information on contacting technical support.
VISION:Transact for CICS Getting Started Guide	Contains fundamental descriptions of VISION:Transact, along with sections on features, benefits, hardware requirements, and documentation.
VISION:Transact for CICS Installation Guide	Describes how to install VISION:Transact.
VISION:Transact for CICS Reference Guide	Describes all of the concepts necessary to understand how VISION:Transact works and how applications can be developed using VISION:Transact. It covers both basic and advanced applications.

Name	Description
VISION:Transact for CICS ASL Reference Guide	Contains information specific to the use and operation of Advanced Syntax Language (ASL). (This book assumes that you are familiar with VISION:Transact.) Also describes the conversion from fixed form syntax to ASL.
VISION:Transact for CICS Specifications Guide	Contains all valid fixed format syntax specifications for VISION:Transact statements.
VISION:Transact for CICS Messages Guide	Lists VISION:Transact messages and codes and gives detailed explanations. Also assists programmers in identifying and resolving problems that may occur during the execution of VISION:Transact.
VISION:Transact for CICS Environment Guide	Describes detailed information for the use of VISION:Transact and the Common Library subsystem in the CICS environment.
VISION:Transact for CICS Toolkit Reference Guide	Describes the VISION:Transact Toolkit subroutines which enable you to perform various functions that are not readily available using native VISION:Transact programming operations. Also describes how to tailor and use the Toolkit.

Name	Description
VISION:Transact for CICS Workbench for ISPF Reference Guide	Contains a brief introduction to VISION:Workbench for ISPF, an illustrated application, and an in-depth discussion of the panels in VISION:Workbench for ISPF.
VISION:Transact for CICS Release Summary	Contains a brief summary of the new enhancements for the current release of VISION:Transact, and enhancements to existing features.

Appendix

A

Using the New Enhancements

Introduction

The purpose of this appendix is to describe the new capabilities in the VISION:Transact for CICS Release 7.8 product and present sufficient detail for you to begin to develop application programs exploiting the new functionality provided by this release. This release of VISION:Transact introduces the capability of application programs generated by VISION:Transact to run as COMMAREA programs under CICS Transaction Server as well as 3270 Terminal programs. Prior to this release, application programs generated by VISION:Transact could only run as 3270 Terminal programs under CICS Transaction Server.

Existing VISION:Transact application programs intended as 3270 Terminal programs may be compiled with this new release and will continue to execute identically as when compiled with previous releases.

Design Philosophy

The philosophy upon which the new capabilities of VISION:Transact Release 7.8 were developed was to ensure strict source language compatibility with previous releases of the product yet still allow the newly compiled application programs to operate as either a COMMAREA program or a 3270 Terminal program under CICS TS. This objective has been achieved such that existing VISION:Transact source programs can be recompiled using VISION:Transact Release 7.8 and the resulting application

object code will run without modification as either a COMMAREA program or a 3270 Terminal program. The only requirement for a COMMAREA program is that some external program perform the required presentation functions, most likely external to CICS, and then make the data available to the application program via the COMMAREA.

At application program startup time, if a COMMAREA is provided by CICS linkage, the VISION:Transact application program compiled with this new release will run as a COMMAREA program. Otherwise, it will run as a 3270 Terminal program just as it would have with earlier releases of VISION:Transact.

COMMAREA Program Concepts

Within VISION:Transact, the data within the COMMAREA is treated in a similar manner as if it had been received from a 3270 terminal. All of the input validation, conversion, target action, and output formatting capabilities applicable to 3270 screen data in previous releases of VISION:Transact now apply to COMMAREA data as well. Except for layout considerations and such attributes as display intensity, cursor position, and so on, COMMAREA data and 3270 screen data are treated identically. In general, whenever the VISION:Transact documentation discusses the data content of a screen, the equivalent meaning may be applied to the data content of the COMMAREA. The term screen may be assumed to also mean COMMAREA except where otherwise noted.

The existing Screen Definition statements (SD, SL, SS, S0, and Sn) are used to define the 3270 screen or COMMAREA data. Some specifications on these statements may have no effect (for example, 3270 intensity specifications) when the application program is running as a COMMAREA program. This will be discussed more fully in Specification Changes on page A-4.

As part of the application generation process, VISION:Transact will now produce both BMS screen mapping statements and COBOL data definition statements for the COMMAREA. The COBOL data definition statements define the data layout and field contents for the COMMAREA. When the application source code references more than one screen, the COMMAREA becomes a composite area containing all screens referenced by the application - each logical screen becomes a section within the COMMAREA.

When received by a VISION:Transact COMMAREA application program, all data within the COMMAREA is assumed to be EBCDIC character data. Any necessary conversions from ASCII to EBCDIC must occur before the data is received by VISION:Transact.

Operational Considerations

VISION:Transact source programs are compiled in the identical way that they have been in previous releases. The only consideration is that an additional DD statement is recognized and used as the destination for the COBOL data definition of the COMMAREA. The new DD statement is named M5COMMAR and the specified output destination will contain the 80-byte COBOL data definition source statements that describe the COMMAREA. If the DD statement is not provided, no COBOL statements will be output. Following is an example of the JCL statements required for application generation runs (RC statement Run Type = G):

```
//APPGEN   EXEC PGM=MARKV,REGION=2M
//STEPLIB    DD DSN=...
//M5LIB      DD DSN=...         (Common Library dataset)
//M5LIST     DD ...(Source statement listings)
//M5COMMAR   DD ...(COMMAREA definition statements)
//M5TPTBL    DD ...(CICS definition examples)
//M5MAP      DD ...(BMS Mapping statements)
//M5PUNCH    DD ...(Generated object code)
//M5PREP     DD ...(DB2 access module statements)
```

Note that COBOL statements describing the COMMAREA are only output during application generation runs (RC statement Run Type code set to G), never during definition, catalog maintenance, or source statement retrieval runs (RC statement Run Type codes of blank/D or S).

There are no special CICS TS definition requirements for application execution when the application will run as a COMMAREA program. If the generated VISION:Transact application will only run as a COMMAREA program, then there is no need to compile the BMS mapping statements, define the BMS mapset to CICS TS, or define a transaction code to invoke the program.

Specification Changes

This section discusses the specification changes and special considerations that relate to individual Screen Definition statements. As stated earlier, a COMMAREA may be thought of as a logical screen or combination of logical screens. As such, the existing statement types are used to define the screen regardless of whether the definition will be for a real 3270 screen or only for a COMMAREA logical screen. The following sections discuss each of the five screen definition statements: SD, SL, SS, S0, and Sn.

SD Statement

The only specification change on the SD statements is the acceptance of the special DEVICE ID entry (columns 36-43) with a value of COMMAREA to indicate that the screen being defined is intended for a COMMAREA program only and any constraints unique to 3270 Terminal programs do not apply. No BMS mapping statements will be generated for screens that specify a DEVICE ID of COMMAREA.

SL Statement

There are no specification changes for the SL statement. If SL statements are supplied when the SD statement specifies a DEVICE ID of COMMAREA, the only function that the SL statements serve is to specify the length of the field in the COMMAREA via the presence of a field designator character sequence for a field.

SS Statement

There are no specification changes for the SS statement.

S0 Statement

The HIGLIGHTING, CURSOR, ERROR HANDLING, PROTECTED, and DISPLAY ANY MODIFICATION entries have no effect when the application program is used as a COMMAREA program. The NUMERIC entry specification of A will result in no validation being performed for the field when the application program is used as a COMMAREA program. When the application program could potentially be used as a COMMAREA program, a NUMERIC entry specification other than A should be used.

The following changes apply to the EDIT/VALIDATION OPERATOR entry (columns 41-42):

The LN operator and operand may be specified even if the screen field is not a virtual field. When specified for non-virtual fields and a corresponding field designator sequence is present on SL statements, the LN operand value must match the length of the corresponding field designator sequence. When no SL statements are provided, a statement using the LN operator and operand to specify the length of the field is required.

A new NC operator may be specified to provide a name to be used for the field in the COBOL statements that define the COMMAREA. The NC operand may be a valid COBOL element name up to 24 characters long. If no NC operator is provided for a field, the NM operand is used. If neither an NC or an NM operator is provided, the Source/Destination Field Name (columns 17-24) is used as the COBOL element name. The NC and NM operators may both be specified for the same field.

Sn Statement

The EDIT/VALIDATION OPERATOR entry (columns 41-42) changes described for the S0 statement also applies to the Sn statement.

COMMAREA Details

This section documents some details related to the COMMAREA content that external programs invoking a VISION:Transact application program as a COMMAREA program should be aware of. The best way to document this information is by examples.

Example

This very simple application includes a single screen with two input fields and one output field. Note that in this example, the screen definition includes layout (SL) statements for a 3270 screen.

Example 1 - Advantage VISION:Transact source code

```
        1         2         3         4         5         6         7
----+----0----+----0----+----0----+----0----+----0----+----0----+----0--
IVP2    RC                 #   A    GC         IVP2PGM
IVP2SCN1SC              OSOCOMPUTE              IVP2SCN1
IVP2SCN1SDR _@  IVP2    IVP2MAP
IVP2SCN1SL  30 DIVIDEND: _____
IVP2SCN1SL  40 DIVISOR:  _____
IVP2SCN1SL  60 QUOTIENT: _______
IVP2SCN1SS 1    1
IVP2SCN1S0 1   TDIVIDENDIY     Y     P  NM DIVIDEND
IVP2SCN1S0 1   TDIVISOR IY     Y     P  NM DIVISOR
IVP2SCN1S0 1   TQUOTIENTO              NM QUOTIENT
COMPUTE ER                                          O              P
##PROC
DIVIDEND: FIELD F 4
DIVISOR:  FIELD F 4
QUOTIENT: FIELD F 4
;
LET T.QUOTIENT = T.DIVIDEND / T.DIVISOR
##PEND
```

Example 1 - Generated COMMAREA definition statements

```
       1        2        3        4        5        6        7
----+----0----+----0----+----0----+----0----+----0----+----0----+----0--
      01 DFHCOMMAREA.
     *
         05 IVP2SCN1.
            10 DIVIDEND-ATTR                          PIC XXX.
            10 DIVIDEND                               PIC X(05).
            10 DIVISOR-ATTR                           PIC XXX.
            10 DIVISOR                                PIC X(05).
            10 QUOTIENT-ATTR                          PIC XXX.
            10 QUOTIENT                               PIC X(08).
            10 MSGLINE-ATTR                           PIC XXX.
            10 MSGLINE                                PIC X(79).
            10 TRANSACTION-CONTROL REDEFINES MSGLINE.
               15 ATTNID                              PIC X(05).
               15 FILLER                              PIC X.
               15 CONVERSE                            PIC XXX.
               15 FILLER                              PIC X.
               15 CONVERSATION-ID                     PIC X(12).
```

Special Aspects of VISION:Transact COMMAREA

The notes below highlight some special aspects of a VISION:Transact COMMAREA.

1. The MSGLINE is added to the COMMAREA just as it is to a 3270 screen when the field is not explicitly defined for the screen.
2. Each field on the screen becomes two fields in the COMMAREA. The first field defines a 3-byte prefix that will contain special attribute codes when the COMMAREA is returned to the calling program. The characters -ATTR will be appended to the screen field name to form the name of the attribute field. The second field will contain the input and/or output data and will be named by the rules for deriving the screen field name or by the name from the S0/Sn statement NC operator when specified.

3. The 3-byte attribute prefix field will contain special attribute codes when the application program returns the COMMAREA to the calling program. This attribute prefix will be blank when the field is processed as expected and no messages are issued relating to the field. Otherwise it will contain one of the following codes:

   - INV - The field became invalid during internal processing. The field itself will contain an asterisk (*) character.
   - MIS - The field is missing on the source database. The field itself will contain the dash (-) character.
   - OVF - An overflow occurred when converting the source field to the COMMAREA field on output. The field itself will contain a plus (+) character. This will never occur if the source field is already a character type field, even if the field must be truncated in order to fit.
   - A message id such as Fnn, Snn, or Xnn indicating that the error message in the MSGLINE field applies to this field. The error codes are described in the Online Execution Messages section of the *Advantage VISION:Transact for CICS Messages Guide*. Note that the code F00 is used for messages that result from the MS operator on a PR statement. The code F02 is used for messages that result from the MS operator on an Sn statement. Identifying the field with the F00 attribute requires that the SET ERROR command (in ASL) or the SE operation (PR statement) is performed.

4. The MSGLINE field will contain the text of the error message that is output, whether from the standard system messages or from user specified coding. This field will contain blanks if no message is output.

5. The attribute prefix for the MSGLINE field will contain MSG if a message has been generated. If no message has been generated but one of the special attribute codes (INV, MIS, or OVF) has been set for a field, the MSGLINE field attribute prefix will contain the first such code that was applied. Thus, when the COMMAREA is returned to its original source, only the MSGLINE field attribute prefix need be examined to determine if any other fields need be examined for error conditions.

6. The MSGLINE field is over-defined with the Transaction Control group of fields. The Transaction Control fields are examined on input and used as follows:

   - The ATTNID field is used to set the ATTNID flag field and simulates the pressing of various 3270 keys. Values that are recognized and processed appropriately are ENTER, CLEAR PFnnn, AND PAnnn. If the field contains blanks, ENTER is assumed. The appropriate setting of this field can be used to simulate the pressing of specific 3270 keys when an application was originally designed for use with 3270 terminals or is intended to be used as both a 3270 Terminal program and a COMMAREA program. Note that when the ATTNID field value is set to CLEAR, no application program processing will occur, only the cleanup of any CICS Temporary Storage queues left over from previous interactions of a pseudo-conversation (see discussion below). This action is identical to the action taken when the CLEAR key is pressed after an interaction with a VISION:Transact Terminal program.

   - The CONVERSE field is used to indicate whether the COMMAREA program is to operate in a pseudo-conversation mode or not. When the field contains a value of YES, pseudo-conversational mode is assumed. Any other value will indicate that no pseudo-conversational mode is requested. Pseudo-conversational mode is required for a VISION:Transact application program when the application design requires that designated Working Storage and Pseudo File data be saved between invocations of the application program that processes

the pseudo-conversation. Although one would typically not design a COMMAREA program in this manner, this functionality is provided to allow a 3270 Transact program that depends upon designated Working Storage or Pseudo File data be saved and restored between interactions of the pseudo-conversation. Such application programs can still function without modification as a COMMAREA programs requiring multiple invocations simulating a pseudo-conversation.

- The CONVERSATION-ID field is used to specify a unique id that will be used to uniquely identify the saved Working Storage and Pseudo File objects between invocations of the COMMAREA program participating in a pseudo-conversation. It will also be used to maintain a record in CICS Temporary Storage that identifies the current state of each pseudo-conversation. The digits making up the TCP/IP address of the work station from which the COMMAREA input originates might be a good source for a unique conversation id.

Example 2

This example illustrates a COMMAREA that is derived from a screen definition that includes more than one segment. The presence of multiply occurring segments (rows) in a screen definition corresponds to multiply occurring group fields (arrays) in the COMMAREA.

Example 2 - Advantage VISION:Transact source code

```
        1        2        3        4        5        6        7
----+----0----+----0----+----0----+----0----+----0----+----0----+----0--
IVP6    RC                #   A    GC         IVP6PGM
IVP6    RFCIS2NSTL   FILE                        M
IVP6SCN1SC             OS                     IVP6SCN1
IVP6SCN1SDR _@  IVP6    IVP6MAP    COMMAREA
IVP6SCN1SS 1    1
IVP6SCN1S0 1   NINSTNO  B           M  NC INSTALLATION-NUMBER
IVP6SCN1S1                            NL 5
IVP6SCN1S0 1   NINSTNAMEO             NC INSTALLATION-NAME
IVP6SCN1S1                            NL 35
IVP6SCN1S0 1   NICITY   O             NC CITY
IVP6SCN1S1                            NL 35
IVP6SCN1S0 1   NISTATE  O             NC STATE
IVP6SCN1S1                            NL 20
IVP6SCN1SS 2   20
IVP6SCN1S0 2   NPRODKEY O             NC PRODUCT-KEY
IVP6SCN1S1                            NL 15
IVP6SCN1S0 2   NBOOKDATEO             NC BOOKDATE
IVP6SCN1S1                            NL 6
IVP6SCN1S0 2   NREVCODE O             NC REVENUE-CODE
IVP6SCN1S1                            NL 2
IVP6SCN1S0 2   NPRODCODEO             NC PRODUCT-CODE
IVP6SCN1S1                            NL 6
```

Note that the screen definition above contains two segments; segment 1 occurs one time and segment 2 occurs 20 times. This example also illustrates the use of the new NC operator on the S0 statement and the new usage of the NL operator on the Sn statement. The use of these operators was described more fully in the previous Specification Changes section.

Also, note that the screen definition does not contain any SL statements and the SD statement DEVICE ID entry (column 36) contains the value COMMAREA. This specification indicates that the logical screen does not have inherent limits on the number of rows or number of columns available. Use of the COMMAREA term for DEVICE ID is an intrinsic term meaning that it need not be defined in the M5PARM Device Table.

Example 2 - Generated COMMAREA definition statements

```
       1        2        3        4        5        6        7
----+----0----+----0----+----0----+----0----+----0----+----0----+----0--
      01 DFHCOMMAREA.
     *
         05 IVP6SCN1.
            10 IVP6SCN1-01.
               15 INSTALLATION-NUMBER-ATTR              PIC XXX.
               15 INSTALLATION-NUMBER                   PIC X(05).
               15 INSTALLATION-NAME-ATTR                PIC XXX.
               15 INSTALLATION-NAME                     PIC X(35).
               15 CITY-ATTR                             PIC XXX.
               15 CITY                                  PIC X(20).
               15 STATE-ATTR                            PIC XXX.
               15 STATE                                 PIC X(02).
               15 IVP6SCN1-02 OCCURS 20 TIMES.
                  20 PRODUCT-KEY-ATTR                   PIC XXX.
                  20 PRODUCT-KEY                        PIC X(14).
                  20 BOOKDATE-ATTR                      PIC XXX.
                  20 BOOKDATE                           PIC X(06).
                  20 REVENUE-CODE-ATTR                  PIC XXX.
                  20 REVENUE-CODE                       PIC X(02).
                  20 PRODUCT-CODE-ATTR                  PIC XXX.
                  20 PRODUCT-CODE                       PIC X(06).
            10 MSGLINE-ATTR                            PIC XXX.
            10 MSGLINE                                 PIC X(79).
            10 TRANSACTION-CONTROL REDEFINES MSGLINE.
               15 ATTNID                               PIC X(05).
               15 FILLER                               PIC X.
               15 CONVERSE                             PIC XXX.
               15 FILLER                               PIC X.
               15 CONVERSATION-ID                      PIC X(12).
```

Whenever a screen definition includes more than one segment, the generated COMMAREA definition will include a group field for each segment where the group field name is the screen name followed by the segment number (see IVP6SCN1-01 and IVP6SCN1-02 above).

Example 3

This example illustrates a COMMAREA containing multiple screens sections. The example is taken from the sample code shown in Appendix B of the *Advantage VISION:Transact for CICS Reference Guide* (formerly Concepts and Facilities).

Example 3 - Advantage VISION:Transact source code

```
        1         2         3         4         5         6         7
----+----0----+----0----+----0----+----0----+----0----+----0----+----0--
M501    RC                #    C   GC M14CSIT
M501    RFCUSTOMER   FILE                          U
M501    RFITEMX      FILE1                         U
M501    RFITEMX      FILE2 SYNCFLD1SYNCFLD2        T
CUSTUPD4SC              XR1CUTYPE1
CUSTUPD4SC     FATTNID  EQCENTER
CUSTUPD4SC              OS                        CUSTUPD4
CUSTUPD4SC    EFATTNID  EQCPF006
CUSTUPD4SC              OSOSYNCLOAD               ORDUPDT4
CUSTUPD4SC    EFATTNID  EQCPF007
CUSTUPD4SC              OS                        ORDADD4
CUSTUPD4SC    EFATTNID  EQCPF010
CUSTUPD4SC              OS                        ORDMENU4
ORDUPDT4SC              XR1OUR1N1
ORDUPDT4SC              XR2OUTYPE2
ORDUPDT4SC     FATTNID  EQCPF001
ORDUPDT4SC              OS                        CUSTUPD4
ORDUPDT4SC    EFATTNID  EQCPF010
ORDUPDT4SC              OS                        ORDMENU4
ORDUPDT4SC    EFATTNID  EQCENTER
ORDUPDT4SC              OS                        ORDUPDT4
ORDADD4 SC              XR1VLDORDAD
ORDADD4 SC              XR2OAR2N1
ORDADD4 SC     FATTNID  EQCPF001
ORDADD4 SC              OS                        CUSTUPD4
ORDADD4 SC    EFATTNID  EQCPF006
ORDADD4 SC              OS                        ORDUPDT4
ORDADD4 SC    EFATTNID  EQCPF010
```

```
ORDADD4 SC             OS                    ORDMENU4
ORDADD4 SC    EFATTNID  EQCENTER
ORDADD4 SC             OS                    ORDADD4
CUSTUPD4SDR _@  M501    CUSTSET
CUSTUPD4SL  10  CUSTUPD4
CUSTUPD4SL  30           CUSTOMER NUMBER ____
CUSTUPD4SL  40           CUSTOMER NAME   ______________________________
CUSTUPD4SL  50           CUSTOMER PHONE  _________
CUSTUPD4SL  70        ACT ORDER                P.O.   DUE   INVOICE
CUSTUPD4SL  71  ORDER
CUSTUPD4SL  80        ION NUMBER  DATE  SELLER  NUMBER  DATE  GENERATED
CUSTUPD4SL  81  COMPL
CUSTUPD4SL  90         -  -----  ------  ---    -----  ------     -
CUSTUPD4SL  91    -
CUSTUPD4SL 110         _  _____  ______  ___    _____  ______     _
CUSTUPD4SL 111    _
CUSTUPD4SL 130         _  _____  ______  ___    _____  ______     _
CUSTUPD4SL 131    _
CUSTUPD4SL 150         _  _____  ______  ___    _____  ______     _
CUSTUPD4SL 151    _
CUSTUPD4SL 170         _  _____  ______  ___    _____  ______     _
CUSTUPD4SL 171    _
CUSTUPD4SL 190         _  _____  ______  ___    _____  ______     _
CUSTUPD4SL 191    _
CUSTUPD4SL 210 NEXT SCREEN OF ORDERS=PA1       UPDATE=U ACT + ENTER
CUSTUPD4SL 220 ORDER DETAILS=X ACT + PF6       ADD ORDER=PF7
CUSTUPD4SL 221     RETURN TO MENU=PF10
CUSTUPD4SS 1    1
CUSTUPD4S0 1   NCUSTNO  B           X  NM CUSTNO
CUSTUPD4S1                          M  IFFATTNID  EQCPF006
CUSTUPD4S1                          M  IFFATTNID  EQCPF007
CUSTUPD4S1                          U  IFFATTNID  EQCENTER
CUSTUPD4S0 1   NCUSTNAMEB           P  NM CUSTNAME
CUSTUPD4S0 1   NCUSTPH  B           P  NM CUSTPH
CUSTUPD4SS 2    5
CUSTUPD4S0 2           I            N  NM ACTION
CUSTUPD4S0 2   NORDERNO B           X  NM ORDERNO
CUSTUPD4S1                          U  IFSACTION  EQCU
```

```
CUSTUPD4S1                          M  IFSACTION  EQCX
CUSTUPD4S0 2   NORDRDATEB           P  NM ORDRDATE
CUSTUPD4S0 2   NORPERSONB           P  NM ORPERSON
CUSTUPD4S0 2   NORDPONUMB           P  NM ORDPONUM
CUSTUPD4S0 2   NORDDUDATB           P  NM ORDDUDAT
CUSTUPD4S0 2   NORDINVGNB           P  NM ORDINVGN
CUSTUPD4S0 2   NORDCMPLTB           P  NM ORDCMPLT
ORDMENU4SDR _@  M510    M510
ORDMENU4SL  10  ORDMENU4
ORDMENU4SL  30          CHOOSE SCREEN USING THE FOLLOWING FUNCTION KEYS:
ORDMENU4SL  50                  PF1---CUSTOMER ADD/UPDATE
ORDMENU4SL  70                  PF2---ITEM BROWSE BY PRODUCT KEY
ORDMENU4SL  90                  PF3---BACK-ORDERED ITEM
ORDMENU4SL 110                  PF4---ITEM ADD/UPDATE
ORDMENU4SL 130                  PF5---END TRANSACTION
ORDMENU4SL 160                  ENTER CUSTOMER OR ITEM NUMBER   ______
ORDMENU4SL 180        (ITEM NUMBER MAY BE ONE LETTER PRODUCT CODE IF PF2
ORDMENU4SL 181  IS USED)
ORDMENU4SS 1    1
ORDMENU4S0 1   NCUSTNO  I           X  NM ITEMNO
ORDMENU4S1                          V  IFSPFKEY   EQCPF001
ORDMENU4S0 1  S1ITEMNO  I           X
ORDMENU4S1                          M  IFSPFKEY   EQCPF003
ORDMENU4S1                          V  IFSPFKEY   EQCPF004
ORDMENU4S0 1  V         I           N  NM PFKEY
ORDMENU4S1                             LN 5
ORDMENU4S1                             EQCPF001
ORDMENU4S1                           1OEQCPF002
ORDMENU4S1                           1OEQCPF003
ORDMENU4S1                           1OEQCPF004
ORDMENU4S1                           1OEQCPF005
ORDUPDT4SDR _@
ORDUPDT4SL  10  ORDUPDT4
ORDUPDT4SL  11 UPDATE=U ACT + ENTER
ORDUPDT4SL  20  CUSTOMER NUMBER ____
ORDUPDT4SL  21 CUST UPDT SCRN=PF1
ORDUPDT4SL  30  ORDER NUMBER    ____
ORDUPDT4SL  31 RETURN TO MENU=PF10
```

```
ORDUPDT4SL  40  CUSTOMER NAME   ______________________________
ORDUPDT4SL  51           QTY
ORDUPDT4SL  60  ACT ITEM          ITEM             PRICE    QUANTITY
ORDUPDT4SL  61  BACK      ON
ORDUPDT4SL  70  ION NUMBER        NAME                      ORDERED
ORDUPDT4SL  71 ORDERED   HAND
ORDUPDT4SL  80   -  ------- -------------------- --------- ---------- --
ORDUPDT4SL  81 -------- -------
ORDUPDT4SL  90   _  ______ ________________ ________ _________ __
ORDUPDT4SL  91 _______ ______
ORDUPDT4SL 100   _  ______ ________________ ________ _________ __
ORDUPDT4SL 101 _______ ______
ORDUPDT4SL 110   _  ______ ________________ ________ _________ __
ORDUPDT4SL 111 _______ ______
ORDUPDT4SL 120   _  ______ ________________ ________ _________ __
ORDUPDT4SL 121 _______ ______
ORDUPDT4SL 130   _  ______ ________________ ________ _________ __
ORDUPDT4SL 131 _______ ______
ORDUPDT4SL 140   _  ______ ________________ ________ _________ __
ORDUPDT4SL 141 _______ ______
ORDUPDT4SL 150   _  ______ ________________ ________ _________ __
ORDUPDT4SL 151 _______ ______
ORDUPDT4SL 160   _  ______ ________________ ________ _________ __
ORDUPDT4SL 161 _______ ______
ORDUPDT4SL 170   _  ______ ________________ ________ _________ __
ORDUPDT4SL 171 _______ ______
ORDUPDT4SL 180   _  ______ ________________ ________ _________ __
ORDUPDT4SL 181 _______ ______
ORDUPDT4SL 190   _  ______ ________________ ________ _________ __
ORDUPDT4SL 191 _______ ______
ORDUPDT4SL 200   _  ______ ________________ ________ _________ __
ORDUPDT4SL 201 _______ ______
ORDUPDT4SL 210   _  ______ ________________ ________ _________ __
ORDUPDT4SL 211 _______ ______
ORDUPDT4SL 220   _  ______ ________________ ________ _________ __
ORDUPDT4SL 221 _______ ______
ORDUPDT4SL 230   _  ______ ________________ ________ _________ __
ORDUPDT4SL 231 _______ ______
```

```
ORDUPDT4SS 1    1
ORDUPDT4S0 1   NCUSTNO  B   HY       X  NM CUSTNO
ORDUPDT4S1                           M  IFFATTNID  EQCPF001
ORDUPDT4S1                           M  IFFATTNID  EQCENTER
ORDUPDT4S0 1   NORDERNO B    Y       X  NM ORDERNO
ORDUPDT4S1                           Q  IFFATTNID  EQCENTER
ORDUPDT4S0 1   NCUSTNAMEO               NM CUSTNAME
ORDUPDT4SS 2   15
ORDUPDT4S0 2           I   H         N  NM ACTION
ORDUPDT4S0 2   NITEMORD B            X  NM ITEMORD
ORDUPDT4S1                           V  IFSACTION  EQCU
ORDUPDT4S0 2  S1ITEMNO  I            X
ORDUPDT4S1                           U  IFSACTION  EQCU
ORDUPDT4S0 2   2ITEMNAMEO               NM ITEMNAME
ORDUPDT4S0 2   NITMPRICEB            P  NM ITMPRICE
ORDUPDT4S0 2   NITMQTYORB            P  NM ITMQTYOR
ORDUPDT4S0 2   NQTYBKORDB            P  NM QTYBKORD
ORDUPDT4S0 2   2QANONHNDO               NM QANONHND
ORDADD4 SDR _@
ORDADD4 SL  10  ORDADD4
ORDADD4 SL  11 ADD NEW DATA=ENTER
ORDADD4 SL  21 CUST UPDT SCRN=PF1
ORDADD4 SL  30  CUSTOMER NUMBER ____
ORDADD4 SL  31 VIEW ADDED ITEMS=PF6
ORDADD4 SL  40  CUSTOMER NAME  ______________________________
ORDADD4 SL  41 RETURN TO MENU=PF10
ORDADD4 SL  50  CUSTOMER PHONE  _________
ORDADD4 SL  70  ORDER NUMBER _____ SELLER ___
ORDADD4 SL  80  P.O. NUMBER _____ DUE DATE ______ ORDER DATE _____
ORDADD4 SL 100       ITEM      PRICE    QUANTITY    BACK
ORDADD4 SL 110       NUMBER             ORDERED    ORDERED
ORDADD4 SL 120       ------- --------- ---------- ---------
ORDADD4 SL 130       _______ _________ __________ _________
ORDADD4 SL 140       _______ _________ __________ _________
ORDADD4 SL 150       _______ _________ __________ _________
ORDADD4 SL 160       _______ _________ __________ _________
ORDADD4 SL 170       _______ _________ __________ _________
ORDADD4 SL 180       _______ _________ __________ _________
```

```
ORDADD4 SL 190      _______ ________ _________ _________
ORDADD4 SL 200      _______ ________ _________ _________
ORDADD4 SL 210      _______ ________ _________ _________
ORDADD4 SL 220      _______ ________ _________ _________
ORDADD4 SL 230      _______ ________ _________ _________
ORDADD4 SS 1    1
ORDADD4 S0 1   NCUSTNO  B    Y      X  NM CUSTNO
ORDADD4 S1                         M  IFFATTNID  EQCPF001
ORDADD4 S1                         M  IFFATTNID  EQCPF006
ORDADD4 S1                         M  IFFATTNID  EQCENTER
ORDADD4 S0 1   NCUSTNAMEO             NM CUSTNAME
ORDADD4 S0 1   NCUSTPH  O             NM CUSTPH
ORDADD4 S0 1   NORDERNO B          X  NM ORDERNO
ORDADD4 S1                         M  IFFATTNID  EQCPF006
ORDADD4 S1                         V  IFFATTNID  EQCENTER
ORDADD4 S0 1   NORPERSONB          P  NM ORPERSON
ORDADD4 S0 1   NORDPONUMB          P  NM ORDPONUM
ORDADD4 S0 1   NORDDUDATB          P  NM ORDDUDAT
ORDADD4 S0 1   NORDRDATEB          P  NM ORDRDATE
ORDADD4 SS 2   11
ORDADD4 S0 2   NITEMORD I          X  NM ITEMORD
ORDADD4 S1                         I  IFFATTNID  EQCENTER
ORDADD4 S0 2  S1ITEMNO  I          X
ORDADD4 S1                         U  IFFATTNID  EQCENTER
ORDADD4 S0 2   NITMPRICEI          R  NM ITMPRICE
ORDADD4 S0 2   NITMQTYORI          R  NM ITMQTYOR
ORDADD4 S0 2  S1QANONHNDI          S
ORDADD4 S1                            LE1QANONHND
ORDADD4 S1                            MSCNOT ENOUGH IN STOCK
ORDADD4 S0 2   NQTYBKORDI          R  NM QTYBKORD
VALDCUPDER                                    S             P
##PROC
; THIS PROCEDURE VALIDATES THE FUNCTION KEY PRESSED
;
IF F.ATTNID NE 'ENTER', 'PF006', 'PF007', 'PF010'
  MSG 'INVALID KEY PRESSED - RETRY'
END
##PEND
```

```
FINDX   ER                                          S           P
##PROC
; THIS PROCEDURE MAKES SURE THAT ONLY ONE 'X' IS ENTERED
; WHEN PF6 IS PRESSED ON SCREEN CUSTUPD4
;
IF F.ATTNID EQ 'PF006' AND,
   S.ACTION EQ 'X'
  LET T.COUNTX = T.COUNTX + 1
  IF T.COUNTX > 1
    MSG 'ONLY ONE ''X'' PER PF6'
  END
END
##PEND
ONEXCUPDER                                          S           P
##PROC
; THIS PROCEDURE MAKES SURE THAT AT LEAST ONE 'X'
; WAS ENTERED IF PF6 WAS PRESSED.
;
IF F.ATTNID EQ 'PF006' AND,
   T.COUNTX NE 1
  MSG 'AT LEASET ONE ''X'' PER PF6'
END
##PEND
SYNCLOADER                                          O           P
##PROC
; THIS PROCEDURE SETS UP THE SYNCRONIZATION FIELDS FOR
; THE ITEM BROWSE WHEN APPROPRIATE.
;
IF F.ATTNID EQ 'PF006'
  LET T.SYNCFLD1 = ITEMORD
  LET T.SYNCFLD2 = ITEMORD
END
##PEND
OUR1N1  ER                                          1           P
##PROC
; THIS PROCEDURE VALIDATES THE FUNCTION KEY PRESSED
; BY THE OPERATOR FOR SCREEN ORDUPDT4
;
```

```
IF F.ATTNID EQ 'ENTER', 'PF001', 'PF010'
; IF PF1 OR PF10 HAVE BEEN PRESSED, MAKE SURE ACTION IS BLANK
  IF F.ATTNID NE 'ENTER'
    LET S.ACTION = ' '
  END
ELSE
  MSG 'INVALID KEY PRESSED - RETRY'
END
##PEND
OUR2N1  ER                                          S             P
##PROC
; THIS PROCEDURE MAKES SURE THAT AN ITEM (WHOSE QUANTITY
; ORDERED HAS BEEN CHANGED) HASN'T ALREADY BEEN SHIPPED.
;
IF F.ATTNID EQ 'ENTER' AND S.ACTION EQ 'U'
  ; DOES THIS ITEM ALREADY EXIST ON THIS ORDER
  IF ITEMORD EQ ITEMORD
    ; IF SO, HAS THE QUANITY ORDERED BEEN CHANGED
    IF S.ITMQTYOR EQ ITMQTYOR
      CONTINUE
    ELSE
      ; IF NOT, HAS THE ITEM BEEN SHIPPED?
      IF ITEMORD EQ ITEMSHIP
        LET T.MSG = 'CANNOT CHANGE QUANTITY ORDERED '
        LET PF(T.MSG,32,31) = '- ITEM ALREADY SHIPPED'
        SET S.ITEMORD
        MSG T.MSG
      END
    END
  END
  IF S.ITMPRICE EQ ' '
    LET S.ITMPRICE = 1.ITMPRICE
  END
END
##PEND
OUR2N2  ER                                          S             P
##PROC
; THIS PROCEDURE CONTROLS THE UPDATE OF THE QUANITY ON
```

```
; HAND FIELD IN THE ITEM DATA BASE
;
IF F.ATTNID EQ 'ENTER' AND S.ACTION EQ 'U'
  ; IS ITEM ALREADY ON THIS ORDER?
  IF ITEMORD EQ ITEMORD
    ;IF SO, HAS THE QUANTITY ORDERED BEEN CHANGED
    IF S.ITMQTYOR EQ ITMQTYOR
      CONTINUE
    ELSE
      LET 1.QANONHND = 1.QANONHND + ITMQTYOR
    END
  END
  LET T.QANONHND = 1.QANONHND - S.ITMQTYOR
  IF T.QANONHND LT 0
    IF ITEMORD EQ ITEMORD
      LET 1.QANONHND = 1.QANONHND - ITMQTYOR
    END
    LET 1.QANONHND = T.QANONHND * -1
    LET T.MSG = 'QUANTITY ORDERED IS '
    LET PF(T.MSG,21,7) = T.QANONHND
    LET PF(T.MSG,31,31) = 'MORE THAN QUANTITY ON HAND'
    SET S.ITMQTYOR
    MSG T.MSG
  END
  LET 1.QANONHND = T.QANONHND
END
##PEND
VLDORDADER                                          1              P
##PROC
; THIS PROCEDURE VALIDATES THE FUNCTION KEY PRESSED
; FOR THE ORDADDR SCREEEN
;
IF F.ATTNID NE 'ENTER', 'PF001', 'PF006', 'PF010'
  MSG 'FUNCTION KEY INVALID - RETRY'
END
##PEND
OAR2N1  ER                                          2              P
##PROC
```

```
; IF PRICE HAS NOT BEEN ENTERED ON THIS NEW ITEM, GET
; CURRENT PRICE FROM THE DATA BASE.
;
IF S.ITMPRICE EQ ' '
  LET S.ITMPRICE = 1.ITMPRICE
END
##PEND
OUTYPE2 ER                                    2          P
##PROC
; CALL TYPE2 PROCEDURES FOR ORDUPDT4
;
CALL OUR2N1
CALL OUR2N2
##PEND
CUTYPE1 ER                                    1          P
##PROC
; CALL TYPE 1 PROCEDURES FOR CUSTUPD4
;
CALL VALDCUPD
CALL FINDX
CALL ONEXCUPD
##PEND
TEMPS   ER                                    N          P
##PROC
; TEMP FIELD DEFINITIONS
SYNCFLD1: FIELD C  7
SYNCFLD2: FIELD C  7
SCREENNO: FIELD Z  2
NOSCREEN: FIELD Z  2
MSG:      FIELD C 74
SCRNSLCT: FIELD C  1
UPDATE:   FIELD C  1 INIT 'U'
BACKORD:  FIELD C  1 INIT 'B'
MENU:     FIELD C  1 INIT 'M'
ERROR:    FIELD C  1 INIT 'E'
FIRSTOCC: FIELD P  2
LASTOCC:  FIELD P  2
COUNTX:   FIELD P  2
```

```
QANONHND: FIELD C  7
##PEND
```

Note that this application program references four screens. This application was originally designed as a 3270 Terminal program and may be used as a COMMAREA program if the ATTNID field in the COMMAREA is set to the appropriate PFnnn values. The ATTNID field value controls what section of the COMMAREA is to be used as input data and what section in the COMMAREA is to be used for output during a particular interaction within a pseudo-conversation. The definition below shows how these four screens are combined into the COMMAREA.

Example 3 - Generated COMMAREA definition statements

```
       1        2        3        4        5        6        7
----+----0----+----0----+----0----+----0----+----0----+----0----+----0--
      01 DFHCOMMAREA.
     *
         05 CUSTUPD4.
            10 CUSTUPD4-01.
               15 CUSTNO-ATTR                       PIC XXX.
               15 CUSTNO                            PIC X(05).
               15 CUSTNAME-ATTR                     PIC XXX.
               15 CUSTNAME                          PIC X(30).
               15 CUSTPH-ATTR                       PIC XXX.
               15 CUSTPH                            PIC X(10).
               15 CUSTUPD4-02 OCCURS 05 TIMES.
                  20 ACTION-ATTR                    PIC XXX.
                  20 ACTION                         PIC X(01).
                  20 ORDERNO-ATTR                   PIC XXX.
                  20 ORDERNO                        PIC X(05).
                  20 ORDRDATE-ATTR                  PIC XXX.
                  20 ORDRDATE                       PIC X(06).
                  20 ORPERSON-ATTR                  PIC XXX.
                  20 ORPERSON                       PIC X(03).
                  20 ORDPONUM-ATTR                  PIC XXX.
                  20 ORDPONUM                       PIC X(05).
                  20 ORDDUDAT-ATTR                  PIC XXX.
                  20 ORDDUDAT                       PIC X(06).
                  20 ORDINVGN-ATTR                  PIC XXX.
                  20 ORDINVGN                       PIC X(01).
                  20 ORDCMPLT-ATTR                  PIC XXX.
                  20 ORDCMPLT                       PIC X(01).
               10 MSGLINE-ATTR                      PIC XXX.
               10 MSGLINE                           PIC X(79).
               10 TRANSACTION-CONTROL REDEFINES MSGLINE.
                  15 ATTNID                         PIC X(05).
                  15 FILLER                         PIC X.
                  15 CONVERSE                       PIC XXX.
                  15 FILLER                         PIC X.
                  15 CONVERSATION-ID                PIC X(12).
```

```
*
    05 ORDUPDT4.
       10 ORDUPDT4-01.
          15 CUSTNO-ATTR                    PIC XXX.
          15 CUSTNO                         PIC X(05).
          15 ORDERNO-ATTR                   PIC XXX.
          15 ORDERNO                        PIC X(05).
          15 CUSTNAME-ATTR                  PIC XXX.
          15 CUSTNAME                       PIC X(30).
          15 ORDUPDT4-02 OCCURS 15 TIMES.
             20 ACTION-ATTR                 PIC XXX.
             20 ACTION                      PIC X(01).
             20 ITEMORD-ATTR                PIC XXX.
             20 ITEMORD                     PIC X(07).
             20 ITEMNAME-ATTR               PIC XXX.
             20 ITEMNAME                    PIC X(20).
             20 ITMPRICE-ATTR               PIC XXX.
             20 ITMPRICE                    PIC X(09).
             20 ITMQTYOR-ATTR               PIC XXX.
             20 ITMQTYOR                    PIC X(10).
             20 QTYBKORD-ATTR               PIC XXX.
             20 QTYBKORD                    PIC X(10).
             20 QANONHND-ATTR               PIC XXX.
             20 QANONHND                    PIC X(07).
       10 MSGLINE-ATTR                      PIC XXX.
       10 MSGLINE                           PIC X(79).
*
    05 ORDADD4.
       10 ORDADD4-01.
          15 CUSTNO-ATTR                    PIC XXX.
          15 CUSTNO                         PIC X(05).
          15 CUSTNAME-ATTR                  PIC XXX.
          15 CUSTNAME                       PIC X(30).
          15 CUSTPH-ATTR                    PIC XXX.
          15 CUSTPH                         PIC X(10).
          15 ORDERNO-ATTR                   PIC XXX.
          15 ORDERNO                        PIC X(05).
          15 ORPERSON-ATTR                  PIC XXX.
```

```
           15 ORPERSON                        PIC X(03).
           15 ORDPONUM-ATTR                   PIC XXX.
           15 ORDPONUM                        PIC X(05).
           15 ORDDUDAT-ATTR                   PIC XXX.
           15 ORDDUDAT                        PIC X(06).
           15 ORDRDATE-ATTR                   PIC XXX.
           15 ORDRDATE                        PIC X(06).
           15 ORDADD4-02 OCCURS 11 TIMES.
              20 ITEMORD-ATTR                 PIC XXX.
              20 ITEMORD                      PIC X(07).
              20 ITMPRICE-ATTR                PIC XXX.
              20 ITMPRICE                     PIC X(09).
              20 ITMQTYOR-ATTR                PIC XXX.
              20 ITMQTYOR                     PIC X(10).
              20 QTYBKORD-ATTR                PIC XXX.
              20 QTYBKORD                     PIC X(10).
        10 MSGLINE-ATTR                       PIC XXX.
        10 MSGLINE                            PIC X(79).
*
     05 ORDMENU4.
        10 ITEMNO-ATTR                        PIC XXX.
        10 ITEMNO                             PIC X(07).
        10 PFKEY-ATTR                         PIC XXX.
        10 PFKEY                              PIC X(05).
        10 MSGLINE-ATTR                       PIC XXX.
        10 MSGLINE                            PIC X(79).
```

Program Debugging

The execution tracing facilities provided by VISION:Transact to assist in application program debugging continue to be available for COMMAREA application programs. The only restriction for COMMAREA application programs is that the 3270 Terminal may not be used as a destination for trace messages. The CICS TS Temporary Storage queue and the CICS TS Transient Data queue destinations are available to both COMMAREA application programs and 3270 Terminal application programs.

The Temporary Storage queue name for trace messages when an application program is running as a 3270 Terminal program is M5ETxxxx where xxxx is the CICS TS Terminal ID. The Temporary Storage queue name for trace messages when an application is running as a COMMAREA program is M5ETyyyyyyyyyyyy where yyyyyyyyyyyy is the conversation id provided in the CONVERSATION-ID field of the input COMMAREA. If this field contains blanks, then the name is simply M5ET. Thus, when tracing is enabled for multiple COMMAREA application programs, it would be prudent to assign a unique CONVERSATION-ID value even though the application itself does not support a pseudo conversation so that application program trace messages can be routed to various queues.

The Transient Data queue name for trace messages is always M5ET regardless of whether the application program is running as a 3270 Terminal program or a COMMAREA program. If the Transient Data queue for M5ET is not defined to CICS TS or the destination does not initialize properly, trace messages to this destination are suppressed.