

CA-IDMS[®]

Visual DBA

User Guide

2.0



Computer Associates™

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

© 2001 Computer Associates International, Inc., One Computer Associates Plaza, Islandia, New York 11749. All rights reserved.

All trademarks, trade names, service marks, or logos referenced herein belong to their respective companies.

Contents

Chapter 1: Welcome to CA-IDMS Visual DBA!

CA-IDMS Visual DBA Features	1-1
CA-IDMS Objects You Can Manage	1-2
Manipulating CA-IDMS Objects	1-6
Viewing and Maintaining Multiple CA-IDMS Systems	1-7
Managing Objects Using Drag-and-Drop	1-7
Increasing Productivity	1-7
Command Consoles	1-7
Customizing the User Interface	1-8
What You Need to Know	1-8
Getting Help	1-8
Related Documentation	1-9

Chapter 2: Installing and Starting CA-IDMS Visual DBA

Software Requirements	2-1
Installing CA-IDMS Visual DBA on the PC	2-1
Installing CA-IDMS Visual DBA on the Mainframe	2-2
Starting Your DBA Session	2-6
Starting CA-IDMS Visual DBA	2-6
The Application Window	2-6
The Workspace Toolbar	2-7
The Nodes Window Toolbar	2-8
Connecting to a Datasource/Node	2-9
Adding a New Datasource/Node	2-10
Opening a Database Object Manager Window	2-11
Securing CA-IDMS Visual DBA Objects	2-12
Restricting Access to Objects	2-13
Assigning Privileges to View Object Instances	2-14
Assigning Privileges to Update Object Instances	2-14
Assigning Privileges to Create, Alter, and Drop Object Instances	2-15

Assigning Privileges to Grant and Revoke Object Privileges	2-15
What's Next	2-15

Chapter 3: A Tour of the Database Object Manager Window

The DOM Window	3-1
Using the DOM Window Toolbar	3-3
Expanding and Collapsing the Object Tree	3-4
Expanding the Tree	3-4
Collapsing the Tree	3-5
Changing the Tree Structure	3-6
Restart from Position	3-6
Tear-Out Window	3-7
Scratchpad	3-9
Manipulating Tree Objects	3-10
Finding Objects	3-12
Selecting and Copying Objects	3-12
Viewing, Executing, and Logging Object Syntax	3-13
Refreshing the Tree	3-13
Connecting to Multiple CA-IDMS Systems	3-14
Displaying Object Attributes in the Detail Information Pane	3-16
Tab Examples	3-17
Setting Display Options for Your Session	3-29
Creating, Opening, and Saving Environments	3-30
Trying It Out Yourself	3-30
What's Next	3-33

Chapter 4: Using the Command Console

Using the Command Console Workspace	4-2
Command Console Toolbar	4-2
Command Console Editor	4-3
Command Console Results Pane	4-4
Running a Command Script	4-5
Using the SQL Assistant	4-7
Setting Command Console Preferences	4-11

Appendix A: Security TABL Resource Tables

Overview A-1

Table 1: Objectnames in Tree Order A-2

Table 2: Objectnames in Alphabetical Order A-13

Table 3: Objectnames and Schema Table in Alphabetical Order A-25

Table 4: Schema Table in Alphabetical Order A-37

Appendix B: Uninstalling and Reinstalling

Uninstalling CA-IDMS Visual DBA on the PC B-1

 Uninstalling CA-IDMS Visual DBA 1.0 B-1

 Uninstalling CA-IDMS Visual DBA 2.0 B-1

Uninstalling the mainframe CA-IDMS Visual DBA SQL Definitions B-1

 Uninstalling CA-IDMS Visual DBA 1.0 B-1

 Uninstalling CA-IDMS Visual DBA 2.0 B-2

Reinstalling CA-IDMS Visual DBA 2.0 B-2

Appendix C: Using the Demo20.cfg File

Using the Demo20.cfg File C-1

Welcome to CA-IDMS Visual DBA!



CA-IDMS Visual DBA is a robust, comprehensive tool that enables you to manage CA-IDMS database objects from a single, object-oriented graphical user interface (GUI). It also allows the database administrator (DBA) to perform essential, everyday tasks quickly and easily – using an easy-to-use, intuitive, graphical user interface under Microsoft Windows.

With CA-IDMS Visual DBA, you no longer need to be concerned with the details of how to administer your database, such as learning the syntax for multiple online compilers or DCMT commands. CA-IDMS Visual DBA simplifies the operations that you perform routinely, enabling you to increase your productivity and decrease the learning curve typically associated with database administration.

CA-IDMS Visual DBA Features

CA-IDMS Visual DBA provides you with the following features, which enable you to:

- Display all CA-IDMS objects in a hierarchical tree. You can create, alter, drop, and display objects and assign privileges for each object in the tree
- Connect to multiple CA-IDMS systems during one CA-IDMS Visual DBA session
- Increase productivity and decrease your learning curve because you no longer need to know the syntax of multiple online compilers or DCMT commands
- Customize the tree, change fonts, set automatic refresh parameters and save the current environment to a file for later use
- Use multiple command consoles for SQL, for the CA-IDMS compilers and for DCMT and DCUF
- Visualize the properties of CA-IDMS objects in the Detail Information pane in many ways, using selectable tabs to maximize efficiency in comparing and evaluating them.

Note: Pie charts for monitoring program and storage pool use are available, just as bar diagrams for statistical information

The user interface for CA-IDMS Visual DBA uses many Windows features that should already be familiar to you. These include:

- Drill-down icons
- Popup menus to manipulate objects
- Tear-out windows to customize your view of the hierarchical object tree display
- The ability to drag-and-drop and cut-and-paste objects

CA-IDMS Objects You Can Manage

CA-IDMS Visual DBA manipulates a large number of CA-IDMS objects that are managed on the mainframe using

- the schema, subschema, DDDL (IDD), and system generation compilers
- the Online Command Facility (OCF)
- the Batch Command Facility (IDMSBCF)
- DCMT and DCUF system tasks.

The following list summarizes up to level three the objects you can create, alter, delete, and display using CA-IDMS Visual DBA:

```
.Dictionary
..Dictionary / SQL Schema
...Dictionary / SQL Schema / Table in Schema
...Dictionary / SQL Schema / View in Schema
...Dictionary / SQL Schema / Table Procedure in Schema
..Dictionary / SQL Schema / Procedure in Schema
...Dictionary / SQL Schema / Non SQL Table in Schema
...Dictionary / SQL Schema / Constraint
...Dictionary / SQL Schema / Access Module
...Dictionary / SQL Schema / Grantee for Schema

..Dictionary / Table Like
...Dictionary / Table Like / Table
...Dictionary / Table Like / View
...Dictionary / Table Like / Table Procedure
...Dictionary / Table Like / Procedure
...Dictionary / Table Like / Non SQL Table

..Dictionary / Relational Command Module

..Dictionary / Non SQL Schema
...Dictionary / Non SQL Schema / Area in Non SQL Schema
...Dictionary / Non SQL Schema / Record
...Dictionary / Non SQL Schema / Set
...Dictionary / Non SQL Schema / Subschema
...Dictionary / Non SQL Schema / Class / Attribute
...Dictionary / Non SQL Schema / Comment
```

```
...Dictionary / Non SQL Schema / Grantee for Use
...Dictionary / Non SQL Schema / Registree for Schema
...Dictionary / Non SQL Schema / Responsible for Schema

..Dictionary / IDD Class
...Dictionary / IDD Class / Attribute

..Dictionary / IDD Record
...Dictionary / IDD Record / IDD Record Synonym
...Dictionary / IDD Record / IDD Record Element

..Dictionary / Segment
...Dictionary / Segment / File
...Dictionary / Segment / Area
...Dictionary / Segment / DMCL including Segment
...Dictionary / Segment / DBName including Segment
...Dictionary / Segment / Schema referencing Segment
...Dictionary / Segment / Grantee for Segment

..Dictionary / DBTable
...Dictionary / DBTable / Subschema Mapping
...Dictionary / DBTable / DBName
...Dictionary / DBTable / DMCL using DBTable
...Dictionary / DBTable / DBGroup in DBTable
...Dictionary / DBTable / Grantee for DBTable

..Dictionary / DMCL
...Dictionary / DMCL / Segment in DMCL
...Dictionary / DMCL / Segment.File Override
...Dictionary / DMCL / Segment.Area Override
...Dictionary / DMCL / File in DMCL
...Dictionary / DMCL / Area in DMCL
...Dictionary / DMCL / Shared File in DMCL
...Dictionary / DMCL / Shared Area in DMCL
...Dictionary / DMCL / Shared Cache in DMCL
...Dictionary / DMCL / Buffer
...Dictionary / DMCL / Journal
...Dictionary / DMCL / Grantee for DMCL

..Dictionary / System
...Dictionary / System / Load List
...Dictionary / System / Program
...Dictionary / System / Queue
...Dictionary / System / Autotask
...Dictionary / System / Task
...Dictionary / System / Storage Pool
...Dictionary / System / XA Storage Pool
...Dictionary / System / Line
...Dictionary / System / Lterm
...Dictionary / System / Printer
...Dictionary / System / Destination
...Dictionary / System / Node
...Dictionary / System / Resource Table
...Dictionary / System / Rununit
...Dictionary / System / Map Table
...Dictionary / System / Key Table

..Dictionary / System Resource & Profile
...Dictionary / System Resource & Profile / Category
...Dictionary / System Resource & Profile / Activity
...Dictionary / System Resource & Profile / System Id
...Dictionary / System Resource & Profile / System Profile
...Dictionary / System Resource & Profile / User Profile

..Dictionary / Group
...Dictionary / Group / User in Group
```

```
...Dictionary / Group / Privilege for Group
...Dictionary / Group / Grantee on Group

..Dictionary / Central User
...Dictionary / Central User / Group User belongs
...Dictionary / Central User / Privilege for User
...Dictionary / Central User / Grantee on User

..Dictionary / Dictionary User
...Dictionary / Dictionary User / Entity Type Authority
...Dictionary / Dictionary User / Registration for Dictionary User
...Dictionary / Dictionary User / Responsibility of Dictionary User

..Dictionary / Grantee for Administration
...Dictionary / Grantee for Administration / Grantee for DCADMIN
...Dictionary / Grantee for Administration / Grantee for SYSADMIN

.CV DBTable
..CV DBTable / CV Subschema Mapping

..CV DBTable / CV DBName
...CV DBTable / CV DBName / CV Segment in DBName
...CV DBTable / CV DBName / CV Subschema Mapping in DBName

..CV DBTable / CV DBGroup
...CV DBTable / CV DBGroup / CV Backend in DBGroup

.CV DMCL
..CV DMCL / CV Segment
...CV DMCL / CV Segment / CV File in Segment
...CV DMCL / CV Segment / CV Area in Segment
...CV DMCL / CV Segment / CV DBName including Segment

..CV DMCL / CV File

..CV DMCL / CV Area

..CV DMCL / CV Database Buffer

..CV DMCL / CV Journal Buffer

..CV DMCL / CV Journal

.CV System
..CV System / CV Load List

..CV System / CV Program

..CV System / CV Program Pool new Sysgen

..CV System / CV Reent Prog Pool new Sysgen

..CV System / CV Queue

..CV System / CV Task

..CV System / CV Storage Pool

..CV System / CV XA Storage Pool

..CV System / CV XA Storage Pool new Sysgen

..CV System / CV Line
...CV System / CV Line / CV PTerm in Line
...CV System / CV Line / CV LTerm in Line
```

- ..CV System / CV Line new Sysgen
- ..CV System / CV Pterm
- ..CV System / CV Lterm
- ..CV System / CV Printer
- ..CV System / CV Destination
- ..CV System / CV Node
- ..CV System / CV Resource Table
- ..CV System / CV Rununit
- ...CV System / CV Rununit / CV Rununit Detail

- .Activity
- ..Activity / Active Program
- ...Activity / Active Program / Active Program Details

- ..Activity / Active Storage
- ...Activity / Active Storage / Active Storage Details

- ..Activity / Active Task
- ...Activity / Active Task / Active Task Details

- ..Activity / User signed on

- .Central Version
- ..Central Version / Data Sharing Summary

- ..Central Version / Data Sharing Group Member

- ..Central Version / Data Sharing LIST Structure

- ..Central Version / Data Sharing LOCK Structure

- ..Central Version / Deadlock

- ..Central Version / DCMT Command Outstanding

- ..Central Version / External Rununit

- ..Central Version / Journal Status

- ..Central Version / Lock on Area

- ..Central Version / Lock on Lterm

- ..Central Version / Log

- ..Central Version / Log Driver

- ..Central Version / Shared Cache
- ...Central Version / Shared Cache / File in Shared Cache

- ..Central Version / Transaction

- .DC
- ..DC / ADSO

- ..DC / DDS

- ..DC / Limit

```
..DC / Loadlib
..DC / LU
...DC / LU / Modeent of LU
..DC / MPMode
..DC / MT Queue Depth
..DC / Nucleus Reload
..DC / Reply pending
..DC / Report Class / Destination
...DC / Report Class / Destination / Report
..DC / SNA Pterm
..DC / Snap
..DC / SubTasks
..DC / Time
...DC / Time / Time initiated Task
..DC / UCF Terminal
.Statistics
..Statistics / Statistics Segment
...Statistics / Statistics Segment / Statistics File in Segment
...Statistics / Statistics Segment / Statistics Area in Segment
..Statistics / Statistics File
..Statistics / Statistics Area
..Statistics / Statistics Buffer
..Statistics / Statistics Interval
..Statistics / Statistics Lock
..Statistics / Statistics System
```

Manipulating CA-IDMS Objects

CA-IDMS Visual DBA presents your database objects using an *object-oriented* approach. It perceives the world as a collection of objects that interact with each other. For most database object types, you can select it and right-click to display a popup menu that enables you to create, alter, drop the object, and assign privileges to it.

Viewing and Maintaining Multiple CA-IDMS Systems

CA-IDMS Visual DBA is designed to make you a more productive and responsive database administrator. One of the ways that you can accomplish this is by connecting to multiple CA-IDMS systems simultaneously and customizing the view of data for each – in its own *Database Object Manager* window.

Another major benefit that CA-IDMS Visual DBA provides is its ability to save a configured environment. All open windows in your workspace and the data within them, your connections, and certain environment settings – are saved in a configuration file enabling you to immediately begin your activities after starting CA-IDMS Visual DBA.

Managing Objects Using Drag-and-Drop

In addition to presenting the database environment in a graphical format, CA-IDMS Visual DBA provides the ability to copy object definitions from one sub-branch or from the Detail Information pane to any other sub-branch that has objects of the same type. This includes copying definitions from one CA-IDMS central version or system to another.

You simply select an object or group of objects in the same sub-branch or in the Detail Information pane, and then drag and drop it to the target database. CA-IDMS Visual DBA does the appropriate copy and verifies that any integrities involved in the entities are properly maintained.

Using drag-and-drop techniques, greatly simplifies database administration tasks, especially for test and production database maintenance.

Increasing Productivity

With CA-IDMS Visual DBA, you can manage your CA-IDMS systems simply by pointing and clicking. This saves precious time, because you no longer need to remember lengthy command syntax and switch settings.

Command Consoles

This tool lets you directly edit and submit command syntax for the CA-IDMS compilers and DCMT / DCUF. A history of commands and results is automatically kept. Commands can also be saved and recalled as command scripts.

For SQL commands, an SQL Assistant is available to help you create queries.

Customizing the User Interface

Like other Windows products, CA-IDMS Visual DBA provides many customization features that allow you to tailor the environment to suit your needs. Not only can you configure such options as the fonts used in various windows, refresh settings, the status bar display, and others, but more importantly, you can control how your data is displayed.

For example, you can define your own custom view of a dictionary or catalog. You can reorient the view to see the system from the perspective of any other object. This feature allows you to easily access the information you need to view and maintain.

What You Need to Know

This guide assumes that you are already familiar with the concepts and features associated with your CA-IDMS installation. If you are unfamiliar with these concepts, please refer to your CA-IDMS documentation for more information.

In addition to an understanding of CA-IDMS concepts and features, this guide assumes that you are familiar with Windows terminology and navigational techniques. This includes how to work with standard Windows items like menus, dialog boxes, the Clipboard, and the Control Panel.

If you are unfamiliar with Windows standards, please refer to your Windows documentation before using CA-IDMS Visual DBA.

Getting Help



CA-IDMS Visual DBA provides online help, which can be used to display information on your console as you work. From the CA-IDMS Visual DBA workspace environment, you can also press the F1 key for context-sensitive help on menu commands, dialogs, and active windows.

Related Documentation

The entire CA-IDMS document set is viewable on the CA-IDMS Documentation CD. The following documents, in addition to this one and the online help, may be the most useful to you when you use this product:

- *IDD DDDL Reference* for information about class and attribute definitions and IDD record definitions
- *Database Administration* for information about defining DMCLs, schemas, and subschemas
- *SQL Reference* for information about defining SQL schemas, tables, and views
- *System Generation* for information about defining the components of a CA-IDMS system
- *Security Administration* for information about assigning privileges to users
- System tasks and operator commands for information on the runtime objects

Installing and Starting CA-IDMS Visual DBA

Software Requirements

CA-IDMS Visual DBA runs on any mainframe platform that supports CA-IDMS Server.

To run CA-IDMS Visual DBA, you need on the mainframe:

- CA-IDMS Release 12.01 genlevel 9506 or later
- CA-IDMS SQL Option
- CA-IDMS Server Release 2.0 or above

To run CA-IDMS Visual DBA on a PC, you need:

- Microsoft Windows 95/98/ME/NT 4.0/2000/XP
- CA-IDMS Server Version 4.1 or later

Installing CA-IDMS Visual DBA on the PC

To install CA-IDMS Visual DBA on the PC, place the installation CD in your CD drive and the CA-IDMS Visual DBA setup program will start automatically. Alternatively, run **setup.exe** from the installation CD. The setup program guides you through the installation process.

CA-IDMS Visual DBA 2.0 offers three installation options:

1. Typical

This option copies all of the product components on the installation CD into the CA-IDMS Visual DBA 2.0 product folder. This includes the program files and help files needed for execution on the PC; the mainframe files that must be uploaded to the mainframe to install the mainframe component of the product; the example file (demo20.cfg); the CA-IDMS Visual DBA 2.0 User Guide, and the readme file readme.txt, which is also provided in HTML format for viewing through your browser (readme.htm). The mainframe files are copied into the Mainfram subfolder.

2. Compact

This option copies the program files and the help files for the PC into the CA-IDMS Visual DBA 2.0 product folder. It also copies the mainframe installation files into the Mainfram subfolder. It does not copy the example file or the CA-IDMS Visual DBA 2.0 User Guide to the product folder.

3. Custom

This option allows you to select the product components that are copied into the CA-IDMS Visual DBA 2.0 product folder. The product components that may be selected are: program files, example files, help files, mainframe and documentation.

CA-IDMS Visual DBA 2.0 coexists with CA-IDMS Visual DBA 1.0. There is no need to uninstall version 1.0 prior to installing version 2.0.

Please check Appendix B for instructions on uninstalling prior versions and on reinstallation.

Installing CA-IDMS Visual DBA 2.0 on the Mainframe

The CA-IDMS Visual DBA 2.0 setup program on the PC copies the files needed for installation on the mainframe into the Mainfram folder of the CA-IDMS Visual DBA 2.0 product folder on the PC. If you wish to copy only the mainframe files from the installation CD onto the PC, run the PC setup program and select CUSTOM install.

Prior to starting the mainframe installation, check the readme file (either readme.txt or readme.htm) on the installation CD for any apars that may be required for your specific mainframe environment.

Note: Step One may be omitted for installation of CA-IDMS Visual DBA 2.0 on CA-IDMS 15.0 SP0 and higher. Step Two may be omitted for installation of CA-IDMS Visual DBA 2.0 on CA-IDMS 15.0 SP1 and higher.

Step One: Install the IDMSQNCO, IDMSVIDD and IDMSVDCT table procedure load modules

OS/390, z/OS and VSE/ESA procedure

Upload the idmsqnco.object, idmsvidd.object, and idmsvdct.object files from the Mainfram folder to the mainframe using a BINARY file transfer protocol.

- For OS/390 and z/OS, place these members in an object library of your choice.
- For VSE/ESA, create an IDMSQNCO.OBJ member, an IDMSVIDD.OBJ member, and an IDMSVDCT.OBJ member in a LIBR sublibrary of your choice.

Upload the linkedit control statements from the Mainfram folder to the mainframe using a TEXT file transfer protocol.

- For OS/390 and z/OS, upload the idmsvdba.linksmvsv file. Then use the contents of this file to link edit the IDMSQNCO, IDMSVIDD, and IDMSVDCT modules into the library of your choice (these modules are reentrant). This library must be added to your CA-IDMS library concatenation.
- For VSE/ESA, upload the the idmsvdba.linksvse linkedit control statements. Then use the contents of this file to link edit the IDMSQNCO, IDMSVIDD, and IDMSVDCT modules into a sublibrary of your choice (these modules are reentrant). This sublibrary must be added to your CA-IDMS library.

BS2000/OSD procedure

Upload the file idmsvdba.loadlib or idmsvdba.loadlibNK4 for NK4 disks from the Mainfram folder to the mainframe using a BINARY file protocol. Add this load library to your CA-IDMS system or copy the modules IDMSQNCO, IDMSVIDD, and IDMSVDCT to an existing library. Use type R on CA-IDMS 12.0 and 14.x, use type L on CA-IDMS 15.0.

Step Two: Add the IDMSQNCO, IDMSVIDD and IDMSVDCT table procedures to the sysgen

Add the following PROGRAM statements for IDMSQNCO, IDMSVIDD and IDMSVDCT to your CA-IDMS sysgen:

```
ADD PROGRAM IDMSQNCO LANGUAGE ASSEMBLER.
```

```
ADD PROGRAM IDMSVIDD LANGUAGE ASSEMBLER.
```

```
ADD PROGRAM IDMSVDCT LANGUAGE ASSEMBLER.
```

VALIDATE and GENERATE your system.

Step Three: Install the CA-IDMS Visual DBA 2.0 SQL definitions

CA-IDMS Visual DBA uses SQL to access all of the CA-IDMS mainframe objects that it manages. This step executes the SQL DDL statements to add the SQL schemas, views, and table procedures definitions required by CA-IDMS Visual DBA 2.0. The CA-IDMS Visual DBA definitions must be installed in the SQL catalog for each dictionary that is to be managed by CA-IDMS Visual DBA.

Note: Each dictionary to be controlled **must** have a catalog associated with that dictionary.

Note: You may need to increase the catalog size to accommodate the added views and table procedures. This may be especially true for the CATSYS segment files if you used the original installation defaults. You should have at least 1000 pages in your DDLCAT area for all catalogs that will be used with CA-IDMS Visual DBA.

CA-IDMS Visual DBA 2.0 SQL definitions coexist with the CA-IDMS Visual DBA 1.0 SQL definitions. There is no need to uninstall version 1.0 definitions prior to installing version 2.0 definitions. Please check Appendix B for instructions on uninstalling prior versions and reinstallation.

There are two alternatives for installing the CA-IDMS Visual DBA SQL definitions:

- Install using the mainframe command facility IDMSBCF
- Install using CA-IDMS Visual DBA itself

Installing CA-IDMS Visual DBA SQL Definitions Using IDMSBCF

This option submits the SQL DDL statements required by CA-IDMS Visual DBA 2.0 directly to CA-IDMS for processing:

- Locate the SQL script with definitions needed by CA-IDMS Visual DBA for your environment in the Mainfram folder. This script is CA-IDMS release-dependent. The name of this script is:
 - VDB2R12X.SQL for CA-IDMS Rel 12.0.
 - VDB2R14X.SQL for CA-IDMS Rel 14.x.
 - VDB2R15X.SQL for CA-IDMS Rel 15.x .
- Edit the SQL script of your choice to define the names of the dictionaries that must be accessible by CA-IDMS Visual DBA. Directions for editing are embedded in the script file as comments.

- Next, upload the file to the mainframe using a TEXT file transfer protocol. Then execute the Batch Command Facility (IDMSBCF), using the uploaded file as input, against all dictionaries that will be visible from CA-IDMS Visual DBA.

Installing CA-IDMS Visual DBA SQL Definitions Using Visual DBA

This option submits the required SQL DDL statements as ODBC requests from CA-IDMS Visual DBA 2.0 using CA-IDMS Server for processing:

- Run CA-IDMS Visual DBA 2.0 on your PC. The initial window contains a Node window in the left pane. Expanding the root node, *Node*, displays all the available ODBC data sources using a CA-IDMS ODBC driver.
- Select the datasource for the target dictionary for the install. If no appropriate node (ODBC data source) to your mainframe CV/Dictionary exists, perform the following steps:
 - Add a node using the Add function of the Node menu or use the Add button on the toolbar of the node window. (See **Adding a New Datasource/Node** later in this chapter for more details.)
 - Define the ODBC data source as a system or user ODBC data source. The Dictionary to be specified in the ODBC Datasource definition is the target dictionary for the installation.
 - Open a Command Console to your Central Version (CV) by selecting the Cmd Console function of the Node menu or by clicking the Cmd Console button in the toolbar of the node window. You will be prompted for a password to connect with the CA-IDMS CV. (See the **Using the Command Console** chapter later in this guide for more details.)

After a successful connection the command console appears. From the command console, perform the following steps:

- Select the default dictionary of the node or ODBC Datasource in the first unlabeled list box of the command console window. The second unlabeled list box contains the default processor "SQL". Keep this default processor.
- Use the Open Script function of the Script menu to select the installation SQL script that corresponds to your CA-IDMS release. These script files are located in the Mainfram folder of the CA-IDMS Visual DBA product folder:
 - VDB2R12X.SQL for CA-IDMS Rel 12.0.
 - VDB2R14X.SQL for CA-IDMS Rel 14.x.
 - VDB2R15X.SQL for CA-IDMS Rel 15.x.
- Edit the SQL script in the command console window, as explained in the comments, to define the names of the dictionaries that must be accessible by CA-IDMS Visual DBA.

- Click the Trace Tab to see the execution of the script progress.
- Click the GO button or press F5.

The installation is successful if the script terminates without errors; warnings are usually acceptable.

Starting Your DBA Session

Once Windows and CA-IDMS are started, you can begin your work session by starting CA-IDMS Visual DBA and connecting to a CA-IDMS system.

Starting CA-IDMS Visual DBA

Start CA-IDMS Visual DBA by selecting CA-IDMS Visual DBA 2.0 through the Windows Start menu.

To start the CA-IDMS Visual DBA application, follow these steps:

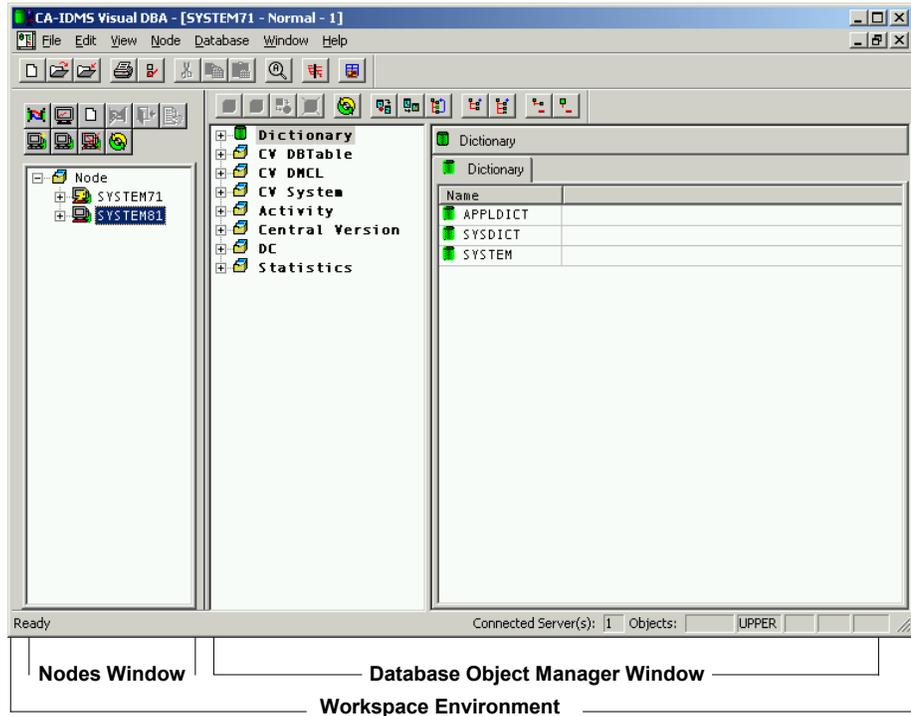
1. From the Start menu, choose Programs.
2. Choose the CA-IDMS Visual DBA 2.0 submenu.
3. Click CA-IDMS Visual DBA 2.0.

The Application Window

CA-IDMS Visual DBA appears in its own application window. This main window is referred to as the *workspace environment* and is the primary work area for your tasks. Initially, it has two sub-windows:

- The **Nodes window**, located on the left is used to display and manage the nodes or ODBC data sources. It is a moveable window.
- The right sub-window initially appears empty. However, once you connect to a CA-IDMS system, the **Database Object Manager (DOM) window** appears with dictionary and runtime objects at the root level.

The Workspace Environment window includes a menu bar and several toolbars. These are documented in detail in online help. The Workspace Toolbar and the Nodes Window Toolbar are described in brief below.



The Workspace Toolbar

As described below, the Workspace Toolbar enables you to perform the following functions:



Button	Description
	Creates a new workspace environment. If the current environment has never been saved or has been changed during the session, Visual DBA prompts you to save it. When you end your Visual DBA session, the current workspace configuration is closed – closing all windows, terminating connections, and setting default system settings.
	Opens a workspace that was previously saved. If a password was previously specified for the workspace, a dialog box appears asking you to provide the password. When the operation is completed, the workspace environment displays in the CA-IDMS Visual DBA window.

Button	Description
	<p>Saves the current workspace environment.</p> <p>The following information is saved:</p> <ul style="list-style-type: none"> ■ Node connections ■ All open windows (including size and placement), and the data displayed in each window ■ Date the file was saved, which is used when opening the saved configuration to determine whether the data in the windows should be refreshed ■ Database information already loaded previously
	Prints the information in the active window.
	<p>Establishes your preferences.</p> <p>Modifies system-wide parameters, such as fonts, session preferences, printer setup options, refresh settings, OK action and command console options.</p>
	Removes text from its current location in the Command console, and copies it to the Clipboard.
	Copies a database object from its current location in the Database Object Manager window to the Clipboard or copies text from the Command console to the Clipboard, where you can retrieve the object or text and insert it elsewhere using the Paste command.
	Pastes the current contents of the Clipboard to a location in the Database Object Manager tree or pastes commands from the Clipboard to another location.
	Finds and <i>jumps to</i> an object or object category in the currently selected Database Object Manager window.
	Arranges all open windows so that they are side-by-side with no overlap.
	Lets you browse the error history log.

The Nodes Window Toolbar

As described below, the Nodes Window Toolbar enables you to perform the following functions:



Button	Description
	Connects to the selected node (CA-IDMS Datasource) and creates a DOM window
	Opens a command console on the selected node.
	Creates a DOM scratchpad window on the selected node (CA-IDMS Datasource).
	Disconnects from the selected node. All opened windows on the selected node will be closed.
	Closes the window selected from the Open Windows branch.
	Makes the window selected from the Open Windows branch the active window.
	Adds a node (CA-IDMS Datasource) to the Node list.
	Alters a selected node (CA-IDMS Datasource) from the Node list.
	Permanently removes the selected node from the Node list.
	Refreshes the Node list.

Connecting to a Datasource/Node

After you start CA-IDMS Visual DBA, connect to a CA-IDMS system to begin your DBA tasks. There are multiple ways of accomplishing the following tasks.

To accomplish this:

1. Select a node or CA-IDMS ODBC Datasource from the expanded nodes window.

2. Select the Connect DOM function from the Node menu or click the Connect DOM button  on the Nodes window toolbar, or right-click in the Nodes window to display the Node Menu.

Note: If no appropriate node is available, one must be added.

Adding a New Datasource/Node

To add a new Datasource or node:

1. Select the Add function from the Node window or click the Add button in the Nodes window toolbar , or right-click in the nodes window to use the Node menu.

CA-IDMS Visual DBA prompts you to determine if the new data source must be system or user.

2. Respond to the prompt.

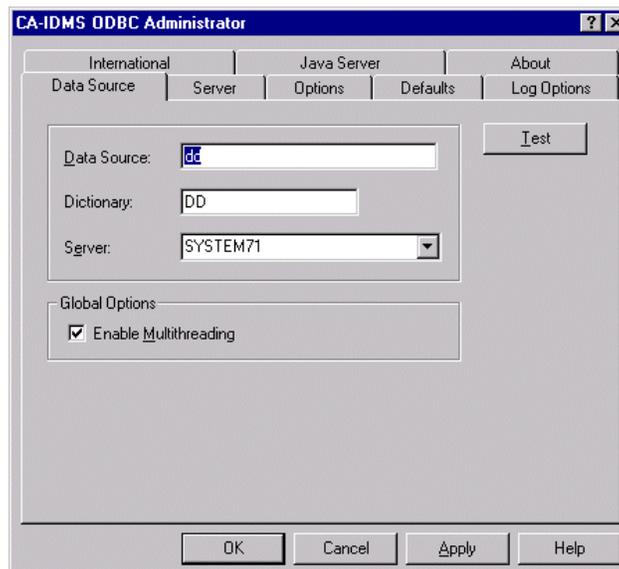
The CA-IDMS ODBC Administrator is invoked.

3. Provide a name for the new data source.

For the dictionary name, enter the name of the dictionary in which you installed the mainframe component of CA-IDMS Visual DBA.

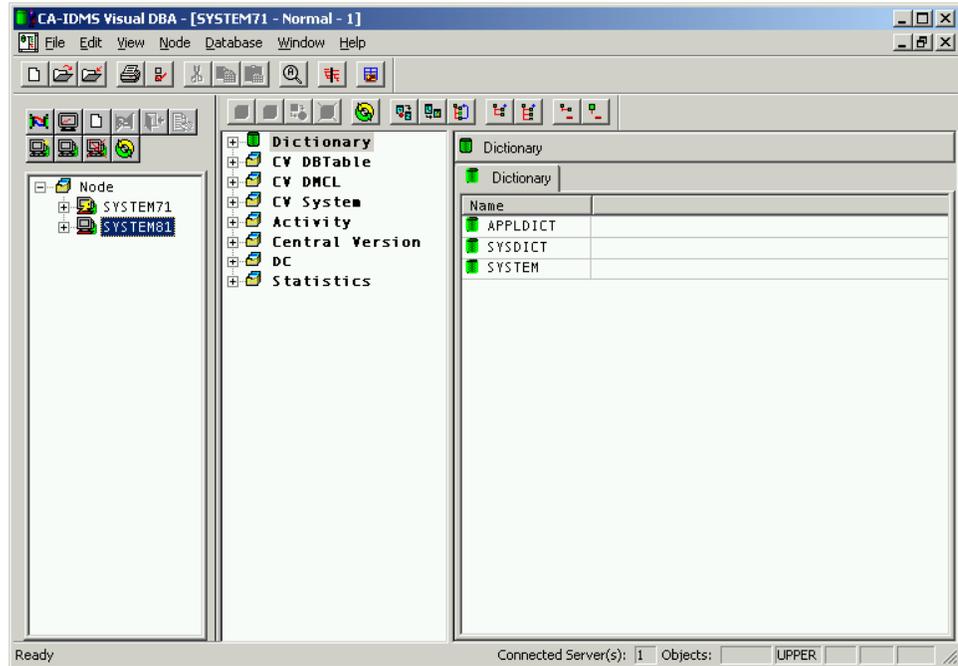
4. From the server list box, select a CA-IDMS system.

The following is an example:



Opening a Database Object Manager Window

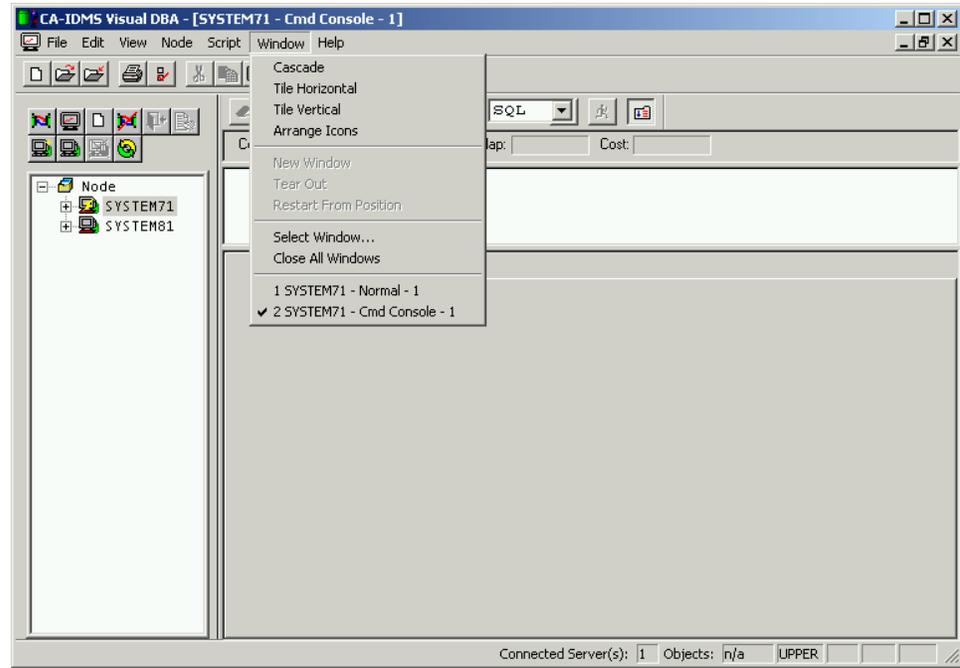
After you select a data source or node and connect to it, CA-IDMS Visual DBA displays a Database Object Manager window that looks similar to the following sample window.



After the initial connection, you can use the previous procedure to open multiple windows for the same data source or to connect to a different data source.

Note: If you choose to reconnect to the same data source, a new window opens and overlays the existing one. Use the functions in the Window menu to manage the windows.

In the following example, the same data source is used multiple times. The Window menu shows that it is used in both Normal and Command Console modes.



Securing CA-IDMS Visual DBA Objects

CA-IDMS Visual DBA uses CA-IDMS Server and CA-IDMS SQL to access the information stored in the dictionaries. Because of this, CA-IDMS Visual DBA users need to have the necessary authorization for the CA-IDMS resources that are being used.

This section describes the privileges you need to work with objects in CA-IDMS Visual DBA.

These securities involve:

- Viewing object instances in the object tree
- Updating object instances in the object tree
- Creating, altering, and dropping object instances in the object tree
- Granting and revoking user privileges, registering privileges for IDD dictionary entities, and recording user responsibilities for IDD dictionary entities
- DCMT security

CA-IDMS Visual DBA users must be granted privileges for resources of type TABL. The resource names for these TABL resource types are of the form *schema-name.table-name*. The appendix “Security TABL Resource Tables” shows the relationship between the CA-IDMS Visual DBA objects and the TABL resources.

Restricting Access to Objects

To restrict access to certain resource names, and thus to certain CA-IDMS Visual DBA objects, use the information in the appendix “Security TABL Resource Tables” to determine the resource names you need to secure. You can use wildcards in many cases to facilitate the security definition.

For example, to grant access for all objects except the View Definition:

1. Lookup View Definition in Objectnames in Alphabetic Order, Level and Resourcenames Table in the appendix “Security TABL Resource Tables”. The resource name for this object is SYSTEM.SYNTAX.
2. Check the Resourcename in Alphabetic Order, Level and Objectname Table in the appendix, which indicates that by securing resource name SYSTEM.SYNTAX, you automatically also restrict access to at least two additional objects – View Definition and Check Condition. This is because the system table SYSTEM.SYNTAX holds information for View Definition and Check Condition.

So, to limit access to all objects except the View Definition, you need to grant access to all resource names listed in the Resourcename in Alphabetic Order, Level and Objectname Table, except SYSTEM.SYNTAX. This also means that access to Check Condition is also restricted.

Keep in mind that, when limiting access to an object, normally all access to children of the object is also restricted. You do not need to explicitly restrict access to the child resource names. For example, when you restrict access to the SYSVNTWK.SYSTEM resource name, you restrict access to the System object and all of the child objects, such as Program, Queue, Line, and so on.

Some resource names refer to specific non SQL schemas as shown in this table:

Resource Name	Non SQL Schema
IDMSSECS2.*	IDMSSECS
IDMSSECU2.*	IDMSSECS
SYSTSCHM2.*	IDMSNTWK
SYSTSYST2.*	IDMSNTWK

SYSVSYST2 and SYSVNTWK2 are relational schemas that contain views on the non SQL schema IDMSNTWK

If access is granted to one of these resources, you also need to grant Use privilege for the corresponding Non SQL schema. For more information, see the SQL definitions in the CA-IDMS Visual DBA member VDB2Rxxx.SQL in your CA-IDMS source library or in the CA-IDMS Visual DBA mainfram folder

Assigning Privileges to View Object Instances

Tree navigation is a purely retrieval operation. To expand the whole tree in the Database Object Manager window, you must minimally have SELECT privilege for all the resource names of type TABL listed in Resourcename in Alphabetic Order, Level and Objectnametable Table in the appendix “Security TABL Resource Tables”.

Assigning Privileges to Update Object Instances

CA-IDMS Visual DBA uses the CA-IDMS compilers (SCHEMA, SUBSCHEMA, IDD, SYSGEN and BCF/OCF) and the DCMT programs to perform updates to object definitions. It runs these compilers with the help of table procedure SYSCA.VDBAIDD2, SYSCA.VDBADCT2 and SYSCA.VDBADCU2.

To perform any updates with CA-IDMS Visual DBA, you need the SELECT privilege on resource name SYSCA.VDBAIDD2, SYSCA.VDBADCT2 and SYSCA.VDBADCU2.

Similarly, to globally prohibit updates of any of the CA-IDMS Visual DBA objects, you can revoke the SELECT privilege on resource name SYSCA.VDBAIDD2, SYSCA.VDBADCT2 and SYSCA.VDBADCU2 for a user.

Assigning Privileges to Create, Alter, and Drop Object Instances

To create an object instance by altering an existing one, or to alter or drop an object instance, you must be able to view the object and any related objects that might appear in lists in Create/Alter dialogs. To create an object instance without altering an existing one, you do not need to be able to view the object.

To execute the syntax generated by CA-IDMS Visual DBA for the CA-IDMS compilers or DCMT programs, you must be able to pass all standard CA-IDMS security checks. In this way CA-IDMS Visual DBA does not differ from any compiler invoked directly on a CA-IDMS system; that is, to Drop an instance of the DMCL object, you need to have Drop privileges for that DMCL instance.

Assigning Privileges to Grant and Revoke Object Privileges

To invoke the security dialogs, you need to have the authorization to view the grantee (Central User, Dictionary User and Group), and the instances of the granted object.

To execute the security syntax generated by CA-IDMS Visual DBA for the CA-IDMS compilers, you need to pass all standard CA-IDMS security checks. In this way CA-IDMS Visual DBA does not differ from defining security through any direct invocation of BCF/OCF or IDD.

What's Next

Now that you have learned how to start CA-IDMS Visual DBA, connect to a server, open the Database Object Manager window, and secure its objects, you can perform a variety of database administration tasks. The next chapter introduces you to the features of the Database Object Manager Window.

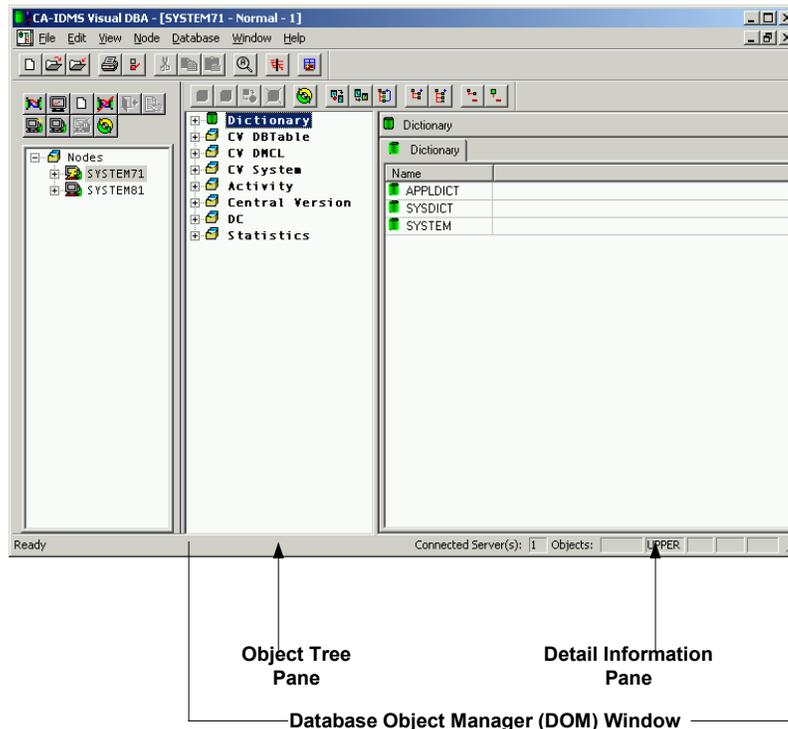
A Tour of the Database Object Manager Window

The DOM Window

At the heart of CA-IDMS Visual DBA, the Database Object Manager provides a convenient and organized way to view and manipulate the database information that is currently stored on a particular server.



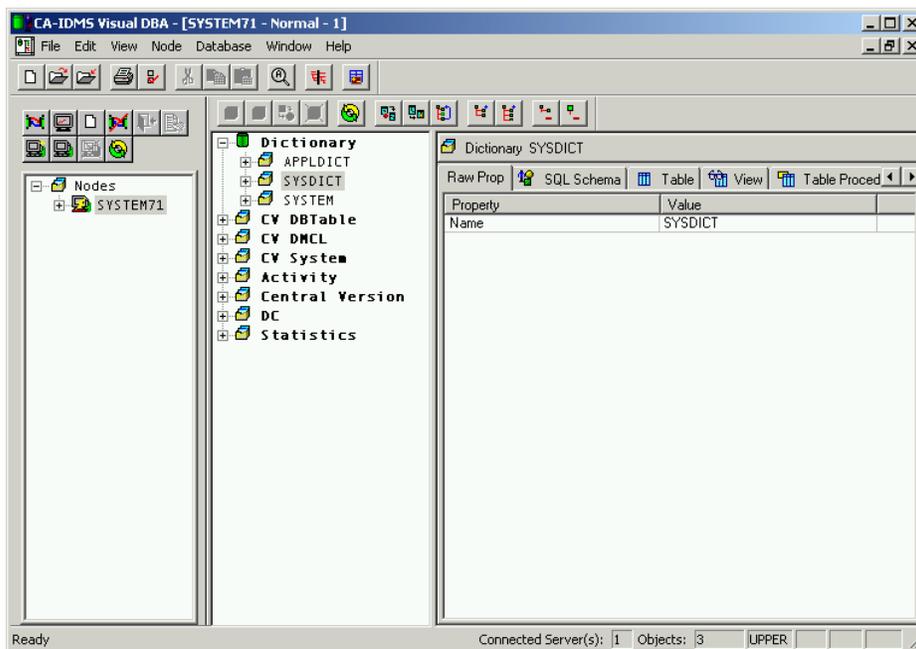
The Database Object Manager (DOM) window appears after you connect to a CA-IDMS database:



When the DOM window opens, it displays all the available object types at root level in the Object Tree pane. Initially, the Dictionary object type is highlighted and the level-one instances of the Dictionary object (APPLDICT, SYSDICT, SYSTEM) display in the Detail Information pane. The caption in the Detail Information pane reflects the identification of the highlighted object type.



In the Object Tree pane, you can change the object type that you want to view. In the Detail Information pane, you can drill down to display the information you want about an object. When you click an object, the Detail Information pane displays labeled tabs that provide attribute information. As discussed later in this chapter, the tabs that are displayed are dependent upon the type of object selected.



Using the Database Object Manager, you can create, alter, drop, and display CA-IDMS database objects, and assign privileges to them. Additionally, you can open and simultaneously work with multiple databases and/or servers in multiple windows.

Using the DOM Window Toolbar

As described below, the DOM Window Toolbar enables you to perform the following functions:



Button	Description
	Creates an object in the active Database Object Manager
	Creates an object and the children of that object in the active Database Object Manager window.
	Changes the attributes of an existing object.
	Drops an object.
	Refreshes data on the fly for either a single branch, all branches, or any object type in the currently selected Database Object Manager window.
	Opens a copy of the current Database Object Manager window.
	Lets you view any selected branch of a Database Object Manager window as the root branch of a new Tear Out window. Your original Database Object Manager window remains intact.
	Converts any selected branch of a Database Object Manager window into the root branch of that window.
	Displays all the sub-branches associated with the current branch. Note: This command fully expands only those sub-branches that are not associated with a cross-referenced object (so that the tree does not expand infinitely).
	Expands and displays all the branches and sub-branches of all the root objects in the current DOM window. Note: This command fully expands only those sub-branches that are not associated with a cross-referenced object (so that the tree does not expand infinitely).
	Collapses the sub-branches of the current branch. Note: When this command is used, CA-IDMS Visual DBA remembers the expansion state of the lower-level sub-branches (that is, the sub-branches of the sub-branches, and so on). When the current branch is subsequently expanded, the previous expansion state is duplicated.

Button	Description
	Collapses all sub-branches associated with every root object category branch.

Expanding and Collapsing the Object Tree

The information in the Database Object Manager window is arranged in a tree structure that clearly shows the relationships among the pieces of information on a server. This Database Object Manager tree initially displays a group of categories or *branches*. The tree is collapsible and expandable—to view related information, merely double-click on the branch of your choice.

Using this tree, you can manipulate information. For example, you may alter an object’s characteristics or change the properties associated with an object.

As you start to add, modify, or delete objects on the server, the Database Object Manager will display the latest information.

Note: You can have more than one Database Object Manager window open at the same time, with each window connected to the same or different data sources:

Expanding the Tree



Since the ability to expand and collapse the tree branches in the Database Object Manager window is an important feature of CA-IDMS Visual DBA, it is a good idea to familiarize yourself with the different ways this can be accomplished.

One Level at a Time



To expand a branch a single level, click on its expansion button (⊕), which displays the sub-branches that exist one level down within the selected branch. (Note that only expandable branches have this button.) You may also double-click anywhere on the branch to expand it. In addition, it is possible to expand a branch by using the Plus (+) key, or the Expand One Level menu command on the View menu.

Expanding Multiple Levels



You can also use the toolbar, control keys, and the View menu when you want to see all the sub-branches under a single branch. Simply select the branch and click on the Expand Branch toolbar button, press the Multiply (*) key, or use the Expand Branch command from the View menu.



Similarly, to view all sub-branches for *every* branch, click on the Expand All Branches toolbar button, press Ctrl + * , or choose the Expand All Branches command from the View menu.

Infinite Drill

One of the most useful aspects of CA-IDMS Visual DBA is its “infinite drill” feature, which graphically illustrates the complex relationships that exist among objects. By expanding – or drilling down through – branches in the tree, you will see that nested within most new sub-branches are lower-level sub-branches that contain related information.

Further, combining infinite drill-down with the features described in the Changing the Tree Structure section, allows you to turn *any* sub-branch into a root branch from which you can drill down.

Collapsing the Tree



To collapse a sub-branch, choose the collapse button (☐) to the left of the object category name or use the appropriate toolbar buttons or menu commands from the View menu, as described below.



You can use the toolbar, control keys, and the View menu when you want to collapse all the sub-branches under a single branch. Simply select the branch and click on the Collapse Branch toolbar button, press the Minus (-) key, or use the Collapse Branch command from the View menu.



Similarly, to collapse all sub-branches for *every* branch, click on the Collapse All Branches toolbar button, press Ctrl + / , or choose the Collapse All Branches command from the View menu.

Changing the Tree Structure

To facilitate your ability to shift your perspective of the database at the touch of a button, CA-IDMS Visual DBA offers three distinct features: Restart from Position, Tear Out and Scratchpad.

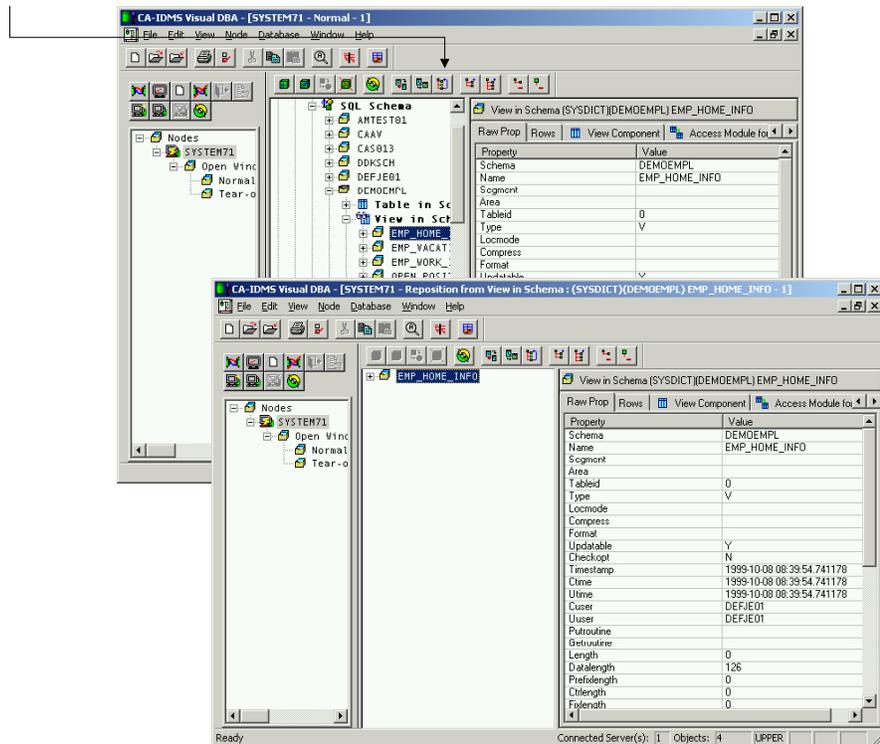
Restart from Position



The information you need often begins on a level far beneath the root object category branch. The layers of sub-branches that you must pass through may be extraneous, in this instance, so you may want to remove them from view. To do this, use the Restart From Position toolbar button to execute the command that converts any branch you select into the root branch of your window.

For example, in the following window, the branch, “EMP_HOME_INFO” is nested deep within the tree. Clicking on the Restart from Position toolbar button changes the window, as follows:

Click Restart from Position button to display deeply nested objects at the root level.



Now, “EMP_HOME_INFO” is the root branch and may be expanded using the same techniques previously described.

Tear-Out Window

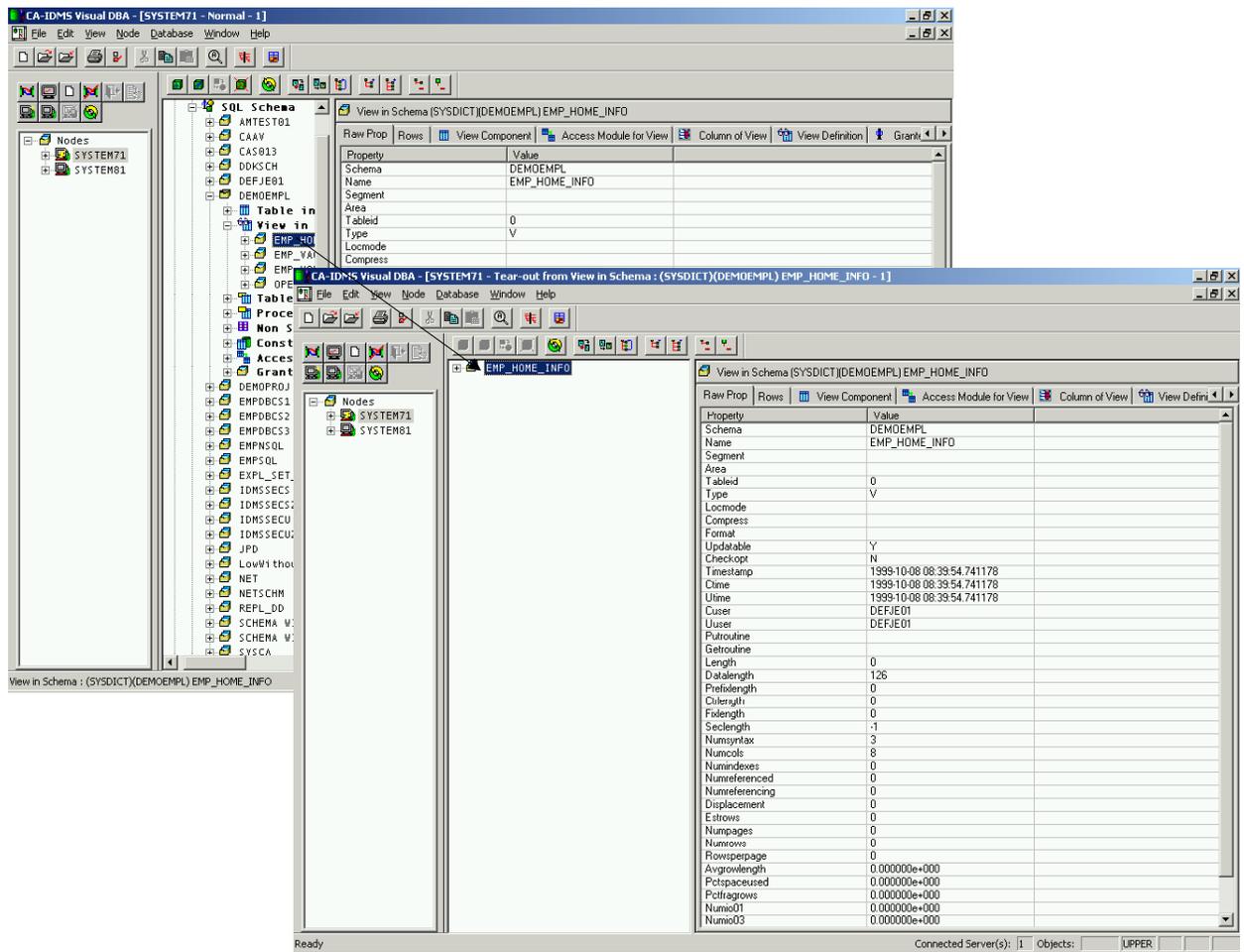


You have seen how the Restart From Position command on the Windows menu alters the existing Database Object Manager window. If you want to view a branch in a *new* window, you may use the Tear-out feature.

By selecting a branch and clicking the Tear Out toolbar button (or choosing the Tear Out command from the Window menu), you can create a new window, while leaving the original window intact. The branch you select becomes the root branch in the new window and may be expanded to reveal all existing sub-branches.

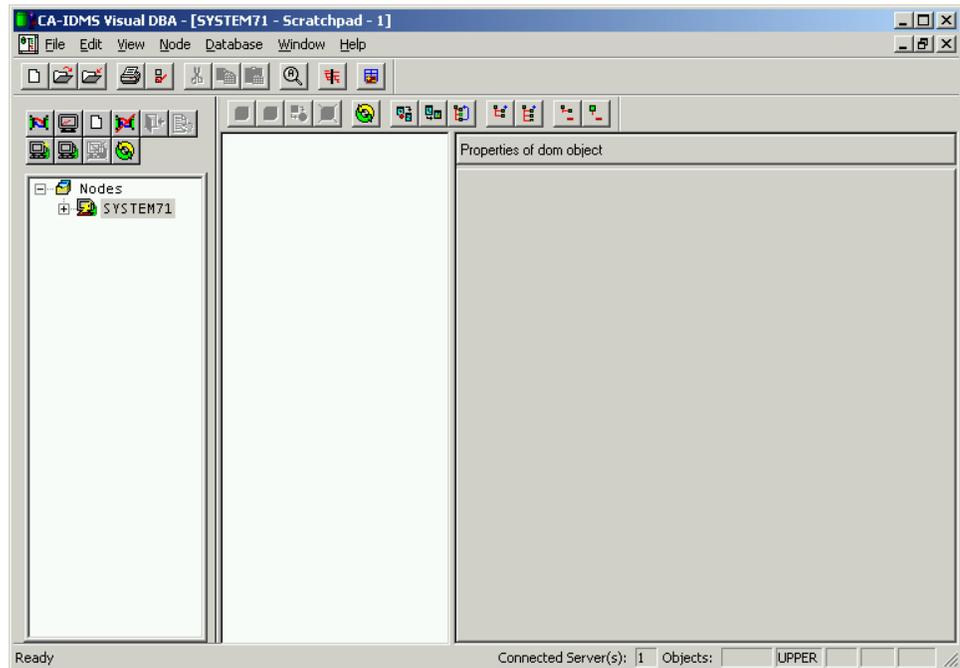


As an example, let’s look at the Database Object Manager window that was used in the previous section. By clicking on the Tear Out toolbar button, a new Database Object Manager window is opened, with “EMP_HOME_INFO” as the root branch:

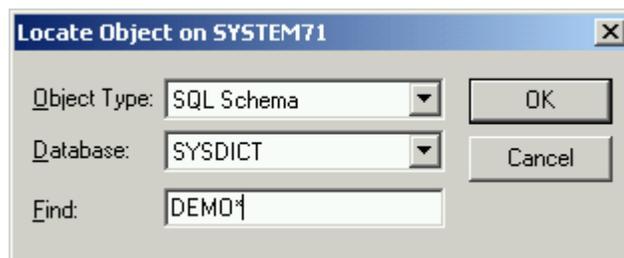


Scratchpad

The DOM Scratchpad command on the Node menu offers yet another powerful way to construct your own customized Database Object Manager window. The Scratchpad window is empty when it is first opened.



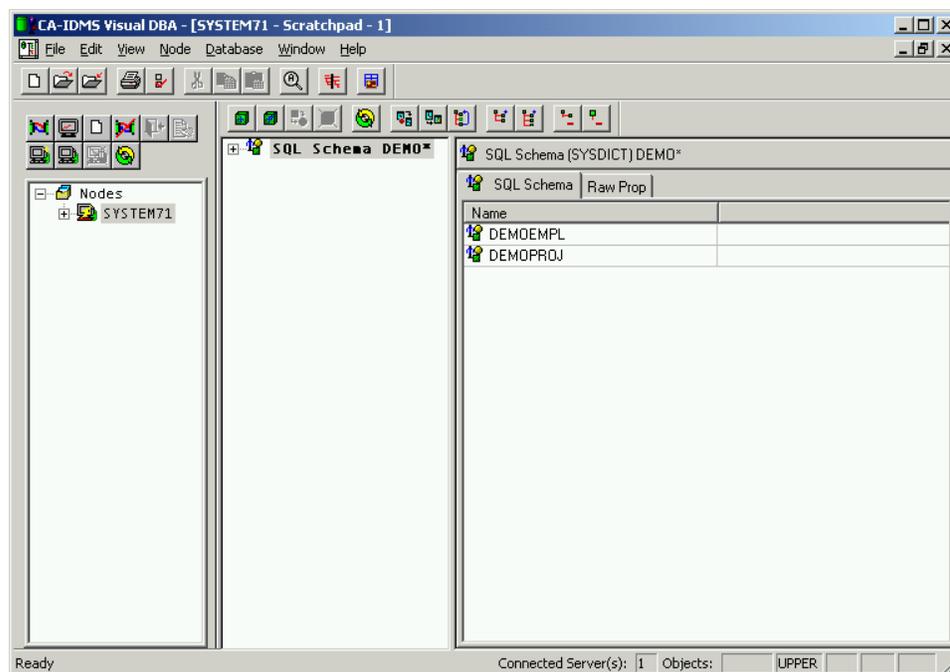
CA-IDMS Visual DBA allows you to easily find any object on the current server and display it as the root branch in the Scratchpad window. To do this, open a Scratchpad window and then choose the Edit Locate menu command, which invokes the Locate Object dialog box:



In the Object Type drop-down list box, choose the object category that you wish to find. In the Find edit control, leave the asterisk (*) unchanged if you want to locate the object category branch. Otherwise, enter the full name of an object (or a wildcard expression).

For example, you may want to find the SQL schemas that start with “demo”. To do this, select SQL Schema from the Object Type drop-down list box. Next, enter **demo*** in the Find edit control, and choose OK. A single branch labeled “SQL Schema demo*.” appears in the Scratchpad window.

Expanding this branch reveals the branches for SQL schemas whose names start with DEMO:



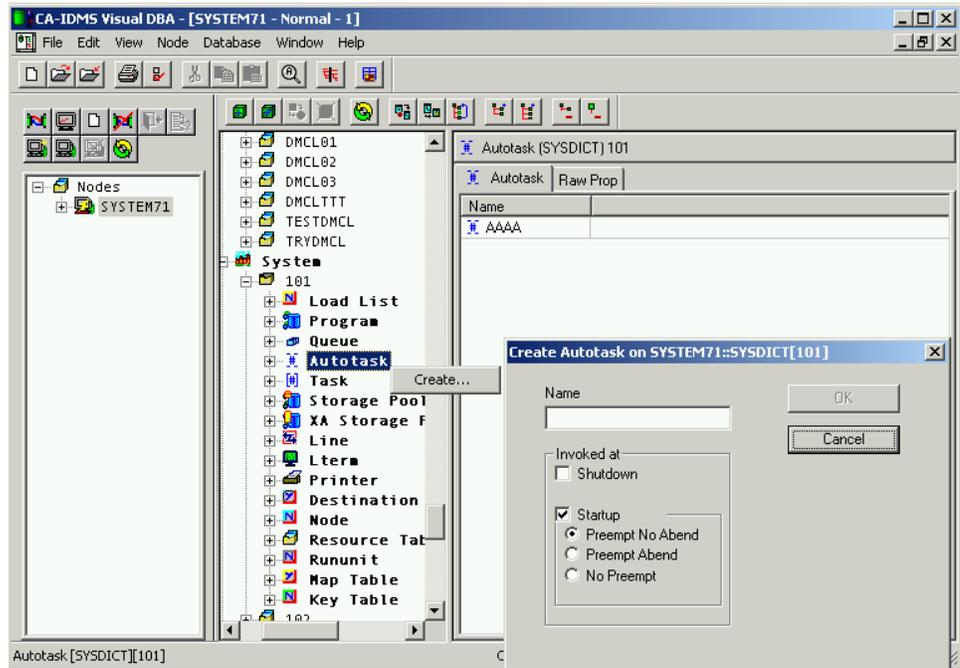
The Window Restart From Position, Tear Out, and Scratchpad menu commands all enable you to construct your own Database Object Manager windows that focus precisely on the information *you* need.

Manipulating Tree Objects

Once you navigate your way to the object in the tree that you want to manipulate, you can create a new object entity, modify an existing one, delete it, display it, and define privileges for it.

To do these things, select the object and right-click. A popup menu appears which displays the actions allowed for the object. Alternatively, you can select the object and use the commands on the Edit menu.

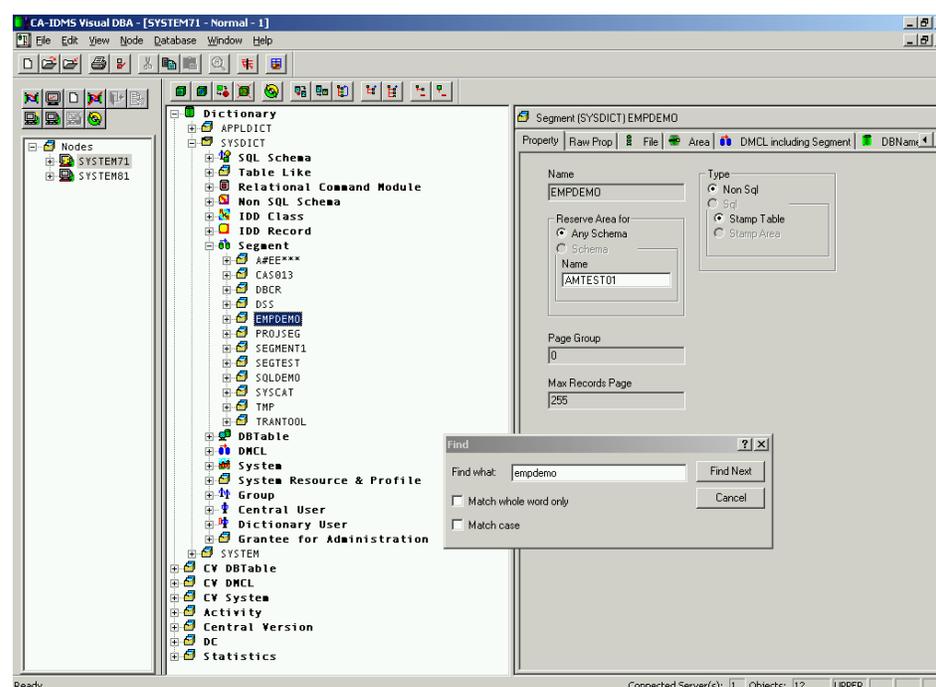
In this screen shot, the Autotask system object is selected. The popup-menu offers the option of creating a new system autotask. When you click Create, a dialog appears that guides you through the Autotask definition. If you need help understanding the dialog options, press F1 for detailed field-level help.



Finding Objects



CA-IDMS Visual DBA offers a search tool to search for an object or object category in an expanded tree. For example, if you want to find a specific segment, click the Find button (or use the Edit Find command). CA-IDMS Visual DBA displays a dialog that prompts you for the search string and other search criteria. When the segment is found, CA-IDMS Visual DBA highlights it in the Object Tree pane and displays associated information in the Detail Information pane.



Selecting and Copying Objects

To select multiple objects in the same branch of a tree or in the child tab of the Detail Information pane:

- Click the left-mouse button and hold the Ctrl key to select a non-contiguous group of objects
- Click the left-mouse button and hold the Shift key to select a range of objects

You can also copy object definitions from one sub-branch or from the Detail Information pane to any other sub-branch that has objects of the same type. This includes copying definitions from one IDMS central version or system to another.

To copy the object definitions, simply select the objects, and drag and drop them on to the target database. CA-IDMS Visual DBA does the appropriate copy and verifies that any integrities involved in the entities are properly maintained.

Drag-and-drop techniques simplify database administration tasks, especially for test and production database maintenance.

Viewing, Executing, and Logging Object Syntax

CA-IDMS Visual DBA lets you decide what happens when you create, alter, or delete an object and then click OK. You can view, execute, and log the syntax that CA-IDMS Visual DBA creates.

To choose the action that occurs when you click OK, select Preferences from the File menu and then click the OK Actions icon in the Preferences window. For each CA-IDMS compiler (IDD, OCF/BCF, schema, subschema, sysgen, or DCMT), select one or more check boxes to log, view, and execute syntax.

By default, CA-IDMS Visual DBA executes the object syntax and displays the syntax only if an error occurs. If you select View Syntax only, it displays, but does not execute the syntax. If you select View Syntax and Exec Syntax, it displays the syntax and asks if you want to execute it.

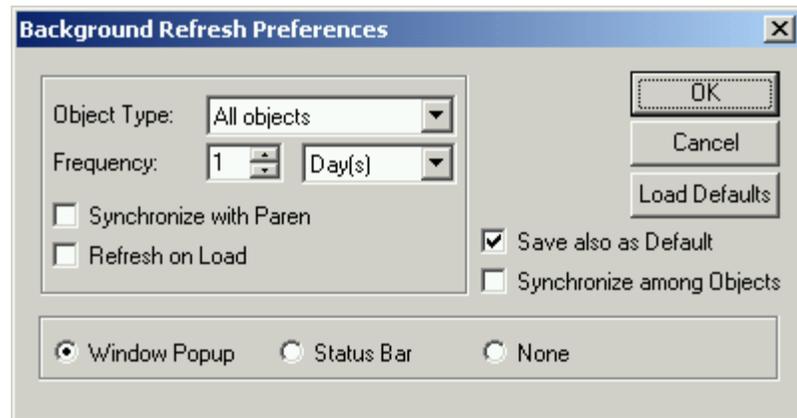
If you select Log Syntax, CA-IDMS Visual DBA creates a log file with an extension that designates the compiler and logs all object syntax for that compiler to the log file. Later, you can open the log file, edit it if you want to, and then upload and execute it as a batch file on the mainframe. You can also execute the log file as a command script in a command console.

Refreshing the Tree



With the Force Refresh command on the View menu or toolbar, you can refresh data "on-the-fly" for either a single branch, all branches, or any object type.

The alternative to the Force Refresh command is Activate Background Refresh, also on the View menu. This command instructs CA-IDMS Visual DBA to refresh the data at a specified frequency, using the parameters defined in the Background Refresh Preferences dialog invoked by the File Preferences command.



You can set the Background Refresh Preferences so that different object types are refreshed at different frequencies.

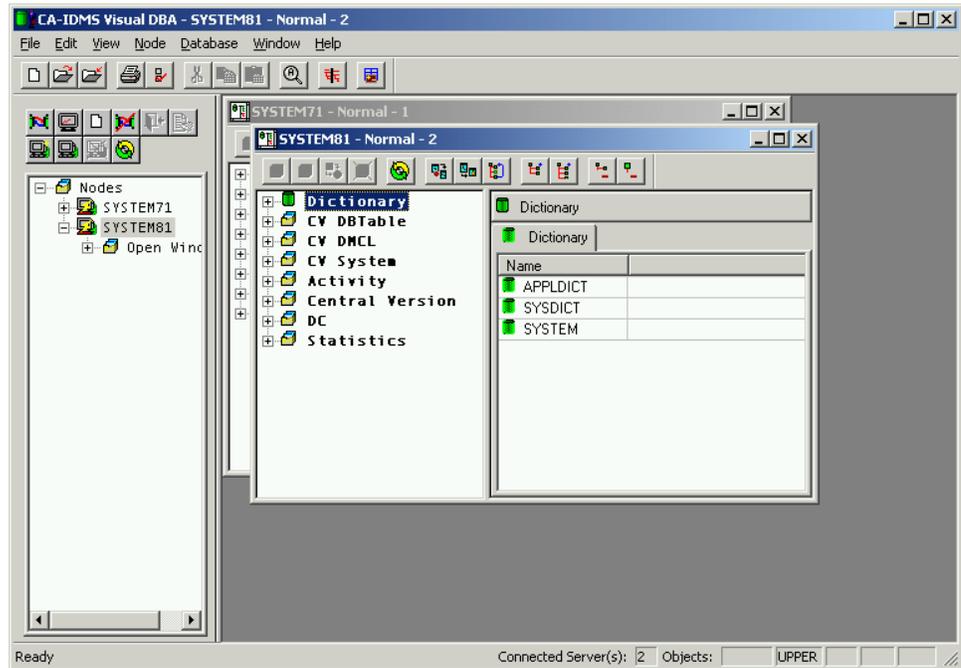
You can also save your refresh setting as default settings to use with current and new workspace environment configurations.

Connecting to Multiple CA-IDMS Systems



To connect to more than one CA-IDMS system, simply select the node from the Node Window and click the Connect button.

CA-IDMS Visual DBA opens another Database Object Manager window, which displays the object tree data for the CA-IDMS system that you just connected to. Using the options on the Windows menu, you can display the open windows in a tiled or cascaded array. Note that the status bar shows how many servers you are connected to.



When you connect to multiple CA-IDMS Visual DBA sessions, you can use the Sessions option in the File Preferences command to limit the number of sessions you can run at one time.



Displaying Object Attributes in the Detail Information Pane

When you click an object in the Object Tree pane, the Detail Information pane displays labeled tabs that provide attribute information. Additional tabs correspond to the child objects for the highlighted object in the Object Tree pane. Each tab provides a specific type of information about the object. The different categories of information are listed below:

Tab	Description
Property	Displays attribute information in the same format as the Alter dialog for the object type. If the Alter dialog has one or more subdialogs, then a separate property tab is displayed for each subdialog, with a tab name that is the same as the subdialog name.
Raw Prop (Object Instance or Dynamic Object)	Displays a table with Property and Value columns. The Property column displays the names of the attributes that exist for the object. The Value column displays the value of the attribute. You cannot alter the object values in the Raw Prop tab.
Raw Prop (Object Class or Static Object)	Displays a table with a column for each attribute. Each row displays the values for all attributes of one object instance.
Rows	Displays information for table-like objects. Visual DBA retrieves rows of information based on the Rows in Cache value that you have defined in the Command Console Dialog. When you scroll to the bottom row, Visual DBA moves the bottom half of the table rows (resulting from the last retrieval) to the top of the table, performs another retrieval, and displays half of the newly-retrieved rows at the bottom of the table.
Grantee	Displays grantees, registrees, or responsables for a privilege. The information is presented in table format with check marks for privilege types. The table can be sorted by column. Simply double-click the column to reorder the table. The tab is labeled either Grantee for, Registree for, or Responsible for followed by the privilege or privilege object. For example, Grantee for Table.

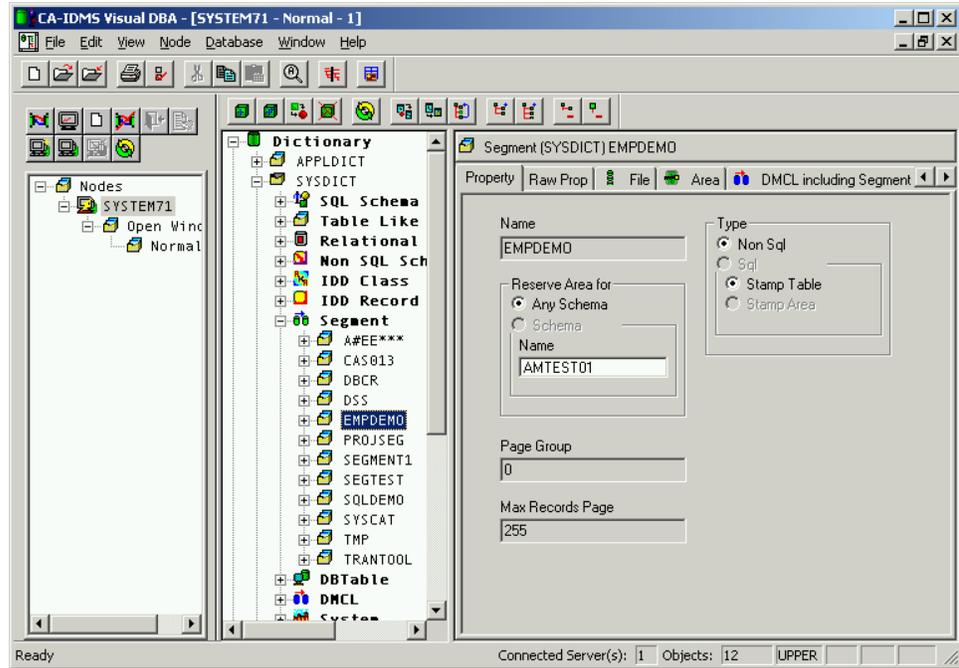
Tab	Description
Privilege	Displays privilege information granted for the Central User, Group, and Dictionary User object instances. The information is presented in table format with check marks for privilege types. The table can be sorted by column. Simply double-click the column to reorder the table.
Pie	Displays information for Active Program or Active Storage object types. The pie chart relates the active usage to the available pool. When you click on the pie slice representing actual usage, Visual DBA displays detailed statistics.
Bar	Displays information about Statistics objects, including Statistics Segments, Statistics Files, Statistics Areas, and Statistics Buffers. You can select any part of the bar diagram to display more detailed information.
Child	<p>Displays the instances of the selected child object class. The label of the tab always consists of an icon followed by the child object class (also called static object) name. The icon is the same used in the label of the object class in the tree and which is visible in front of the object names in the Detail Information pane.</p> <p>Note: For this type of tab, it is possible to double-click any instance of the child object in the Detail Information pane. The tree automatically expands and the selected instance – which is now the current selected object – becomes visible in the tree window. These instances can also be selected for further Alter, Create, Drop, or Drag and Drop operations.</p>

Tab Examples

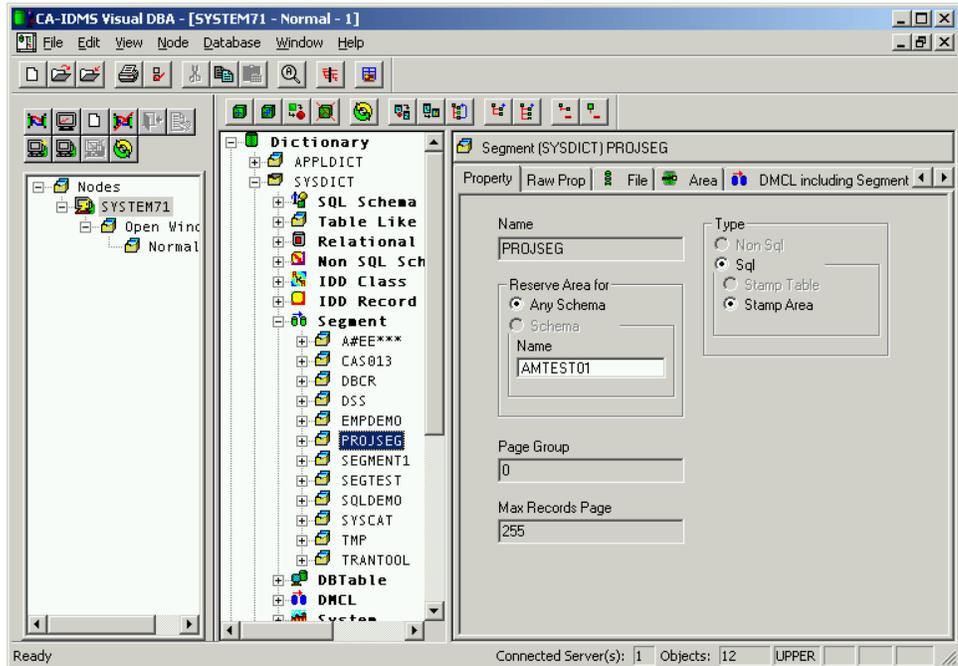
The following examples illustrate different types of tabs:

Property

In the following example, CA-IDMS Visual DBA displays Property tab information for the SYSDICT Segment, EMPDEMO. Note that the caption in the right pane repeats the identification information of the object and its icon. Below the caption, a series of labeled tabs provide different types of attribute information.

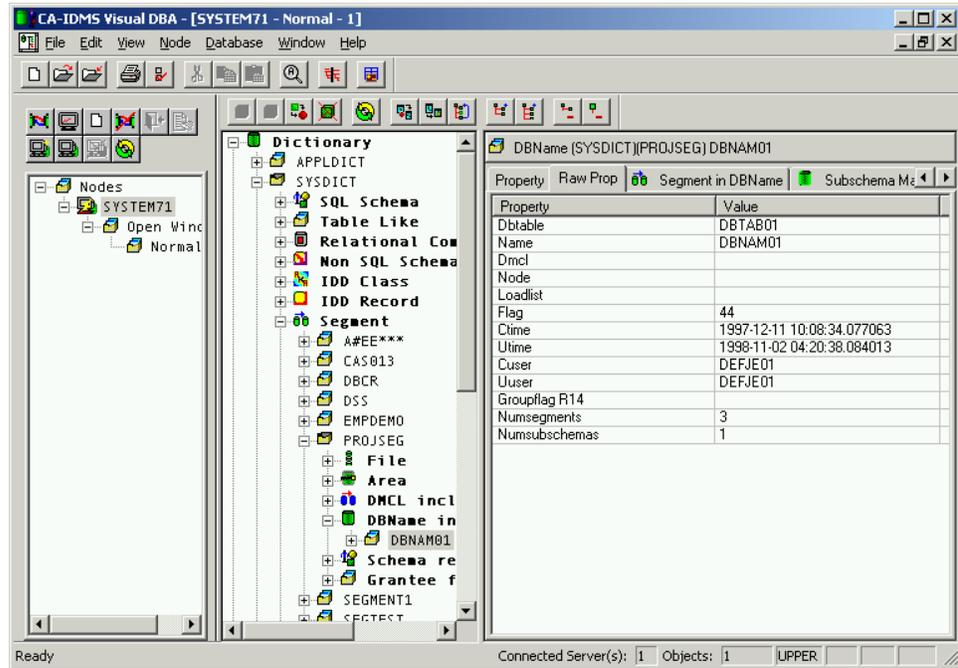


As illustrated in the next example, we can display Property information for another instance of the Segment object class by simply selecting that object from the tree. CA-IDMS Visual DBA retains the tab selection, enabling you to easily make comparisons between object classes and instances.

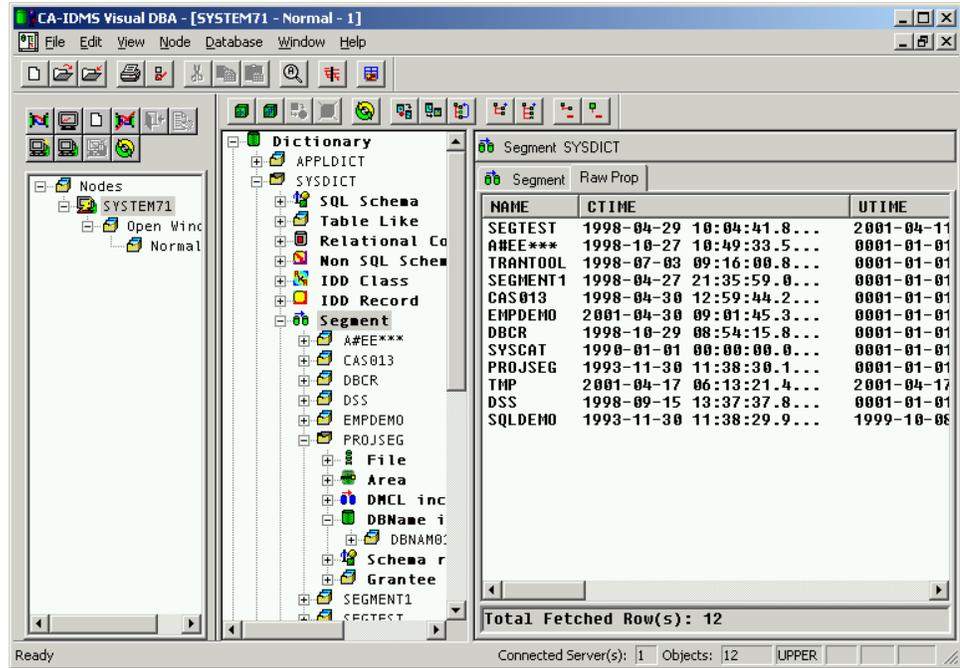


Raw Prop

In the following example, CA-IDMS Visual DBA displays Raw Prop information for the DBNAM01 instance of the Object Class Segment.

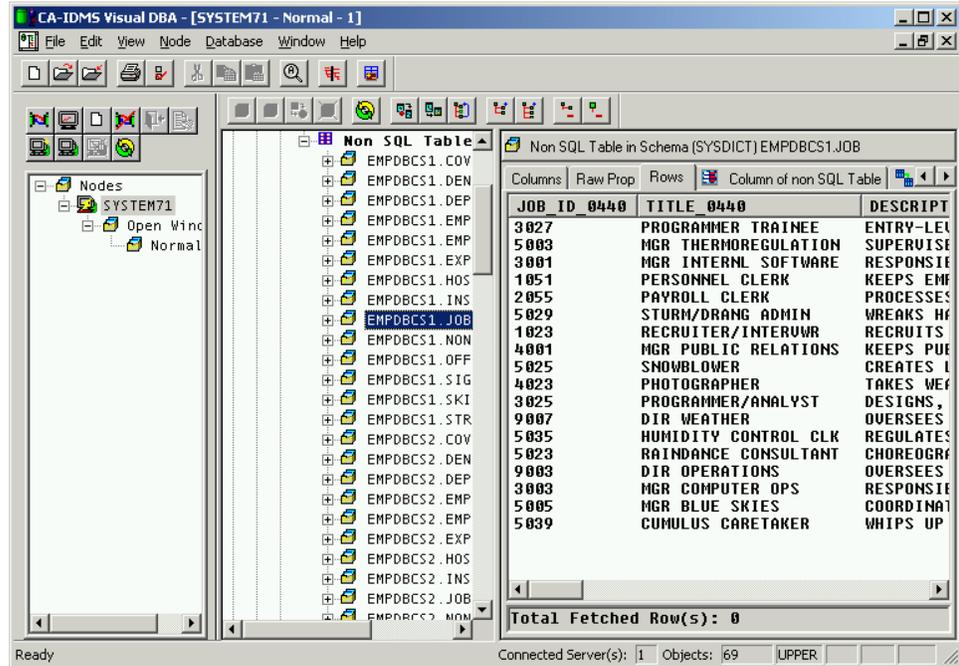


When you select an Object Class or static object, CA-IDMS Visual DBA displays a table with a column for each attribute and a row for each object instance.



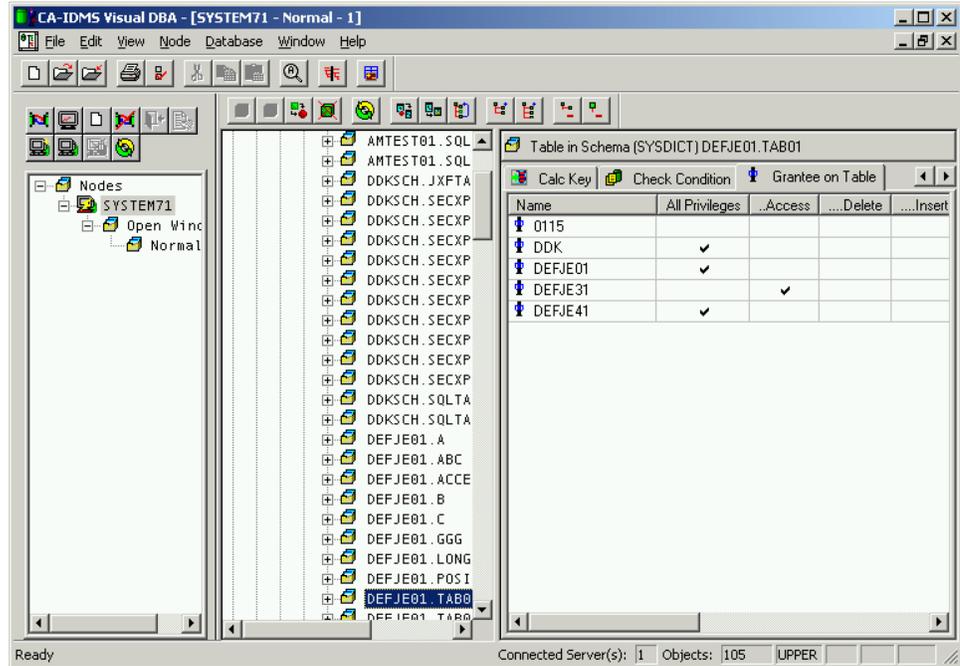
Rows

In the following example, CA-IDMS Visual DBA displays row information for EMPDBCS1.JOB, an instance of a Table Like object in the Non SQL Table in Schema Object Class.



Grantee

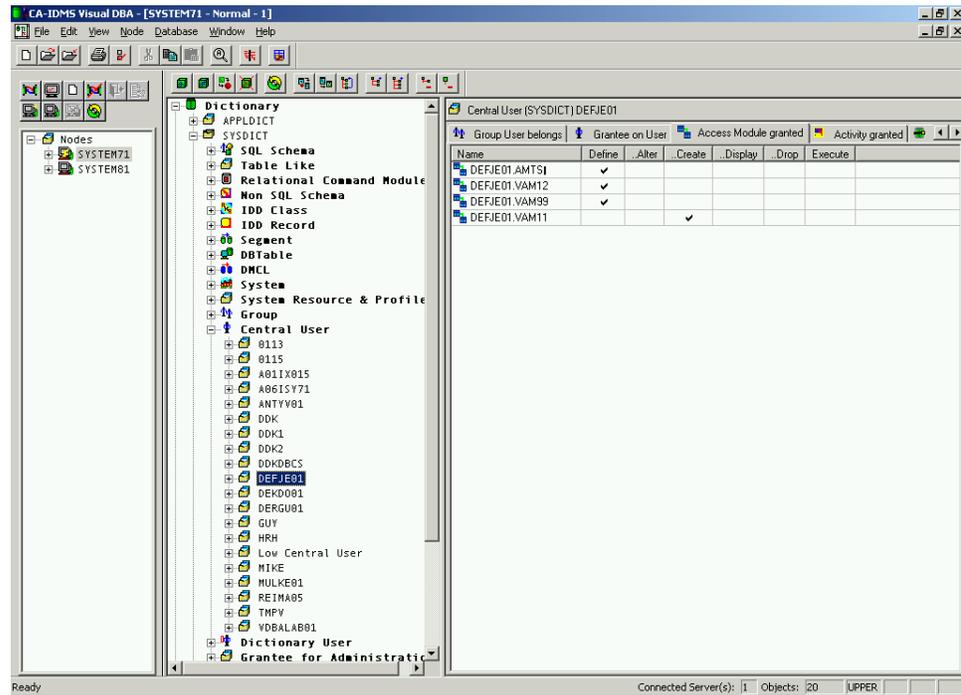
In the following example, CA-IDMS Visual DBA displays grantee information for the Table in Schema DEFJE01.TAB01. Note that the table includes a column for each privilege type and a row for each User ID.



By clicking on the column header, you can sort the data by that column information. When we double-click the Access column, the rows are sorted in order of User ID with that privilege.

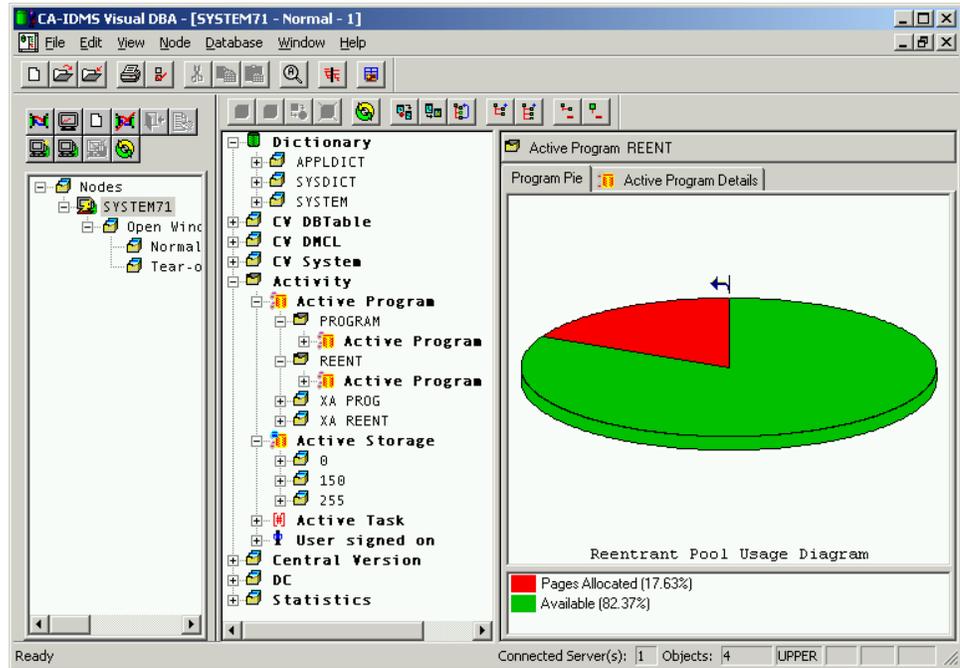
Privilege

In the following example, CA-IDMS Visual DBA displays Privilege information for Central User DEFJE01. Each tab represents a different privilege type. Within each tab, a row appears for each user ID with the selected privilege type. You can sort the column by double-clicking the column header.

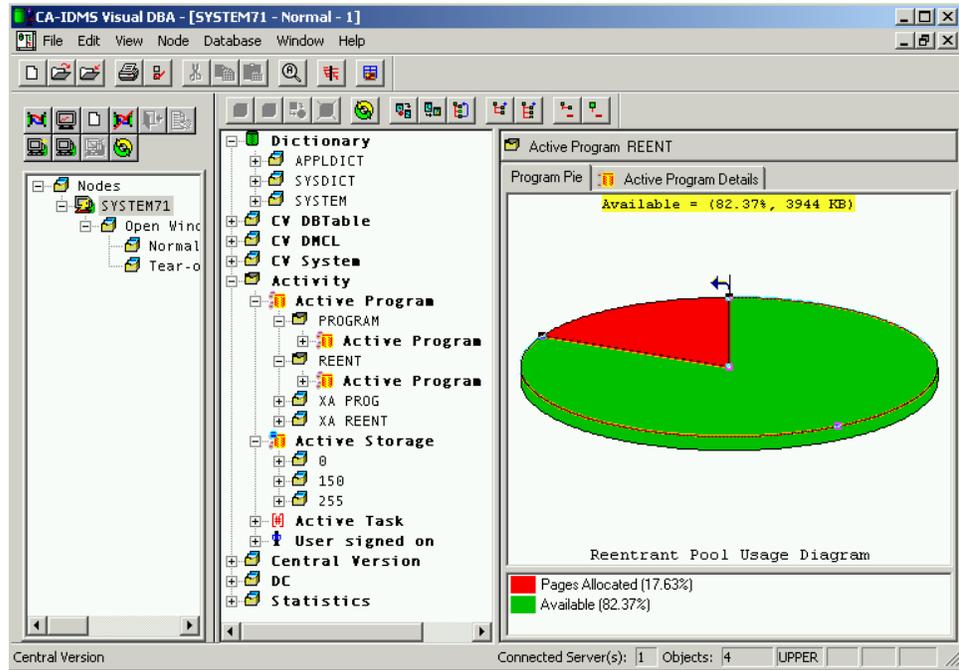


Pie

In the following example, CA-IDMS Visual DBA displays Reentrant Pool usage in pie chart format.

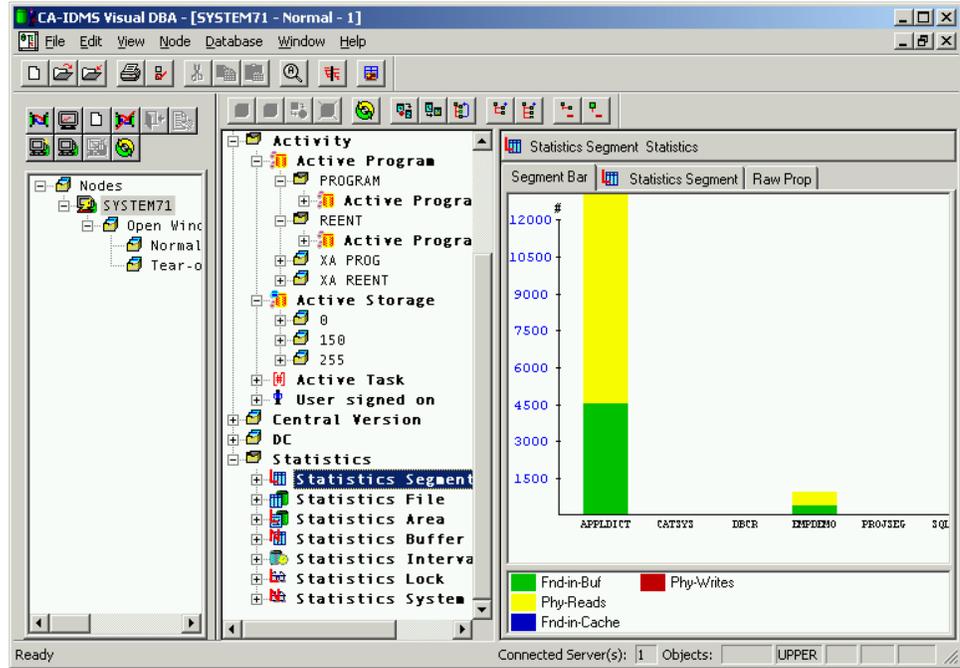


When you click on a pie segment, the actual usage statistic is displayed, as shown below.

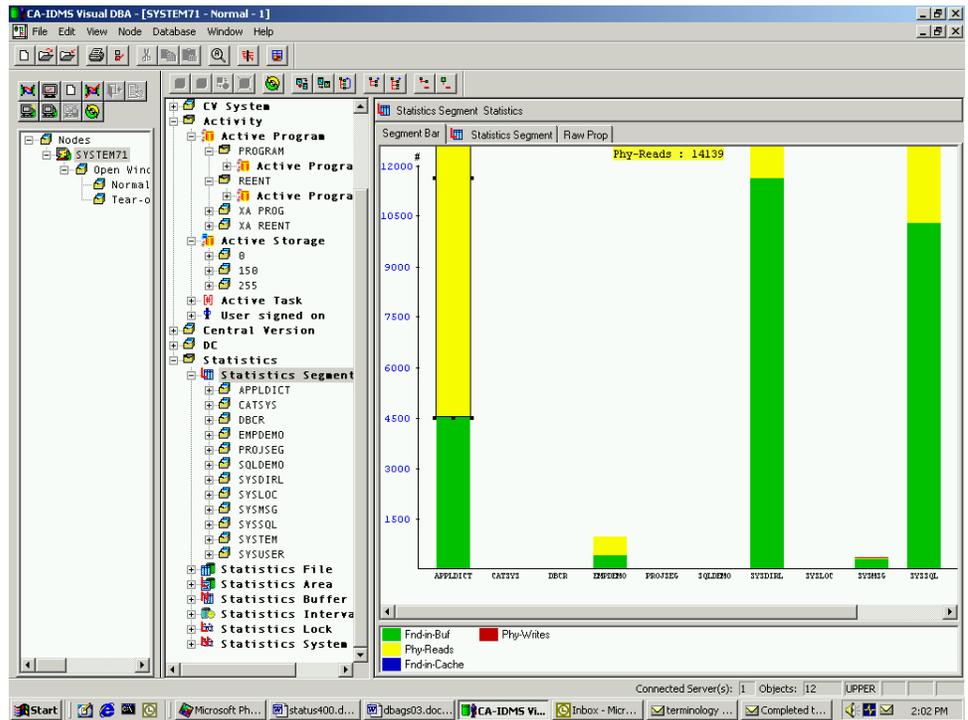


Bar

In the following example, CA-IDMS Visual DBA displays statistical information in bar chart format for Statistics Segment. Note that the information is color coded.

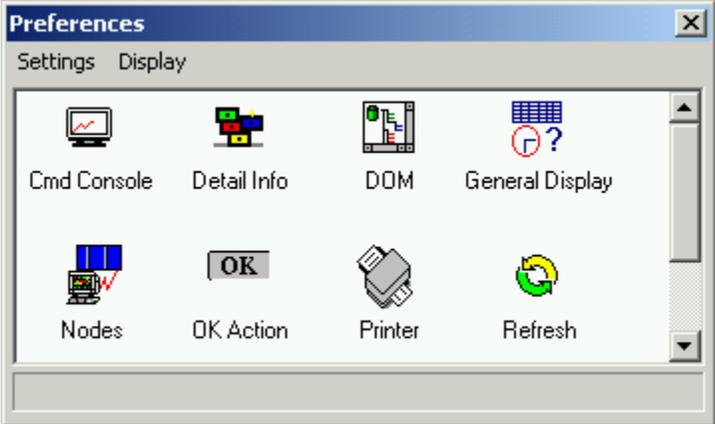


When you click on a portion of the bar, the actual statistics for that segment display. In the following example, we click on the yellow portion of the bar for APPLDICT. The actual number of physical reads (Phy-Reads) displays at the top of the pane. When you click on any of the names shown along the X-axis, all the statistics for the selected name are displayed.



Setting Display Options for Your Session

To customize your CA-IDMS Visual DBA environment even more, you can use options found under File Preferences to select display options for your sessions.



These include:

Item	Description
 Cmd Console	Sets properties for the Command Console, such as the font, number of tabs for history, trace tab size, and the number of rows in cache.
 Detail Info	Sets the font used in most of the Detail Information pane.
 DOM	Sets the font of the Database Object Manager window.
 General Display	Sets general display options, such as the time interval before animation dialogs are displayed, and whether the first document should be maximized at next startup.
 Nodes	Sets the font of the Nodes window.
 OK Action	Sets the action performed when you click OK for CA-IDMS objects that you create or alter. You can choose to log, view, or execute syntax for each object compiler.

Item	Description
 Printer	Sets the printer and defines properties such as paper size and orientation.
 Refresh	Sets the refresh frequency for specified object types.
 Sessions	Sets the number of CA-IDMS sessions allowed and the timeout period.

Creating, Opening, and Saving Environments



Because CA-IDMS Visual DBA gives you so many ways to customize your session options and object tables, naturally, it lets you save your customized environments so that you do not have to re-define them each time you start a session.

For example, suppose you ‘tear-out’ a branch of the object tree and display it in a new window. You can save that view of the tree in a configuration (.CFG) file by using the File Save As command.

The next time you want to use that view of the object tree, you can open the configuration file using the File Open command. Similarly, you can create a new configuration file by clicking File New.

Trying It Out Yourself

Now let’s apply the features of the Database Object Manager to work. In this example, we will create and modify a non-SQL subschema based on an existing subschema definition. To do this, we will use:

- A tear-out window
- The drag-and-drop feature to copy an existing subschema area to the new subschema definition
- The Create and Alter object options

First, open CA-IDMS Visual DBA and connect to a CA-IDMS data source that contains the Commonwealth demo database supplied with CA-IDMS installation. After logging on, drill down the root structure to the subschema object by double-clicking the:

- Dictionary object
- The APPLDICT dictionary object
- Non-SQL schema object

Next, right click the Subschema object and select Create. In the Create Subschema dialog, enter a name for your subschema, such as MYSUBSCH and then click OK:

CA-IDMS Visual DBA refreshes the Database Object Manager and displays the name of the subschema you just created.

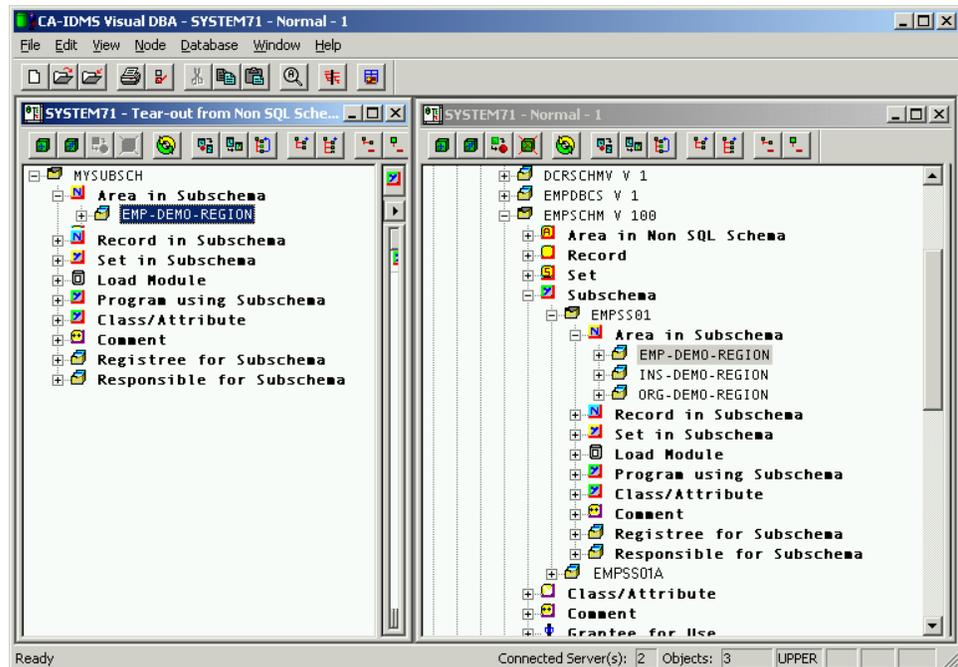


Now, let's populate the new subschema with information copied from an existing subschema. To make it easier to view, first 'tear-out' the subschema you created and display it in a new window. To do this, select the subschema, MYSUBSCH, and click Tear Out from the Window menu. A new window appears with your subschema at the root level. Next, click Tile Vertical button to display both windows side-by-side.

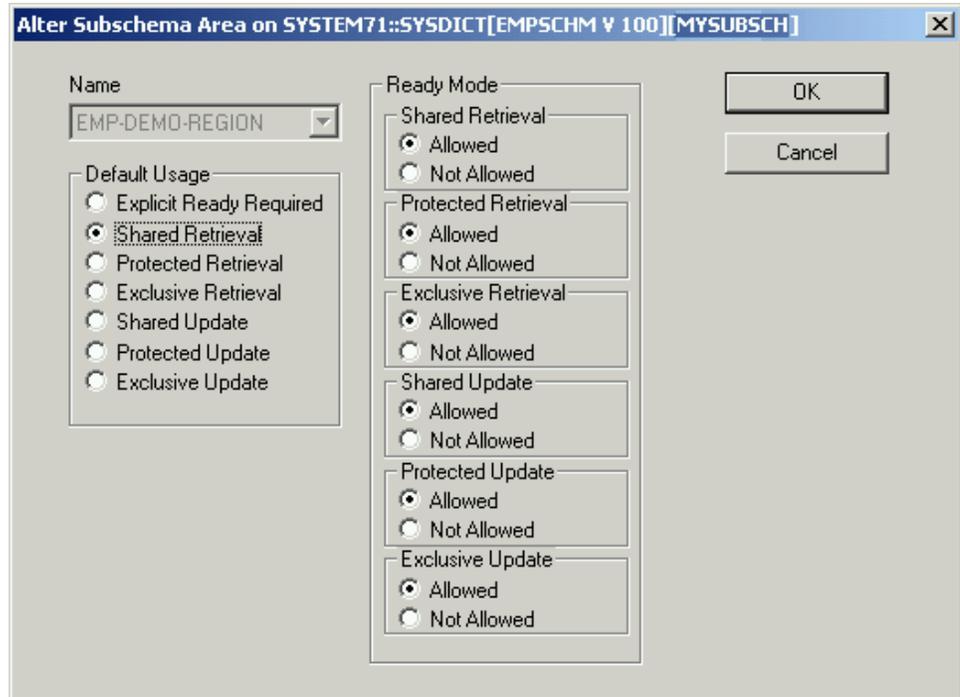
In the original window, double-click the EMPSS01 subschema and then double-click Area in Subschema. We are going to copy the EMP-DEMO-REGION area to the subschema you just created. In the 'Tear-Out' window, double-click MYSUBSCH to display the subschema objects.

To copy the EMP-DEMO-REGION area to the new subschema, drag and drop EMP-DEMO-REGION from the original window to the Area in Subschema object of the 'Tear-Out' window. Notice that the mouse cursor changes to the Area icon when it is positioned over Areas in Subschema.

If you successfully copied the area, your window should look something like this:



Finally, let's modify the default usage of the EMP-DEMO-REGION area in the new subschema. To do this, select it and right-click. From the popup menu, click Alter. Click Shared Retrieval under Default Usage and then click OK:



You have just completed a brief tour of the Database Object Manager. You have seen how easily you can use simply point-and-click techniques to create, copy, and modify database objects.

What's Next

This chapter provided you with an overview of the Database Object Manager window and how to use it to display CA-IDMS objects and manipulate them. Now that you have a general understanding of the Database Object Manager window, let's move on to the Command Console window.

Using the Command Console

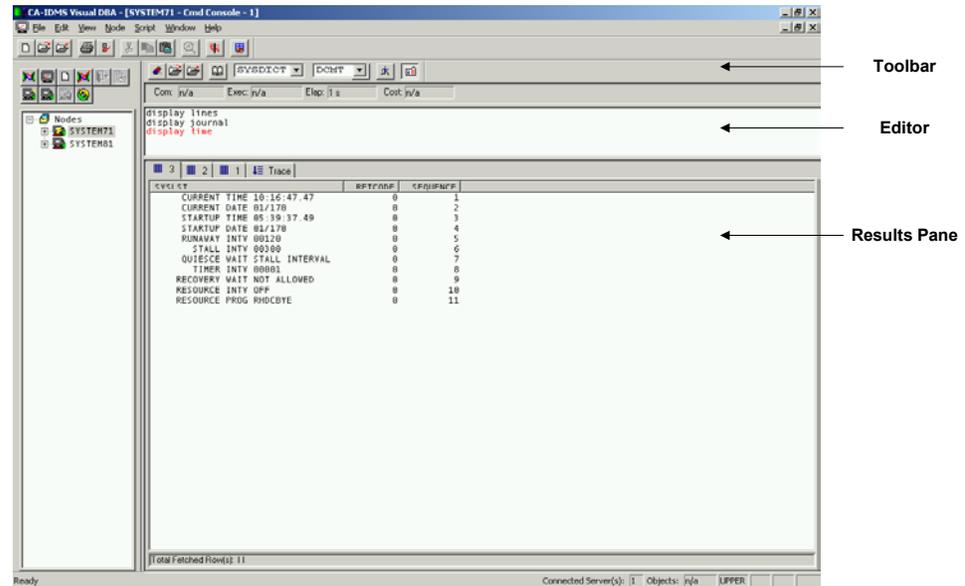
CA-IDMS Visual DBA includes a Command Console that enables you to edit, execute, and view the results of commands and scripts for the following CA-IDMS command processors:

- DCMT
- DCUF
- IDD
- OCF
- Schema
- SQL
- SSC
- Sysgen

Using the Command Console Workspace



To open the Command Console, click the Command Console button, included on the Nodes Window toolbar, or select Cmd Console from the Node Menu. As shown below, the Command Console Workspace includes a number of features that simplify working with CA-IDMS command processors.



Command Console Toolbar

As described below, the Command Console Toolbar enables you to perform the following functions:



Button	Function
	Clears the contents of the Command Console Editor.
	Opens a saved script in the Command Console editor of the active Command Console. Choosing the Open Script command invokes the Open Script dialog box where you can choose the script that you want to open.

Button	Function
	Saves the contents of the Command Console editor. Choosing the Save Script command invokes the Save As (script) dialog box where you can choose a file name and directory for the script.
	Invokes the SQL Assistant. Note: Before using this command, select the appropriate dictionary in the Select Dictionary drop-down list box.
	Enables you to select the dictionary to be used in executing scripts. You must choose a dictionary for all command processor except DCMT and DCUF.
	Enables you to select the command processor to invoke when the Go button is clicked or F5 is pressed. Selecting the command processor also determines which file extension is used when opening or saving command scripts.
	Runs the commands or script that is displayed in the command console editor. An alternative to clicking the Go button is using the F5 key on the keyboard.
	Toggles the display of the Trace tab in the Command Console Result pane on and off.

Command Console Editor

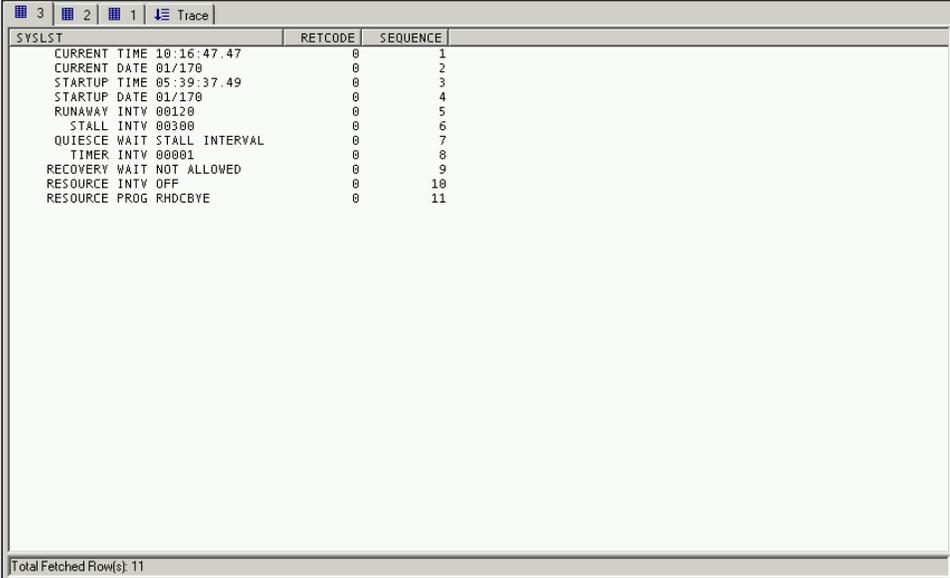
The Command Console Editor displays the active command script. You can create a new script by typing commands directly into the Editor, or you can open an existing script and edit it there. CA-IDMS Visual DBA provides the standard Windows editing commands Cut, Copy, and Paste from both the Workspace Toolbar and the Edit Menu. The Editor window is scrollable and contains the entire command script.

When you open a Results tab (described below), the source command for that tab is highlighted in red in the Editor window as shown in the example below.

```
display lines
display journal
display time
```

Command Console Results Pane

The Command Console Results pane displays the results of executing scripts.



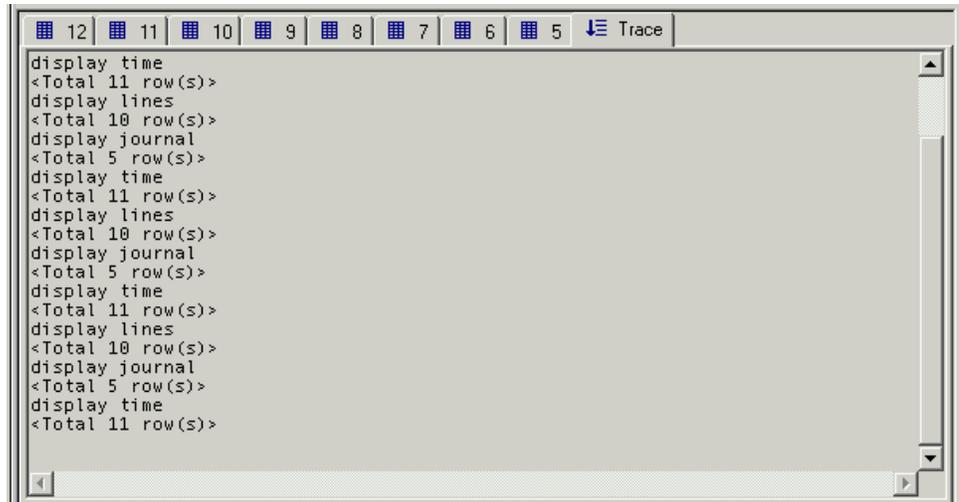
SYSLSST	RETCODE	SEQUENCE
CURRENT TIME 10:16:47.47	0	1
CURRENT DATE 01/170	0	2
STARTUP TIME 05:39:37.49	0	3
STARTUP DATE 01/170	0	4
RUNAWAY INTV 00120	0	5
STALL INTV 00300	0	6
QUIESCE WAIT STALL INTERVAL	0	7
TIMER INTV 00001	0	8
RECOVERY WAIT NOT ALLOWED	0	9
RESOURCE INTV OFF	0	10
RESOURCE PROG RHDCBYE	0	11

Total Fetched Row(s): 11

It includes numbered tabs that display the execution results in sequential order, with the highest number representing the most recently executed command. Click on a tab to review the results that it represents. The number of tabs that are displayed in the Results pane is based on the value that you define in the Preferences dialog. See [Setting Command Console Preferences](#) later in this chapter for more information.



The Results pane also includes a trace window. The Trace tab maintains a trace of the execute commands.



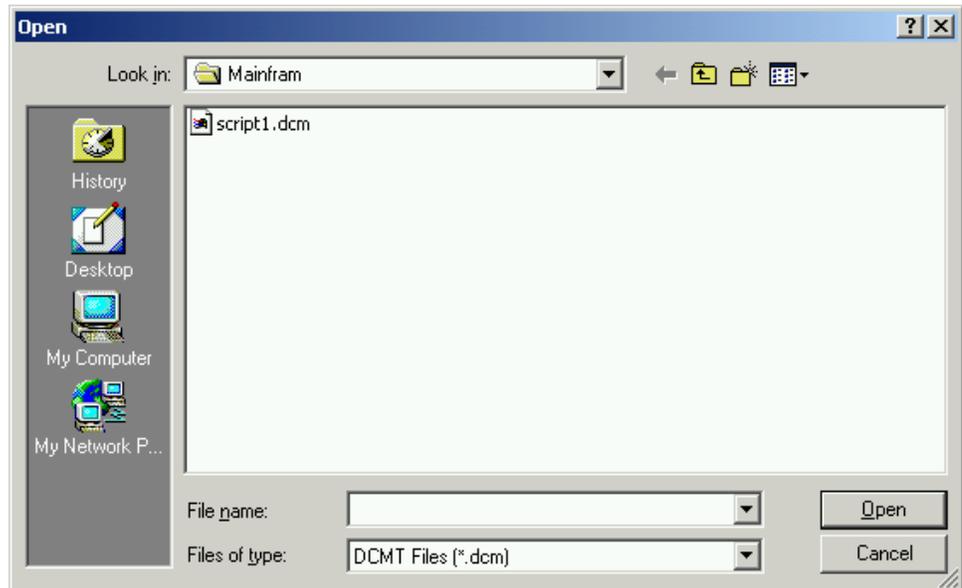
You can toggle the display of the Trace tab on and off by clicking the Trace button on the Command Console Toolbar or by selecting and deselecting the Trace Window command on the Script Menu.

Running a Command Script

To run a command script:

1. Select a Processor by choosing a value from the Select Processor dropdown list.
2. If you are running a script for the SQL, OCF, IDD, Schema, Sysgen, or SSC processors, you must select a dictionary from the Select Dictionary dropdown list.
3. If you are creating a new command script, you can type your text directly into the Command Console Editor.

4. If you are running an existing script, click the Open button  on the Command Console Toolbar or select the Open Script command on the Script Menu. When you do so, CA-IDMS Visual DBA displays the Open dialog where you can choose the script that you want to run.



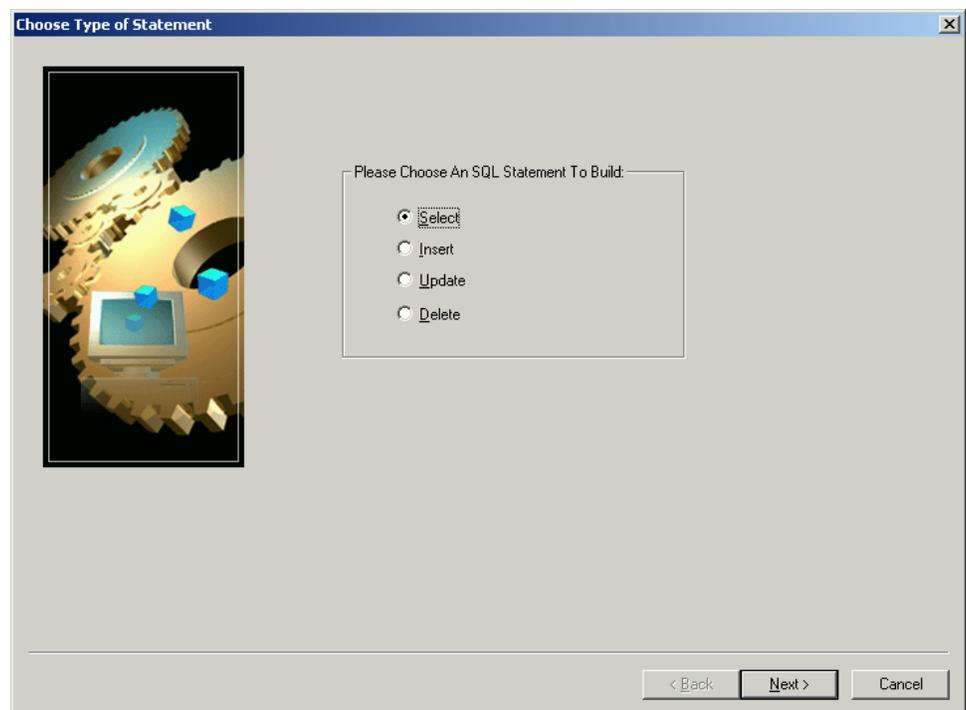
Note that the file extension that is used in the Open dialog is based on the Processor type that you selected in Step 1.

5. Click the Go button  on the Command Console Toolbar, select the Go command from the Script Menu, or press F5 to execute the command script.
6. Review your results in the Results pane.
7. If you have created a new command script or edited an existing script, you can save your script file by clicking the Save button  on the Command Console Toolbar or selecting the Save Script command on the Script Menu. When you do so, CA-IDMS Visual DBA displays the Save dialog where you can save the script and execute it at a later time.

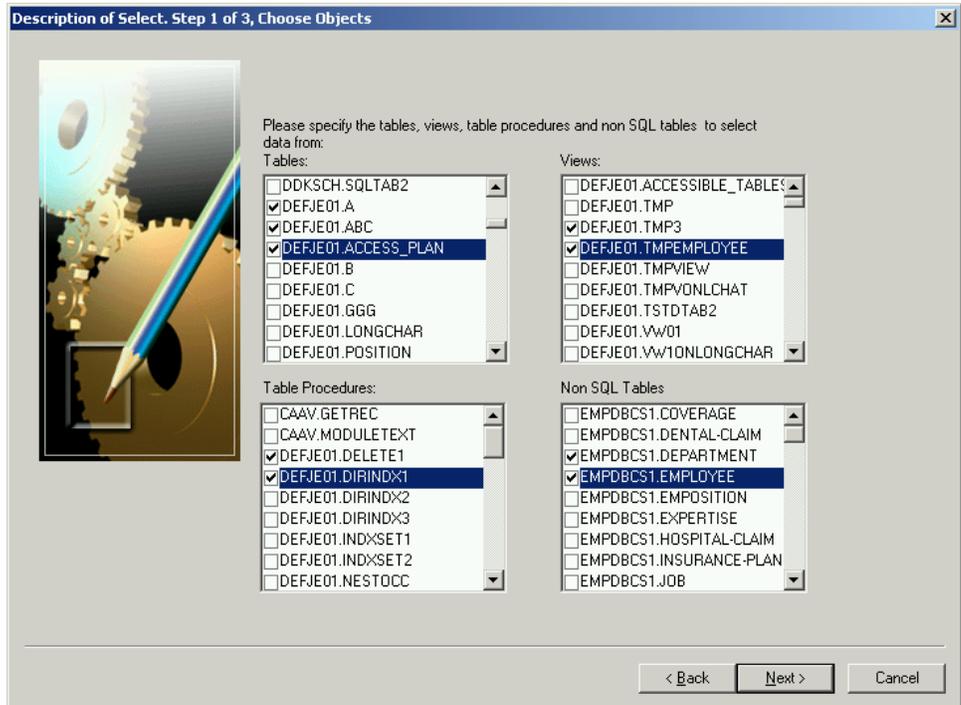
Using the SQL Assistant

The SQL Assistant is a Wizard-driven tool that automates the process of creating an SQL script. Through a series of dialogs you choose the type of command that you want to

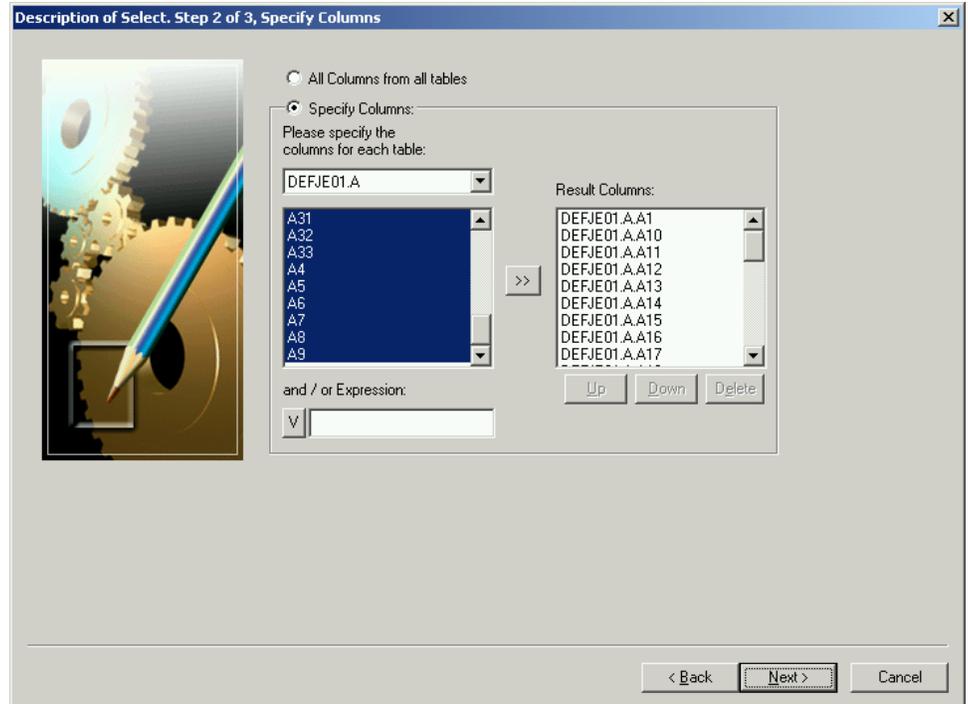
1. Select a dictionary from the Select Dictionary dropdown list.
2. Start the SQL Assistant by clicking the SQL Assistant button  on the Command Console Toolbar or by selecting the SQL Assistant command from the Script Menu.
3. The Choose Type of Statement dialog opens, enabling you to pick the type of SQL command statement that you want to execute.



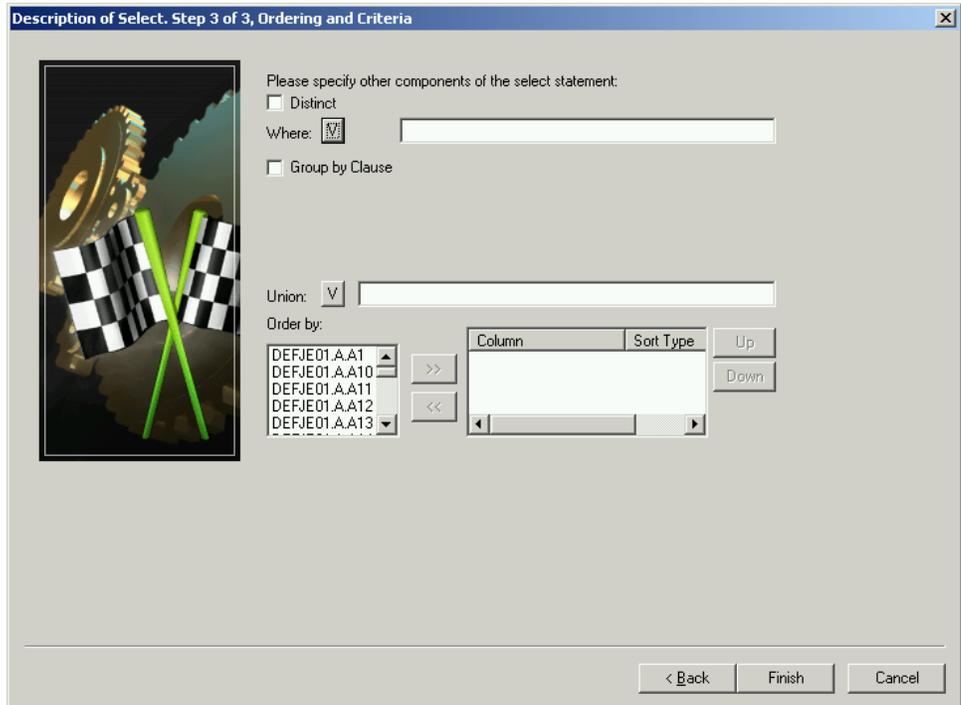
4. When you make your selection and click Next, CA-IDMS Visual DBS displays the first of three Choose Objects dialogs. Here you can select the Tables, Table Procedures, Views, and Non SQL Tables that you want to use as objects.



- When you click Next, CA-IDMS Visual DBA displays a dialog where you specify the columns that you want to include in your results. Click Next to continue.



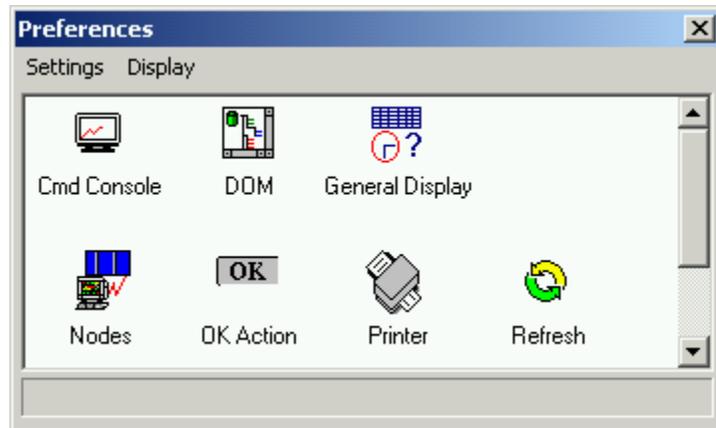
- Next, the Ordering and Criteria dialog opens where you can choose the order of the columns and the sort type (ascending or descending) for the column. When you click Finish, the SQL statements execute and the results are displayed in the Results pane.



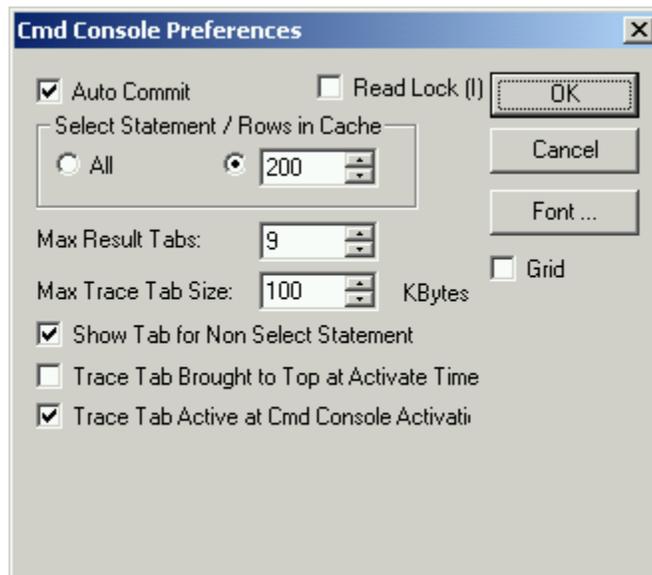
Setting Command Console Preferences

To set preferences for the Command Console:

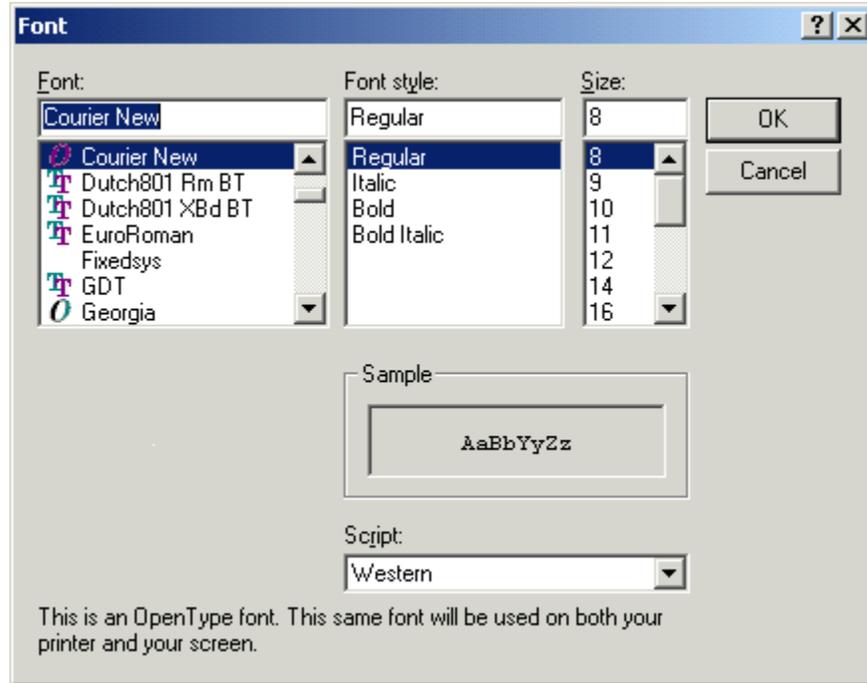
1. Click the Preferences button  on the Workspace Toolbar or select the Preferences command from the File Menu. Both actions open the Preferences dialog.



2. Double click on the Cmd Console icon. When you do so, the Cmd Console Preferences dialog opens where you can:
 - Select trace options
 - Define how many records CA-IDMS Visual DBA should keep in memory. When the cache becomes full, retrieval is suspended until a scroll down is requested. Following a scroll down request, the first half of the cache is emptied and fetching is resumed. Fetching will continue to function until the cache becomes full once again.
 - Specify how many history tabs should be provided
 - Set an indicator that determines whether changes made to the data during the execution of an SQL query should be automatically committed
 - Choose the display font for the Command Console



3. If you click the Font button, the following dialog opens where you can choose the font, font style, and point size of the text used in the Command Console.



Security TABL Resource Tables

Overview

The tables in this appendix contain the information you need to secure the database objects that you view through CA-IDMS Visual DBA. For more information about securing objects, see Chapter 2, “Installing and Starting CA-IDMS Visual DBA.”

Tables 1 through 4 show the relations between the CA-IDMS Visual DBA objects and the TABL resources. Each table contains:

- The objectname, which is the label of the object as used in the tree. The object names are not unique.
- The objectid, which uniquely identifies an object. For example, objectids GP_GR_SCHEMA and GP_US_SCHEMA both have the same objectname, Schema granted.GP_GR_SCHEMA is the schema granted for a Group, while GP_US_SCHEMA is the schema granted for a User.

The remainder of this appendix displays the contents of tables 1 through 4. These tables contain:

Table	Contents
Table 1	Objectname, the level in the tree, and the objectid. The objectnames are listed in the order of the tree, starting with level 0 to level 5.
Table 2	Objectnames ordered alphabetically, level and objectid.
Table 3	Objectname, the level in the tree, and the resourcenames (Schema.table) used by the object. The objectnames are in alphabetic order.
Table 4	Resourcenames (Schema.table) used by the object, the level, and the objectname. The resourcenames are in alphabetic order.

Table 1: Objectnames in Tree Order

Objectname	Level	ObjectId
Dictionary	0	DATABASE
SQL Schema	1	SCHEMA
Table	1	ALL_TABLE
View	1	ALL_VIEW
Table Procedure	1	ALL_TABLEPROC
Procedure	1	ALL_PROCEDURE
Non SQL Table	1	ALL_TABLENSQL
Relational Command Module	1	RCM
Non SQL Schema	1	NSQLSCHEMA
IDD Class	1	CLASS
IDD Record	1	IDDRECORD
Segment	1	SEGMENT
DBTable	1	DBTABLE
DMCL	1	DMCL
System	1	SYSTEM
Category	1	CATEGORY
Activity	1	ACTIVITY
System Id	1	SYSTEMID
System Profile	1	SYSPROFILE
User Profile	1	POFILE
Group	1	GROUP
Central User	1	USER
Dictionary User	1	DICTUSER
Grantee for DCADMIN	1	GU_DCADMIN
Grantee for SYSADMIN	1	GU_SYSADMIN
Table in Schema	2	TABLE
View in Schema	2	VIEW
Table Procedure in Schema	2	TABLEPROC
Procedure in Schema	2	PROCEDURE

Table 1: Objectnames in Tree Order

Objectname	Level	Objectid
Non SQL Table in Schema	2	TABLENSQL
Constraint	2	CONSTRAINT
Access Module	2	AM
Grantee for Schema	2	GU_SCHEMA
Area in Non SQL Schema	2	NSQLAREA
Record	2	RECORDTYP
Set	2	NSQLSET
Subschema	2	SUBSCHEMA
Class / Attribute	2	NSQLATTRIB
Comment	2	NSQLCOMMENT
Grantee for Use	2	GU_NSQLSCHEMA
Registree for Schema	2	RU_NSQLSCHEMA
Responsible for Schema	2	SU_NSQLSCHEMA
Attribute	2	ATTRIBUTE
IDD Record Synonym	2	IDDRECORDSYN
IDD Record Element	2	IDDRECORDELEM
File	2	FILE
Area	2	AREA
DMCL including Segment	2	SEGMENT_DMCL
DBName including Segment	2	SEGMENT_DBNAME
Schema referencing Segment	2	SEGMENT_SCHEMA
Grantee for Segment	2	GU_SEGMENT
Subschema Mapping	2	SSCMAP
DBName	2	DBNAME
DMCL using DBTable	2	DBTABLE_DMCL
DBGroup in DBTable	2	DBGROUP
Grantee for DBTable	2	GU_DBTABLE
Segment in DMCL	2	DMCL_SEGMENT
Segment.File Override	2	DMCL_SEGMENT_FILE_OVER
Segment.Area Override	2	DMCL_SEGMENT_AREA_OVER

Table 1: Objectnames in Tree Order

Objectname	Level	Objectid
File in DMCL	2	DMCL_FILE
Area in DMCL	2	DMCL_AREA
Shared File in DMCL	2	DMCL_DSH_FILE
Shared Area in DMCL	2	DMCL_DSH_AREA
Shared Cache in DMCL	2	DMCL_DSH_CACHE
Database Buffer	2	BUFFER_DB
Journal Buffer	2	BUFFER_JOURNAL
Archive Journal	2	JOURNAL_ARCH
Disk Journal	2	JOURNAL_DISK
Tape Journal	2	JOURNAL_TAPE
Grantee for DMCL	2	GU_DMCL
Load List	2	LOADLIST
Program	2	PROGRAM
Queue	2	QUEUE
Autotask	2	AUTOTASK
Task	2	TASK
Storage Pool	2	STORPOOL
XA Storage Pool	2	XASTORPOOL
Line	2	LINE
Lterm	2	SYSTEM_LTERM
Printer	2	SYSTEM_PRINTER
Destination	2	DESTINATION
Node	2	DCNODE
Local Dictionary in Res Table	2	RESTABLE_DBNAME_LOCAL
Dictionary via Node in Res Table	2	RESTABLE_DBNAME_VIA
Destination Node in Res Table	2	RESTABLE_DESTNODE
Rununit	2	RUNUNIT
Map Table	2	MAPTYPE
Key Table	2	KEYTABLE
Access Module in Category	2	CATEGORY_AM

Table 1: Objectnames in Tree Order

Objectname	Level	Objectid
Load Module in Category	2	CATEGORY_LOADMODULE
Program in Category	2	CATEGORY_PROGRAM
Queue in Category	2	CATEGORY_QUEUE
Rununit in Category	2	CATEGORY_RUNUNIT
Task in Category	2	CATEGORY_TASK
Grantee for Execute	2	GU_CATEGORY
Grantee for Execute	2	GU_ACTIVITY
Grantee for System Id	2	GU_SYSTEMID
Attribute	2	SYSATTRIB
Grantee on System Profile	2	GU_SYSPROFILE
Attribute	2	ATTRIB
Grantee on User Profile	2	GU_PROFILE
User in Group	2	GROUP_USER
Access Module granted	2	GP_GR_AM
Activity granted	2	GP_GR_ACTIVITY
Area granted	2	GP_GR_AREA
Category granted	2	GP_GR_CATEGORY
DBName & Segment granted	2	GP_GR_DBNAME
DBTable granted	2	GP_GR_DBTABLE
DCADMIN granted ?	2	GP_GR_DCADMIN
DMCL granted	2	GP_GR_DMCL
Group granted	2	GP_GR_GROUP
Non SQL Schema granted	2	GP_GR_NSQLSHEMA
Non SQL Table granted	2	GP_GR_TABLENSQL
Schema granted	2	GP_GR_SCHEMA
SYSADMIN granted ?	2	GP_GR_SYSADMIN
System Id granted	2	GP_GR_SYSTEMID
System Profile granted	2	GP_GR_SYSPROFILE
Table granted	2	GP_GR_TABLE
Table Procedure granted	2	GP_GR_TABLEPROC

Table 1: Objectnames in Tree Order

Objectname	Level	Objectid
Procedure granted	2	GP_GR_PROCEDURE
User granted	2	GP_GR_USER
User Profile granted	2	GP_GR_PROFILE
View granted	2	GP_GR_VIEW
Grantee on Group	2	GU_GROUP
Group User belongs	2	USER_GROUP
Access Module granted	2	GP_US_AM
Activity granted	2	GP_US_ACTIVITY
Area granted	2	GP_US_AREA
Category granted	2	GP_US_CATEGORY
DBName & Segment granted	2	GP_US_DBNAME
DBTable granted	2	GP_US_DBTABLE
DCADMIN granted ?	2	GP_US_DCADMIN
DMCL granted	2	GP_US_DMCL
Group granted	2	GP_US_GROUP
Non SQL Schema granted	2	GP_US_NSQLSHEMA
Non SQL Table granted	2	GP_US_TABLENSQL
Schema granted	2	GP_US_SCHEMA
SYSADMIN granted	2	GP_US_SYSADMIN
System Id granted	2	GP_US_SYSTEMID
System Profile granted	2	GP_US_SYSPROFILE
Table granted	2	GP_US_TABLE
Table Procedure granted	2	GP_US_TABLEPROC
Procedure granted	2	GP_US_PROCEDURE
User granted	2	GP_US_USER
User Profile granted	2	GP_US_PROFILE
View granted	2	GP_US_VIEW
Grantee on User	2	GU_USER
Entity Type Authority	2	GP_DU
Non SQL Schema registered	2	RP_DU_NSQLSHEMA

Table 1: Objectnames in Tree Order

Objectname	Level	ObjectId
Subschema registered	2	RP_DU_SUBSCHEMA
Non SQL Schema responsibility	2	SP_DU_NSQLSHEMA
Subschema responsibility	2	SP_DU_SUBSCHEMA
Index	3	INDEX
View on Table	3	TABLE_VIEW
Constraint Table referenced	3	TABLE_CONST_REFCED
Constraint Table referencing	3	TABLE_CONST_REFING
Access Module for Table	3	TABLE_AM
Calc Key	3	CALC
Column	3	COLUMN
Check Condition	3	CHECKCONDITION
Grantee on Table	3	GU_TABLE
View Component	3	VIEW_COMPONENT
Access Module for View	3	VIEW_AM
Column of View	3	VIEWCOLUMN
View Definition	3	VIEWDEFINITION
Grantee on View	3	GU_VIEW
Key in Table Procedure	3	TABLEPROCKEY
Parameter of Table Procedure	3	TABLEPROCCOLUMN
Access Module for Table Procedure	3	TABLEPROC_AM
Grantee on Table Procedure	3	GU_TABLEPROC
Key in Procedure	3	PROCKEY
Parameter of Procedure	3	PROCCOLUMN
Access Module for Procedure	3	PROCEDURE_AM
Grantee on Procedure	3	GU_PROCEDURE
Column of non SQL Table	3	TABLENSQLCOLUMN
Access Module using non SQL Table	3	TABLENSQL_AM
Grantee on non SQL Table	3	GU_TABLENSQL
Referencing Table in Constraint	3	CONST_TABLE_REFING
Referenced Table in Constraint	3	CONST_TABLE_REFCED

Table 1: Objectnames in Tree Order

Objectname	Level	Objectid
Relational Command Module in AM	3	AM_RCM
Schema Mapping of AM	3	AM_SCHEMA_MAP
Area with Ready Mode in AM	3	AM_AREA_OPT
Table accessed	3	AM_TABLE
View accessed	3	AM_VIEW
Table Procedure accessed	3	AM_TABLEPROC
Procedure accessed	3	AM_PROCEDURE
Non SQL Table accessed	3	AM_TABLENSQL
Grantee on Access Module	3	GU_AM
Area Procedure	3	AREAPROC
Record in Area	3	AREA_RECORDTYP
Record Procedure	3	RECORDPROC
Data Compression Table	3	DATACOMPTAB
Record Synonym	3	RECORDSYN
Record Element	3	RECORDELEM
Set owned by Record	3	RECORDTYP_SET_OWNING
Set Record is member	3	RECORDTYP_SET_MEMBER
Structure Shared By	3	RECORDSHARE
Owner Record of Set	3	SET_RECORD_OWNER
Member Record of Set	3	SET_RECORD_MEMBER
Area in Subschema	3	SUBSCHEMA_AREA
Record in Subschema	3	SUBRECORD
Set in Subschema	3	SUBSCHEMA_SET
Load Module	3	LOADMODULE
Program using Subschema	3	SUBSCHEMA_PROGRAM
Class / Attribute	3	SUBSCHEMAATTRIB
Comment	3	SUBSCHEMACOMMENT
Registree for Subschema	3	RU_SUBSCHEMA
Responsible for Subschema	3	SU_SUBSCHEMA
IDD Element Synonym	3	IDDELEMSYN

Table 1: Objectnames in Tree Order

Objectname	Level	Objectld
IDD Element Description	3	IDDELEMSDES
IDD Element Value	3	IDDELEMVALUE
IDD Element Indexed By	3	IDDNAMEDES
Area in File	3	FILE_AREA
DMCL using File	3	FILE_DMCL
File in Area	3	AREA_FILE
DMCL using Area	3	AREA_DMCL
Schema using Area as default	3	AREA_SCHEMA
Table stored in Area	3	AREA_TABLE
Index stored in Area	3	AREA_INDEX
Subarea	3	SYMBOL_SUBAREA
Displacement	3	SYMBOL_DISPLACEMENT
Index	3	SYMBOL_INDEX
Grantee on Area	3	GU_AREA
Segment in DBName	3	DBNAME_SEGMENT
Subschema Mapping in DBName	3	DBNAME_SSCMAP
Schema referencing DBName	3	DBNAME_SCHEMA
DBTable including DBName	3	DBNAME_DBTABLE
Grantee for DBName	3	GU_DBNAME
Component of Loadlist	3	LOADLIST_COMPONENT
Pterm	3	PTERM
Lterm in Line	3	LTERM
Lterm in Destination	3	DESTLTERM
Printer in Destination	3	PRINTER
User in Destination	3	DESTINATION_DICTUSER
Entry of Map Table	3	MAPTYPE_ENTRY
Application	3	APPLICATION
Element Synonym	4	ELEMSYN
Element Description	4	ELEMSDES
Element Value	4	ELEMVALUE

Table 1: Objectnames in Tree Order

Objectname	Level	Objectld
Element Indexed By	4	NAMEDES
Record Procedure	4	MBR_RECORDPROC
Data Compression Table	4	MBR_DATACOMPTAB
Record Synonym	4	MBR_RECORDSYN
Record Element	4	MBR_RECORDELEM
Set owned by Record	4	MBR_RECORDTYP_SET_OWNING
Set Record is member	4	MBR_RECORDTYP_SET_MEMBER
Record Control Key	4	SET_REC_CONTROLKEY
Record Foreign Key	4	SET_REC_FOREIGNKEY
Element of Subschema Record	4	SUBRECORDELEM
Area used in Program	4	SUBPROGAREA
Record used in Program	4	SUBPROGRECORD
Set used in Program	4	SUBPROGSET
Lterm in Pterm	4	PTERM_LTERM
Area Statistics for Program	5	AREASTATISTICS
Record in Area	5	SUBPROGAREA_RECORD
Record Statistics for Program	5	RECORDSTATISTICS
Element of Subschema Record	5	SUBPROGRECORD_ELEM
Set Statistics for Program	5	SETSTATISTICS
Owner Record of Set	5	SUBPROGSET_OWNER
Member Record of Set	5	SUBPROGSET_MEMBER
CV Subschema Mapping	0	CV_SSCMAP
CV DBName	0	CV_DBNAME
CV DBGroup	0	CV_DBGROUP
CV Segment	0	CV_SEGMENT
CV File	0	CV_FILE
CV Area	0	CV_AREA
CV Database Buffer	0	CV_BUFFER_DB
CV Journal Buffer	0	CV_BUFFER_JOURNAL
CV Journal	0	CV_JOURNAL

Table 1: Objectnames in Tree Order

Objectname	Level	Objectid
CV Load List	0	CV_LOADLIST
CV Program	0	CV_PROGRAM
CV Program Pool new Sysgen	0	CV_SYSGEN_PROGPOOL
CV Reent Prog Pool new Sysgen	0	CV_SYSGEN_REPROGPOOL
CV Queue	0	CV_QUEUE
CV Task	0	CV_TASK
CV Storage Pool	0	CV_STORPOOL
CV XA Storage Pool	0	CV_XASTORPOOL
CV XA Storage Pool new Sysgen	0	CV_SYSGEN_XASTORPOOL
CV Line	0	CV_LINE
CV Line new Sysgen	0	CV_SYSGEN_LINE
CV Pterm	0	CV_SYSTEM_PTERM
CV Lterm	0	CV_SYSTEM_LTERM
CV Printer	0	CV_SYSTEM_PRINTER
CV Destination	0	CV_DESTINATION
CV Node	0	CV_DCNODE
CV Resource Table	0	CV_RESTABLE
CV Rununit	0	CV_RUNUNIT
Active Program	0	CV_ACT_PROGPOOL
Active Storage	0	CV_ACT_STORPOOL
Active Task	0	CV_ACT_TASK
User signed on	0	CV_USER
Data Sharing Summary	0	CV_DATASHARING
Data Sharing Group Member	0	CV_DATASHARING_XCF_GRP_MBR
Data Sharing LIST Structure	0	CV_DATASHARING_XES_LIST_DET
Data Sharing LOCK Structure	0	CV_DATASHARING_XES_LOCK_DET
Deadlock	0	CV_DEADLOCK
DCMT Command Outstanding	0	CV_DCMT_CMD
External Rununit	0	CV_EXT_RUNUNIT
Journal Status	0	CV_JOURNAL_STAT

Table 1: Objectnames in Tree Order

Objectname	Level	ObjectId
Lock on Area	0	CV_LOCK_AREA
Lock on Lterm	0	CV_LOCK_LTERM
Log	0	CV_LOG
Log Driver	0	CV_LOG_DRIVER
Shared Cache	0	CV_SHAREDCACHE
Transaction	0	CV_TRANSACTION
ADSO	0	CV_ADSO
DDS	0	CV_DDS
Limit	0	CV_LIMIT
Loadlib	0	CV_LOADLIB
LU	0	CV_LU
MPMode	0	CV_MPMODE
MT Queue Depth	0	CV_MT_QUEUE_DEPTH
Nucleus Reload	0	CV_NUCLEUS
Reply pending	0	CV_REPLY
Report Class / Destination	0	CV_REPORT_CLASS_DEST
SNA Pterm	0	CV_SNA_PTERM
Snap	0	CV_SNAP
SubTasks	0	CV_SUBTASK
Time	0	CV_TIME
UCF Terminal	0	CV_UCF
Statistics Segment	0	CV_STAT_SEGMENT
Statistics File	0	CV_STAT_FILE
Statistics Area	0	CV_STAT_AREA
Statistics Buffer	0	CV_STAT_BUFFER
Statistics Interval	0	CV_STAT_INTERVAL
Statistics Lock	0	CV_STAT_LOCK
Statistics System	0	CV_STAT_SYSTEM
CV Segment in DBName	1	CV_DBNAME_SEGMENT
CV Subschema Mapping in DBName	1	CV_DBNAME_SSCMAP

Objectname	Level	ObjectId
CV Backend in DBGroup	1	CV_DBGROUP_BACKEND
CV File in Segment	1	CV_SEGMENT_FILE
CV Area in Segment	1	CV_SEGMENT_AREA
CV DBName including Segment	1	CV_SEGMENT_DBNAME
CV PTerm in Line	1	CV_LINE_PTERM
CV LTerm in Line	1	CV_LINE_LTERM
CV Rununit Detail	1	CV_RUNUNIT_DET
Active Program Details	1	CV_ACT_PROGPOOL_DET
Active Storage Details	1	CV_ACT_STORPOOL_DET
Active Task Details	1	CV_ACT_TASK_DET
File in Shared Cache	1	CV_SHAREDCACHE_FILE
Modeent of LU	1	CV_LU_MODEENT
Report	1	CV_REPORT
Time initiated Task	1	CV_TIME_TASK
Statistics File in Segment	1	CV_STAT_SEGMENT_FILE
Statistics Area in Segment	1	CV_STAT_SEGMENT_AREA

Table 2: Objectnames in Alphabetical Order

Objectname	Level	ObjectId
ADSO	0	CV_ADSO
Access Module	2	AM
Access Module for Procedure	3	PROCEDURE_AM
Access Module for Table	3	TABLE_AM
Access Module for Table Procedure	3	TABLEPROC_AM
Access Module for View	3	VIEW_AM
Access Module granted	2	GP_GR_AM
Access Module granted	2	GP_US_AM
Access Module in Category	2	CATEGORY_AM

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectid
Access Module using non SQL Table	3	TABLENSQL_AM
Active Program	0	CV_ACT_PROGPOOL
Active Program Details	1	CV_ACT_PROGPOOL_DET
Active Storage	0	CV_ACT_STORPOOL
Active Storage Details	1	CV_ACT_STORPOOL_DET
Active Task	0	CV_ACT_TASK
Active Task Details	1	CV_ACT_TASK_DET
Activity	1	ACTIVITY
Activity granted	2	GP_GR_ACTIVITY
Activity granted	2	GP_US_ACTIVITY
Application	3	APPLICATION
Archive Journal	2	JOURNAL_ARCH
Area	2	AREA
Area Procedure	3	AREAPROC
Area Statistics for Program	5	AREASTATISTICS
Area granted	2	GP_GR_AREA
Area granted	2	GP_US_AREA
Area in DMCL	2	DMCL_AREA
Area in File	3	FILE_AREA
Area in Non SQL Schema	2	NSQLAREA
Area in Subschema	3	SUBSCHEMA_AREA
Area used in Program	4	SUBPROGAREA
Area with Ready Mode in AM	3	AM_AREA_OPT
Attribute	2	ATTRIB
Attribute	2	ATTRIBUTE
Attribute	2	SYSATTRIB
Autotask	2	AUTOTASK
CV Area	0	CV_AREA
CV Area in Segment	1	CV_SEGMENT_AREA
CV Backend in DBGroup	1	CV_DBGROUP_BACKEND

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectld
CV DBGroup	0	CV_DBGROUP
CV DBName	0	CV_DBNAME
CV DBName including Segment	1	CV_SEGMENT_DBNAME
CV Database Buffer	0	CV_BUFFER_DB
CV Destination	0	CV_DESTINATION
CV File	0	CV_FILE
CV File in Segment	1	CV_SEGMENT_FILE
CV Journal	0	CV_JOURNAL
CV Journal Buffer	0	CV_BUFFER_JOURNAL
CV LTerm in Line	1	CV_LINE_LTERM
CV Line	0	CV_LINE
CV Line new Sysgen	0	CV_SYSGEN_LINE
CV Load List	0	CV_LOADLIST
CV Lterm	0	CV_SYSTEM_LTERM
CV Node	0	CV_DCNODE
CV PTerm in Line	1	CV_LINE_PTERM
CV Printer	0	CV_SYSTEM_PRINTER
CV Program	0	CV_PROGRAM
CV Program Pool new Sysgen	0	CV_SYSGEN_PROGPOOL
CV Pterm	0	CV_SYSTEM_PTERM
CV Queue	0	CV_QUEUE
CV Reent Prog Pool new Sysgen	0	CV_SYSGEN_REPROGPOOL
CV Resource Table	0	CV_RESTABLE
CV Rununit	0	CV_RUNUNIT
CV Rununit Detail	1	CV_RUNUNIT_DET
CV Segment	0	CV_SEGMENT
CV Segment in DBName	1	CV_DBNAME_SEGMENT
CV Storage Pool	0	CV_STORPOOL
CV Subschema Mapping	0	CV_SSCMAP
CV Subschema Mapping in DBName	1	CV_DBNAME_SSCMAP

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectid
CV Task	0	CV_TASK
CV XA Storage Pool	0	CV_XASTORPOOL
CV XA Storage Pool new Sysgen	0	CV_SYSGEN_XASTORPOOL
Calc Key	3	CALC
Category	1	CATEGORY
Category granted	2	GP_GR_CATEGORY
Category granted	2	GP_US_CATEGORY
Central User	1	USER
Check Condition	3	CHECKCONDITION
Class / Attribute	2	NSQLATTRIB
Class / Attribute	3	SUBSCHEMAATTRIB
Column	3	COLUMN
Column of View	3	VIEWCOLUMN
Column of non SQL Table	3	TABLENSQLCOLUMN
Comment	2	NSQLCOMMENT
Comment	3	SUBSCHEMACOMMENT
Component of Loadlist	3	LOADLIST_COMPONENT
Constraint	2	CONSTRAINT
Constraint Table referenced	3	TABLE_CONST_REFCED
Constraint Table referencing	3	TABLE_CONST_REFING
DBGroup in DBTable	2	DBGROUP
DBName	2	DBNAME
DBName & Segment granted	2	GP_GR_DBNAME
DBName & Segment granted	2	GP_US_DBNAME
DBName including Segment	2	SEGMENT_DBNAME
DBTable	1	DBTABLE
DBTable granted	2	GP_GR_DBTABLE
DBTable granted	2	GP_US_DBTABLE
DBTable including DBName	3	DBNAME_DBTABLE
DCADMIN granted	2	GP_GR_DCADMIN

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectid
DCADMIN granted	2	GP_US_DCADMIN
DCMT Command Outstanding	0	CV_DCMT_CMD
DDS	0	CV_DDS
DMCL	1	DMCL
DMCL granted	2	GP_GR_DMCL
DMCL granted	2	GP_US_DMCL
DMCL including Segment	2	SEGMENT_DMCL
DMCL using Area	3	AREA_DMCL
DMCL using DBTable	2	DBTABLE_DMCL
DMCL using File	3	FILE_DMCL
Data Compression Table	3	DATACOMPTAB
Data Compression Table	4	MBR_DATACOMPTAB
Data Sharing Group Member	0	CV_DATASHARING_XCF_GRP_MBR
Data Sharing LIST Structure	0	CV_DATASHARING_XES_LIST_DET
Data Sharing LOCK Structure	0	CV_DATASHARING_XES_LOCK_DET
Data Sharing Summary	0	CV_DATASHARING
Database Buffer	2	BUFFER_DB
Deadlock	0	CV_DEADLOCK
Destination	2	DESTINATION
Destination Node in Res Table	2	RESTABLE_DESTNODE
Dictionary	0	DATABASE
Dictionary User	1	DICTUSER
Dictionary via Node in Res Table	2	RESTABLE_DBNAME_VIA
Disk Journal	2	JOURNAL_DISK
Displacement	3	SYMBOL_DISPLACEMENT
Element Description	4	ELEMSDES
Element Indexed By	4	NAMEDES
Element Synonym	4	ELEMSYN
Element Value	4	ELEMVALUE
Element of Subschema Record	4	SUBRECORDELEM

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectid
Element of Subschema Record	5	SUBPROGRECORD_ELEM
Entity Type Authority	2	GP_DU
Entry of Map Table	3	MAPTYPE_ENTRY
External Rununit	0	CV_EXT_RUNUNIT
File	2	FILE
File in Area	3	AREA_FILE
File in DMCL	2	DMCL_FILE
File in Shared Cache	1	CV_SHARED_CACHE_FILE
Grantee for DBName	3	GU_DBNAME
Grantee for DBTable	2	GU_DBTABLE
Grantee for DCADMIN	1	GU_DCADMIN
Grantee for DMCL	2	GU_DMCL
Grantee for Execute	2	GU_ACTIVITY
Grantee for Execute	2	GU_CATEGORY
Grantee for SYSADMIN	1	GU_SYSADMIN
Grantee for Schema	2	GU_SCHEMA
Grantee for Segment	2	GU_SEGMENT
Grantee for System Id	2	GU_SYSTEMID
Grantee for Use	2	GU_NSQLSHEMA
Grantee on Access Module	3	GU_AM
Grantee on Area	3	GU_AREA
Grantee on Group	2	GU_GROUP
Grantee on Procedure	3	GU_PROCEDURE
Grantee on System Profile	2	GU_SYSPROFILE
Grantee on Table	3	GU_TABLE
Grantee on Table Procedure	3	GU_TABLEPROC
Grantee on User	2	GU_USER
Grantee on User Profile	2	GU_PROFILE
Grantee on View	3	GU_VIEW
Grantee on non SQL Table	3	GU_TABLENSQL

Table 2: Objectnames in Alphabetical Order

Objectname	Level	ObjectId
Group	1	GROUP
Group User belongs	2	USER_GROUP
Group granted	2	GP_GR_GROUP
Group granted	2	GP_US_GROUP
IDD Class	1	CLASS
IDD Element Description	3	IDDELEMSDES
IDD Element Indexed By	3	IDDNAMEDES
IDD Element Synonym	3	IDDELEMSYN
IDD Element Value	3	IDDELEMVALUE
IDD Record	1	IDDRECORD
IDD Record Element	2	IDDRECORDELEM
IDD Record Synonym	2	IDDRECORDSYN
Index	3	INDEX
Index	3	SYMBOL_INDEX
Index stored in Area	3	AREA_INDEX
Journal Buffer	2	BUFFER_JOURNAL
Journal Status	0	CV_JOURNAL_STAT
Key Table	2	KEYTABLE
Key in Procedure	3	PROCKEY
Key in Table Procedure	3	TABLEPROCKEY
LU	0	CV_LU
Limit	0	CV_LIMIT
Line	2	LINE
Load List	2	LOADLIST
Load Module	3	LOADMODULE
Load Module in Category	2	CATEGORY_LOADMODULE
Loadlib	0	CV_LOADLIB
Local Dictionary in Res Table	2	RESTABLE_DBNAME_LOCAL
Lock on Area	0	CV_LOCK_AREA
Lock on Lterm	0	CV_LOCK_LTERM

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectid
Log	0	CV_LOG
Log Driver	0	CV_LOG_DRIVER
Lterm	2	SYSTEM_LTERM
Lterm in Destination	3	DESTLTERM
Lterm in Line	3	LTERM
Lterm in Pterm	4	PTERM_LTERM
MPMode	0	CV_MPMODE
MT Queue Depth	0	CV_MT_QUEUE_DEPTH
Map Table	2	MAPTYPE
Member Record of Set	3	SET_RECORD_MEMBER
Member Record of Set	5	SUBPROGSET_MEMBER
Modeent of LU	1	CV_LU_MODEENT
Node	2	DCNODE
Non SQL Schema	1	NSQLSCHEMA
Non SQL Schema granted	2	GP_GR_NSQLSCHEMA
Non SQL Schema granted	2	GP_US_NSQLSCHEMA
Non SQL Schema registered	2	RP_DU_NSQLSCHEMA
Non SQL Schema responsibility	2	SP_DU_NSQLSCHEMA
Non SQL Table	1	ALL_TABLENSQL
Non SQL Table accessed	3	AM_TABLENSQL
Non SQL Table granted	2	GP_GR_TABLENSQL
Non SQL Table granted	2	GP_US_TABLENSQL
Non SQL Table in Schema	2	TABLENSQL
Nucleus Reload	0	CV_NUCLEUS
Owner Record of Set	3	SET_RECORD_OWNER
Owner Record of Set	5	SUBPROGSET_OWNER
Parameter of Procedure	3	PROCCOLUMN
Parameter of Table Procedure	3	TABLEPROCCOLUMN
Printer	2	SYSTEM_PRINTER
Printer in Destination	3	PRINTER

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectid
Procedure	1	ALL_PROCEDURE
Procedure accessed	3	AM_PROCEDURE
Procedure granted	2	GP_GR_PROCEDURE
Procedure granted	2	GP_US_PROCEDURE
Procedure in Schema	2	PROCEDURE
Program	2	PROGRAM
Program in Category	2	CATEGORY_PROGRAM
Program using Subschema	3	SUBSCHEMA_PROGRAM
Pterm	3	PTERM
Queue	2	QUEUE
Queue in Category	2	CATEGORY_QUEUE
Record	2	RECORDTYP
Record Control Key	4	SET_REC_CONTROLKEY
Record Element	3	RECORDELEM
Record Element	4	MBR_RECORDELEM
Record Foreign Key	4	SET_REC_FOREIGNKEY
Record Procedure	3	RECORDPROC
Record Procedure	4	MBR_RECORDPROC
Record Statistics for Program	5	RECORDSTATISTICS
Record Synonym	3	RECORDSYN
Record Synonym	4	MBR_RECORDSYN
Record in Area	3	AREA_RECORDTYP
Record in Area	5	SUBPROGAREA_RECORD
Record in Subschema	3	SUBRECORD
Record used in Program	4	SUBPROGRECORD
Referenced Table in Constraint	3	CONST_TABLE_REFCED
Referencing Table in Constraint	3	CONST_TABLE_REFING
Registree for Schema	2	RU_NSQLSCHEMA
Registree for Subschema	3	RU_SUBSCHEMA
Relational Command Module	1	RCM

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectld
Relational Command Module in AM	3	AM_RCM
Reply pending	0	CV_REPLY
Report	1	CV_REPORT
Report Class / Destination	0	CV_REPORT_CLASS_DEST
Responsible for Schema	2	SU_NSQLSHEMA
Responsible for Subschema	3	SU_SUBSCHEMA
Rununit	2	RUNUNIT
Rununit in Category	2	CATEGORY_RUNUNIT
SNA Pterm	0	CV_SNA_PTERM
SQL Schema	1	SCHEMA
SYSADMIN granted ?	2	GP_GR_SYSADMIN
SYSADMIN granted ?	2	GP_US_SYSADMIN
Schema Mapping of AM	3	AM_SCHEMA_MAP
Schema granted	2	GP_GR_SCHEMA
Schema granted	2	GP_US_SCHEMA
Schema referencing DBName	3	DBNAME_SCHEMA
Schema referencing Segment	2	SEGMENT_SCHEMA
Schema using Area as default	3	AREA_SCHEMA
Segment	1	SEGMENT
Segment in DBName	3	DBNAME_SEGMENT
Segment in DMCL	2	DMCL_SEGMENT
Segment.Area Override	2	DMCL_SEGMENT_AREA_OVER
Segment.File Override	2	DMCL_SEGMENT_FILE_OVER
Set	2	NSQLSET
Set Record is member	3	RECORDTYP_SET_MEMBER
Set Record is member	4	MBR_RECORDTYP_SET_MEMBER
Set Statistics for Program	5	SETSTATISTICS
Set in Subschema	3	SUBSCHEMA_SET
Set owned by Record	3	RECORDTYP_SET_OWNING
Set owned by Record	4	MBR_RECORDTYP_SET_OWNING

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectid
Set used in Program	4	SUBPROGSET
Shared Area in DMCL	2	DMCL_DSH_AREA
Shared Cache	0	CV_SHARED_CACHE
Shared Cache in DMCL	2	DMCL_DSH_CACHE
Shared File in DMCL	2	DMCL_DSH_FILE
Snap	0	CV_SNAP
Statistics Area	0	CV_STAT_AREA
Statistics Area in Segment	1	CV_STAT_SEGMENT_AREA
Statistics Buffer	0	CV_STAT_BUFFER
Statistics File	0	CV_STAT_FILE
Statistics File in Segment	1	CV_STAT_SEGMENT_FILE
Statistics Interval	0	CV_STAT_INTERVAL
Statistics Lock	0	CV_STAT_LOCK
Statistics Segment	0	CV_STAT_SEGMENT
Statistics System	0	CV_STAT_SYSTEM
Storage Pool	2	STORPOOL
Structure Shared By	3	RECORDSHARE
SubTasks	0	CV_SUBTASK
Subarea	3	SYMBOL_SUBAREA
Subschema	2	SUBSCHEMA
Subschema Mapping	2	SSCMAP
Subschema Mapping in DBName	3	DBNAME_SSCMAP
Subschema registered	2	RP_DU_SUBSCHEMA
Subschema responsibility	2	SP_DU_SUBSCHEMA
System	1	SYSTEM
System Id	1	SYSTEMID
System Id granted	2	GP_GR_SYSTEMID
System Id granted	2	GP_US_SYSTEMID
System Profile	1	SYSPROFILE
System Profile granted	2	GP_GR_SYSPROFILE

Table 2: Objectnames in Alphabetical Order

Objectname	Level	Objectid
System Profile granted	2	GP_US_SYSPROFILE
Table	1	ALL_TABLE
Table Procedure	1	ALL_TABLEPROC
Table Procedure accessed	3	AM_TABLEPROC
Table Procedure granted	2	GP_GR_TABLEPROC
Table Procedure granted	2	GP_US_TABLEPROC
Table Procedure in Schema	2	TABLEPROC
Table accessed	3	AM_TABLE
Table granted	2	GP_GR_TABLE
Table granted	2	GP_US_TABLE
Table in Schema	2	TABLE
Table stored in Area	3	AREA_TABLE
Tape Journal	2	JOURNAL_TAPE
Task	2	TASK
Task in Category	2	CATEGORY_TASK
Time	0	CV_TIME
Time initiated Task	1	CV_TIME_TASK
Transaction	0	CV_TRANSACTION
UCF Terminal	0	CV_UCF
User Profile	1	PROFILE
User Profile granted	2	GP_GR_PROFILE
User Profile granted	2	GP_US_PROFILE
User granted	2	GP_GR_USER
User granted	2	GP_US_USER
User in Destination	3	DESTINATION_DICTUSER
User in Group	2	GROUP_USER
User signed on	0	CV_USER
View	1	ALL_VIEW
View Component	3	VIEW_COMPONENT
View Definition	3	VIEWDEFINITION

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	ObjectId
View accessed	3	AM_VIEW
View granted	2	GP_GR_VIEW
View granted	2	GP_US_VIEW
View in Schema	2	VIEW
View on Table	3	TABLE_VIEW
XA Storage Pool	2	XASTORPOOL

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
ADSO	0	SYSCA.VDBADCT2
Access Module	2	SYSTEM.AM
Access Module for Procedure	3	SYSVSYST2.AM_AMDEP
Access Module for Table	3	SYSVSYST2.AM_AMDEP
Access Module for Table Procedure	3	SYSVSYST2.AM_AMDEP
Access Module for View	3	SYSVSYST2.AM_AMDEP
Access Module granted	2	SYSTEM.RESOURCEAUTH
Access Module granted	2	SYSTEM.RESOURCEAUTH
Access Module in Category	2	IDMSSECS2.RESOURCE
Access Module using non SQL Table	3	SYSVSYST2.AM_AMDEP
Active Program	0	SYSCA.VDBADCT2
Active Program Details	1	SYSCA.VDBADCT2
Active Storage	0	SYSCA.VDBADCT2
Active Storage Details	1	SYSCA.VDBADCT2
Active Task	0	SYSCA.VDBADCT2
Active Task Details	1	SYSCA.VDBADCT2
Activity	1	IDMSSECS2.RESOURCE
Activity granted	2	IDMSSECS2.RESGROUPAUTH

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
Activity granted	2	IDMSSECS2.RESGROUPTH
Application	3	SYSVNTWK2.APPLICATION
Archive Journal	2	SYSTEM.JOURNAL
Area	2	SYSTEM.AREA
Area Procedure	3	SYSVNTWK2.AREAPROC
Area Statistics for Program	5	SYSVNTWK2.SUBPROGAREA
Area granted	2	IDMSSECS2.RESOURCEAUTH
Area granted	2	IDMSSECS2.RESOURCEAUTH
Area in DMCL	2	SYSVSYST2.DMCL_AREA
Area in File	3	SYSTEM.FILEMAP
Area in Non SQL Schema	2	SYSVNTWK2.NSQLAREA
Area in Subschema	3	SYSVNTWK2.SUBSCHEMA_AREA
Area used in Program	4	SYSVNTWK2.SUBPROGAREA
Area with Ready Mode in AM	3	SYSVNTWK2.AM_AREA_OPT
Attribute	2	IDMSSECS2.ATTRIBUTE
Attribute	2	IDMSSECU2.ATTRIBUTE
Attribute	2	SYSVNTWK2.ATTRIBUTE
Autotask	2	SYSVNTWK2.AUTOTASK
CV Area	0	SYSKA.VDBADCT2
CV Area in Segment	1	SYSKA.VDBADCT2
CV Backend in DBGroup	1	SYSKA.VDBADCT2
CV DBGroup	0	SYSKA.VDBADCT2
CV DBName	0	SYSKA.VDBADCT2
CV DBName including Segment	1	SYSKA.VDBADCT2
CV Database Buffer	0	SYSKA.VDBADCT2
CV Destination	0	SYSKA.VDBADCT2
CV File	0	SYSKA.VDBADCT2
CV File in Segment	1	SYSKA.VDBADCT2
CV Journal	0	SYSKA.VDBADCT2
CV Journal Buffer	0	SYSKA.VDBADCT2

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
CV LTerm in Line	1	SYSCA.VDBADCT2
CV Line	0	SYSCA.VDBADCT2
CV Line new Sysgen	0	SYSCA.VDBADCT2
CV Load List	0	SYSCA.VDBADCT2
CV Lterm	0	SYSCA.VDBADCT2
CV Node	0	SYSCA.VDBADCT2
CV PTerm in Line	1	SYSCA.VDBADCT2
CV Printer	0	SYSCA.VDBADCT2
CV Program	0	SYSCA.VDBADCT2
CV Program Pool new Sysgen	0	SYSCA.VDBADCT2
CV Pterm	0	SYSCA.VDBADCT2
CV Queue	0	SYSCA.VDBADCT2
CV Reent Prog Pool new Sysgen	0	SYSCA.VDBADCT2
CV Resource Table	0	SYSCA.VDBADCT2
CV Rununit	0	SYSCA.VDBADCT2
CV Rununit Detail	1	SYSCA.VDBADCT2
CV Segment	0	SYSCA.VDBADCT2
CV Segment in DBName	1	SYSCA.VDBADCT2
CV Storage Pool	0	SYSCA.VDBADCT2
CV Subschema Mapping	0	SYSCA.VDBADCT2
CV Subschema Mapping in DBName	1	SYSCA.VDBADCT2
CV Task	0	SYSCA.VDBADCT2
CV XA Storage Pool	0	SYSCA.VDBADCT2
CV XA Storage Pool new Sysgen	0	SYSCA.VDBADCT2
Calc Key	3	SYSVSYST2.VINDEX
Category	1	IDMSSECS2.RESOURCEGROUP
Category granted	2	IDMSSECS2.RESGROUPAUTH
Category granted	2	IDMSSECS2.RESGROUPAUTH
Central User	1	IDMSSECU2.USER
Check Condition	3	SYSTEM.SYNTAX

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
Class / Attribute	2	SYSVNTWK2.NSQLATTRIB
Class / Attribute	3	SYSVNTWK2.SUBSCHEMAATTRIB
Column	3	SYSTEM.COLUMN
Column of View	3	SYSTEM.COLUMN
Column of non SQL Table	3	SYSKA.NONSQL_COLUMN_A
Comment	2	SYSVNTWK2.NSQLCOMMENT
Comment	3	SYSVNTWK2.SUBSCHEMACOMMENT
Component of Loadlist	3	SYSVNTWK2.LOADLIST_COMPONENT
Constraint	2	SYSVSYST2.CONSTRAINT
Constraint Table referenced	3	SYSVSYST2.CONSTRAINT
Constraint Table referencing	3	SYSVSYST2.CONSTRAINT
DBGroup in DBTable	2	SYSTEM.DBNAME
DBName	2	SYSTEM.DBNAME
DBName & Segment granted	2	IDMSSECS2.RESOURCEAUTH
DBName & Segment granted	2	IDMSSECS2.RESOURCEAUTH
DBName including Segment	2	SYSTEM.DBSEGMENT
DBTable	1	SYSTEM.DBTABLE
DBTable granted	2	IDMSSECS2.RESOURCEAUTH
DBTable granted	2	IDMSSECS2.RESOURCEAUTH
DBTable including DBName	3	SYSVSYST2.DBNAME_DBTABLE
DCADMIN granted	2	IDMSSECS2.RESOURCEAUTH
DCADMIN granted	2	IDMSSECS2.RESOURCEAUTH
DCMT Command Outstanding	0	SYSKA.VDBADCT2
DDS	0	SYSKA.VDBADCT2
DMCL	1	SYSVSYST2.VDMCL
DMCL granted	2	IDMSSECS2.RESOURCEAUTH
DMCL granted	2	IDMSSECS2.RESOURCEAUTH
DMCL including Segment	2	SYSVSYST2.VDMCLSEGMENT
DMCL using Area	3	SYSVSYST2.DMCL_AREA
DMCL using DBTable	2	SYSVSYST2.VDMCL

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
DMCL using File	3	SYSVSYST2.DMCL_FILE
Data Compression Table	3	SYSVNTWK2.RECORDPROC
Data Compression Table	4	SYSVNTWK2.RECORDPROC,SYSTSCHM2.SOR-046
Data Sharing Group Member	0	SYSCA.VDBADCT2
Data Sharing LIST Structure	0	SYSCA.VDBADCT2
Data Sharing LOCK Structure	0	SYSCA.VDBADCT2
Data Sharing Summary	0	SYSCA.VDBADCT2
Database Buffer	2	SYSTEM.BUFFER
Deadlock	0	SYSCA.VDBADCT2
Destination	2	SYSVNTWK2.DESTINATION
Destination Node in Res Table	2	SYSVNTWK2.RESTABLE
Dictionary	0	SYSTEM.COLUMN
Dictionary User	1	SYSVNTWK2.DICTUSER
Dictionary via Node in Res Table	2	SYSVNTWK2.RESTABLE
Disk Journal	2	SYSTEM.JOURNAL
Displacement	3	SYSTEM.SYMBOL
Element Description	4	SYSVNTWK2.ELEMSDES
Element Indexed By	4	SYSVNTWK2.NAMEDES
Element Synonym	4	SYSVNTWK2.ELEMSYN
Element Value	4	SYSVNTWK2.ELEMVALUE
Element of Subschema Record	4	SYSVNTWK2.SUBRECORDELEM
Element of Subschema Record	5	SYSVNTWK2.SUBPROGRECORD_ELEM
Entity Type Authority	2	SYSVNTWK2.DICTUSER
Entry of Map Table	3	SYSVNTWK2.MAPTYPE
External Rununit	0	SYSCA.VDBADCT2
File	2	SYSTEM.FILE
File in Area	3	SYSTEM.FILEMAP
File in DMCL	2	SYSVSYST2.DMCL_FILE
File in Shared Cache	1	SYSCA.VDBADCT2
Grantee for DBName	3	IDMSSECS2.RESOURCEAUTH

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
Grantee for DBTable	2	IDMSSECS2.RESOURCEAUTH
Grantee for DCADMIN	1	IDMSSECS2.RESOURCEAUTH
Grantee for DMCL	2	IDMSSECS2.RESOURCEAUTH
Grantee for Execute	2	IDMSSECS2.RESGROUPAUTH
Grantee for Execute	2	IDMSSECS2.RESGROUPAUTH
Grantee for SYSADMIN	1	IDMSSECU2.RESOURCEAUTH
Grantee for Schema	2	SYSTEM.RESOURCEAUTH
Grantee for Segment	2	IDMSSECS2.RESOURCEAUTH
Grantee for System Id	2	IDMSSECS2.RESOURCEAUTH
Grantee for Use	2	IDMSSECS2.RESOURCEAUTH
Grantee on Access Module	3	SYSTEM.RESOURCEAUTH
Grantee on Area	3	IDMSSECS2.RESOURCEAUTH
Grantee on Group	2	IDMSSECU2.RESOURCEAUTH
Grantee on Procedure	3	SYSTEM.RESOURCEAUTH
Grantee on System Profile	2	IDMSSECS2.RESOURCEAUTH
Grantee on Table	3	SYSTEM.RESOURCEAUTH
Grantee on Table Procedure	3	SYSTEM.RESOURCEAUTH
Grantee on User	2	IDMSSECU2.RESOURCEAUTH
Grantee on User Profile	2	IDMSSECU2.RESOURCEAUTH
Grantee on View	3	SYSTEM.RESOURCEAUTH
Grantee on non SQL Table	3	SYSTEM.RESOURCEAUTH
Group	1	IDMSSECU2.USER
Group User belongs	2	IDMSSECU2.USERGROUP
Group granted	2	IDMSSECU2.RESOURCEAUTH
Group granted	2	IDMSSECU2.RESOURCEAUTH
IDD Class	1	SYSVNTWK2.CLASS
IDD Element Description	3	SYSVNTWK2.ELEMSDES
IDD Element Indexed By	3	SYSVNTWK2.NAMEDES
IDD Element Synonym	3	SYSVNTWK2.IDDELEMSYN
IDD Element Value	3	SYSVNTWK2.ELEMVALUE

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
IDD Record	1	SYSVNTWK2.IDDRECORD
IDD Record Element	2	SYSVNTWK2.RECORDELEM
IDD Record Synonym	2	SYSVNTWK2.IDDRECORDSYN
Index	3	SYSTEM.SYMBOL
Index	3	SYSVSYST2.VINDEX
Index stored in Area	3	SYSVSYST2.VINDEX
Journal Buffer	2	SYSTEM.BUFFER
Journal Status	0	SYSVA.VDBADCT2
Key Table	2	SYSVNTWK2.KEYTABLE
Key in Procedure	3	SYSVSYST2.VINDEX
Key in Table Procedure	3	SYSVSYST2.VINDEX
LU	0	SYSVA.VDBADCT2
Limit	0	SYSVA.VDBADCT2
Line	2	SYSVNTWK2.LINE
Load List	2	SYSVNTWK2.LOADLIST
Load Module	3	SYSVNTWK2.LOADMODULE
Load Module in Category	2	IDMSSECS2.RESOURCE
Loadlib	0	SYSVA.VDBADCT2
Local Dictionary in Res Table	2	SYSVNTWK2.RESTABLE
Lock on Area	0	SYSVA.VDBADCT2
Lock on Lterm	0	SYSVA.VDBADCT2
Log	0	SYSVA.VDBADCT2
Log Driver	0	SYSVA.VDBADCT2
Lterm	2	SYSVNTWK2.SYSTEM_LTERM
Lterm in Destination	3	SYSVNTWK2.DESTLTERM
Lterm in Line	3	SYSVNTWK2.PTERM_LTERM
Lterm in Pterm	4	SYSVNTWK2.PTERM_LTERM
MPMode	0	SYSVA.VDBADCT2
MT Queue Depth	0	SYSVA.VDBADCT2
Map Table	2	SYSVNTWK2.MAPTYPE

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
Member Record of Set	3	SYSVNTWK2.SET_RECORD_MEMBER
Member Record of Set	5	SYSVNTWK2.SUBPROGSET_MEMBER
Modeent of LU	1	SYSVA.VDBADCT2
Node	2	SYSVNTWK2.NODE
Non SQL Schema	1	SYSVNTWK2.NSQLSCHEMA
Non SQL Schema granted	2	IDMSSECS2.RESOURCEAUTH
Non SQL Schema granted	2	IDMSSECS2.RESOURCEAUTH
Non SQL Schema registered	2	SYSVNTWK2.U_NSQLSCHEMA
Non SQL Schema responsibility	2	SYSVNTWK2.U_NSQLSCHEMA
Non SQL Table	1	SYSVSYST2.VTABLENSQL
Non SQL Table accessed	3	SYSVSYST2.AM_AMDEP
Non SQL Table granted	2	SYSTEM.RESOURCEAUTH,SYSVSYST2.VTABLENSQL
Non SQL Table granted	2	SYSTEM.RESOURCEAUTH,SYSVSYST2.VTABLENSQL
Non SQL Table in Schema	2	SYSVSYST2.VTABLENSQL
Nucleus Reload	0	SYSVA.VDBADCT2
Owner Record of Set	3	SYSVNTWK2.SET_RECORD_OWNER
Owner Record of Set	5	SYSVNTWK2.SUBPROGSET_OWNER
Parameter of Procedure	3	SYSTEM.COLUMN
Parameter of Table Procedure	3	SYSTEM.COLUMN
Printer	2	SYSVNTWK2.SYSTEM_PRINTER
Printer in Destination	3	SYSVNTWK2.PRINTER
Procedure	1	SYSVSYST2.VTABLE
Procedure accessed	3	SYSVSYST2.AM_AMDEP
Procedure granted	2	SYSTEM.RESOURCEAUTH,SYSTEM.TABLE
Procedure granted	2	SYSTEM.RESOURCEAUTH,SYSTEM.TABLE
Procedure in Schema	2	SYSVSYST2.VTABLE
Program	2	SYSVNTWK2.PROGRAM
Program in Category	2	IDMSSECS2.RESOURCE
Program using Subschema	3	SYSVNTWK2.SUBSCHEMA_PROG
Pterm	3	SYSVNTWK2.PTERM

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
Queue	2	SYSVNTWK2.QUEUE
Queue in Category	2	IDMSSECS2.RESOURCE
Record	2	SYSVNTWK2.RECORDTYP
Record Control Key	4	SYSVNTWK2.SET_REC_CONTROLKEY
Record Element	3	SYSVNTWK2.RECORDELEM
Record Element	4	SYSVNTWK2.RECORDELEM,SYSTSCHM2.SOR-046
Record Foreign Key	4	SYSVNTWK2.SET_REC_FOREIGNKEY
Record Procedure	3	SYSVNTWK2.RECORDPROC
Record Procedure	4	SYSVNTWK2.RECORDPROC,SYSTSCHM2.SOR-046
Record Statistics for Program	5	SYSVNTWK2.SUBPROGRECORD
Record Synonym	3	SYSVNTWK2.RECORDSYN
Record Synonym	4	SYSVNTWK2.RECORDSYN,SYSTSCHM2.SOR-046
Record in Area	3	SYSVNTWK2.RECORDTYP
Record in Area	5	SYSVNTWK2.SUBPROGAREA_RECORD
Record in Subschema	3	SYSVNTWK2.SUBRECORD
Record used in Program	4	SYSVNTWK2.SUBPROGRECORD
Referenced Table in Constraint	3	SYSVSYST2.CONSTRAINT
Referencing Table in Constraint	3	SYSVSYST2.CONSTRAINT
Registree for Schema	2	SYSVNTWK2.U_NSQLSHEMA
Registree for Subschema	3	SYSVNTWK2.U_SUBSCHEMA
Relational Command Module	1	SYSVNTWK2.RCM
Relational Command Module in AM	3	SYSVNTWK2.AM_RCM
Reply pending	0	SYSKA.VDBADCT2
Report	1	SYSKA.VDBADCT2
Report Class / Destination	0	SYSKA.VDBADCT2
Responsible for Schema	2	SYSVNTWK2.U_NSQLSHEMA
Responsible for Subschema	3	SYSVNTWK2.U_SUBSCHEMA
Rununit	2	SYSVNTWK2.RUNUNIT
Rununit in Category	2	IDMSSECS2.RESOURCE
SNA Pterm	0	SYSKA.VDBADCT2

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
SQL Schema	1	SYSTEM.SCHEMA
SYSADMIN granted ?	2	IDMSSECU2.RESOURCEAUTH
SYSADMIN granted ?	2	IDMSSECU2.RESOURCEAUTH
Schema Mapping of AM	3	SYSVNTWK2.AM_SCHEMA_MAP
Schema granted	2	SYSTEM.RESOURCEAUTH
Schema granted	2	SYSTEM.RESOURCEAUTH
Schema referencing DBName	3	SYSVSYST2.DBNAME_SCHEMA
Schema referencing Segment	2	SYSTEM.SCHEMA
Schema using Area as default	3	SYSTEM.SCHEMA
Segment	1	SYSTEM.SEGMENT
Segment in DBName	3	SYSTEM.DBSEGMENT
Segment in DMCL	2	SYSVSYST2.VDMCLSEGMENT
Segment.Area Override	2	SYSVSYST2.VDMCLAREA
Segment.File Override	2	SYSVSYST2.VDMCLFILE
Set	2	SYSVNTWK2.NSQLSET
Set Record is member	3	SYSVNTWK2.RECORDTYP_SET_ME
Set Record is member	4	SYSVNTWK2.MBR_REC_SET_ME
Set Statistics for Program	5	SYSVNTWK2.SUBPROGSET
Set in Subschema	3	SYSVNTWK2.SUBSCHEMA_SET
Set owned by Record	3	SYSVNTWK2.RECORDTYP_SET_OW
Set owned by Record	4	SYSVNTWK2.MBR_REC_SET_OW
Set used in Program	4	SYSVNTWK2.SUBPROGSET
Shared Area in DMCL	2	SYSVSYST2.VDMCL_DSH_AREA
Shared Cache	0	SYSCA.VDBADCT2
Shared Cache in DMCL	2	SYSVSYST2.VDMCL_DSH_CACHE
Shared File in DMCL	2	SYSVSYST2.VDMCL_DSH_FILE
Snap	0	SYSCA.VDBADCT2
Statistics Area	0	SYSCA.VDBADCT2
Statistics Area in Segment	1	SYSCA.VDBADCT2
Statistics Buffer	0	SYSCA.VDBADCT2

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
Statistics File	0	SYSCA.VDBADCT2
Statistics File in Segment	1	SYSCA.VDBADCT2
Statistics Interval	0	SYSCA.VDBADCT2
Statistics Lock	0	SYSCA.VDBADCT2
Statistics Segment	0	SYSCA.VDBADCT2
Statistics System	0	SYSCA.VDBADCT2
Storage Pool	2	SYSVNTWK2.STORPOOL
Structure Shared By	3	SYSVNTWK2.RECORDSHARE
SubTasks	0	SYSCA.VDBADCT2
Subarea	3	SYSTEM.SYMBOL
Subschema	2	SYSVNTWK2.SUBSCHEMA
Subschema Mapping	2	SYSTEM.DBSSC
Subschema Mapping in DBName	3	SYSTEM.DBSSC
Subschema registered	2	SYSVNTWK2.U_SUBSCHEMA
Subschema responsibility	2	SYSVNTWK2.U_SUBSCHEMA
System	1	SYSVNTWK2.SYSTEM
System Id	1	IDMSSECS2.RESOURCE
System Id granted	2	IDMSSECS2.RESOURCEAUTH
System Id granted	2	IDMSSECS2.RESOURCEAUTH
System Profile	1	IDMSSECS2.PROFILE
System Profile granted	2	IDMSSECS2.RESOURCEAUTH
System Profile granted	2	IDMSSECS2.RESOURCEAUTH
Table	1	SYSVSYST2.VTABLE
Table Procedure	1	SYSVSYST2.VTABLE
Table Procedure accessed	3	SYSVSYST2.AM_AMDEP
Table Procedure granted	2	SYSTEM.RESOURCEAUTH,SYSTEM.TABLE
Table Procedure granted	2	SYSTEM.RESOURCEAUTH,SYSTEM.TABLE
Table Procedure in Schema	2	SYSVSYST2.VTABLE
Table accessed	3	SYSVSYST2.AM_AMDEP
Table granted	2	SYSTEM.RESOURCEAUTH,SYSTEM.TABLE

Table 3: Objectnames and Schema Table in Alphabetical Order

Objectname	Level	Schema Table Used by Object
Table granted	2	SYSTEM.RESOURCEAUTH,SYSTEM.TABLE
Table in Schema	2	SYSVSYST2.VTABLE
Table stored in Area	3	SYSVSYST2.VTABLE
Tape Journal	2	SYSTEM.JOURNAL
Task	2	SYSVNTWK2.TASK
Task in Category	2	IDMSSECS2.RESOURCE
Time	0	SYSCA.VDBADCT2
Time initiated Task	1	SYSCA.VDBADCT2
Transaction	0	SYSCA.VDBADCT2
UCF Terminal	0	SYSCA.VDBADCT2
User Profile	1	IDMSSECU2.PROFILE
User Profile granted	2	IDMSSECU2.RESOURCEAUTH
User Profile granted	2	IDMSSECU2.RESOURCEAUTH
User granted	2	IDMSSECU2.RESOURCEAUTH
User granted	2	IDMSSECU2.RESOURCEAUTH
User in Destination	3	SYSVNTWK2.DESTINATION_DICTUS
User in Group	2	IDMSSECU2.USERGROUP
User signed on	0	SYSCA.VDBADCU2
View	1	SYSVSYST2.VTABLE
View Component	3	SYSTEM.VIEWDEP
View Definition	3	SYSTEM.SYNTAX
View accessed	3	SYSVSYST2.AM_AMDEP
View granted	2	SYSTEM.RESOURCEAUTH,SYSTEM.TABLE
View granted	2	SYSTEM.RESOURCEAUTH,SYSTEM.TABLE
View in Schema	2	SYSVSYST2.VTABLE
View on Table	3	SYSTEM.VIEWDEP
XA Storage Pool	2	SYSVNTWK2.STORPOOL

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objectname
IDMSSECS2.ATTRIBUTE	2	Attribute
IDMSSECS2.PROFILE	1	System Profile
IDMSSECS2.RESGROUPAUTH	2	Activity granted
IDMSSECS2.RESGROUPAUTH	2	Activity granted
IDMSSECS2.RESGROUPAUTH	2	Category granted
IDMSSECS2.RESGROUPAUTH	2	Category granted
IDMSSECS2.RESGROUPAUTH	2	Grantee for Execute
IDMSSECS2.RESGROUPAUTH	2	Grantee for Execute
IDMSSECS2.RESOURCE	1	Activity
IDMSSECS2.RESOURCE	1	System Id
IDMSSECS2.RESOURCE	2	Access Module in Category
IDMSSECS2.RESOURCE	2	Load Module in Category
IDMSSECS2.RESOURCE	2	Program in Category
IDMSSECS2.RESOURCE	2	Queue in Category
IDMSSECS2.RESOURCE	2	Rununit in Category
IDMSSECS2.RESOURCE	2	Task in Category
IDMSSECS2.RESOURCEAUTH	1	Grantee for DCADMIN
IDMSSECS2.RESOURCEAUTH	2	Area granted
IDMSSECS2.RESOURCEAUTH	2	Area granted
IDMSSECS2.RESOURCEAUTH	2	DBName & Segment granted
IDMSSECS2.RESOURCEAUTH	2	DBName & Segment granted
IDMSSECS2.RESOURCEAUTH	2	DBTable granted
IDMSSECS2.RESOURCEAUTH	2	DBTable granted
IDMSSECS2.RESOURCEAUTH	2	DCADMIN granted ?
IDMSSECS2.RESOURCEAUTH	2	DCADMIN granted ?
IDMSSECS2.RESOURCEAUTH	2	DMCL granted
IDMSSECS2.RESOURCEAUTH	2	DMCL granted
IDMSSECS2.RESOURCEAUTH	2	Grantee for DBTable
IDMSSECS2.RESOURCEAUTH	2	Grantee for DMCL

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objecname
IDMSSECS2.RESOURCEAUTH	2	Grantee for Segment
IDMSSECS2.RESOURCEAUTH	2	Grantee for System Id
IDMSSECS2.RESOURCEAUTH	2	Grantee for Use
IDMSSECS2.RESOURCEAUTH	2	Grantee on System Profile
IDMSSECS2.RESOURCEAUTH	2	Non SQL Schema granted
IDMSSECS2.RESOURCEAUTH	2	Non SQL Schema granted
IDMSSECS2.RESOURCEAUTH	2	System Id granted
IDMSSECS2.RESOURCEAUTH	2	System Id granted
IDMSSECS2.RESOURCEAUTH	2	System Profile granted
IDMSSECS2.RESOURCEAUTH	2	System Profile granted
IDMSSECS2.RESOURCEAUTH	3	Grantee for DBName
IDMSSECS2.RESOURCEAUTH	3	Grantee on Area
IDMSSECS2.RESOURCEGROUP	1	Category
IDMSSECU2.ATTRIBUTE	2	Attribute
IDMSSECU2.PROFILE	1	User Profile
IDMSSECU2.RESOURCEAUTH	1	Grantee for SYSADMIN
IDMSSECU2.RESOURCEAUTH	2	Grantee on Group
IDMSSECU2.RESOURCEAUTH	2	Grantee on User
IDMSSECU2.RESOURCEAUTH	2	Grantee on User Profile
IDMSSECU2.RESOURCEAUTH	2	Group granted
IDMSSECU2.RESOURCEAUTH	2	Group granted
IDMSSECU2.RESOURCEAUTH	2	SYSADMIN granted ?
IDMSSECU2.RESOURCEAUTH	2	SYSADMIN granted ?
IDMSSECU2.RESOURCEAUTH	2	User Profile granted
IDMSSECU2.RESOURCEAUTH	2	User Profile granted
IDMSSECU2.RESOURCEAUTH	2	User granted
IDMSSECU2.RESOURCEAUTH	2	User granted
IDMSSECU2.USER	1	Central User
IDMSSECU2.USER	1	Group
IDMSSECU2.USERGROUP	2	Group User belongs

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objectname
IDMSSECU2.USERGROUP	2	User in Group
SYSCA.NONSQL_COLUMN_A	3	Column of non SQL Table
SYSCA.VDBADCT2	0	ADSO
SYSCA.VDBADCT2	0	Active Program
SYSCA.VDBADCT2	0	Active Storage
SYSCA.VDBADCT2	0	Active Task
SYSCA.VDBADCT2	0	CV Area
SYSCA.VDBADCT2	0	CV DBGroup
SYSCA.VDBADCT2	0	CV DBName
SYSCA.VDBADCT2	0	CV Database Buffer
SYSCA.VDBADCT2	0	CV Destination
SYSCA.VDBADCT2	0	CV File
SYSCA.VDBADCT2	0	CV Journal
SYSCA.VDBADCT2	0	CV Journal Buffer
SYSCA.VDBADCT2	0	CV Line
SYSCA.VDBADCT2	0	CV Line new Sysgen
SYSCA.VDBADCT2	0	CV Load List
SYSCA.VDBADCT2	0	CV Lterm
SYSCA.VDBADCT2	0	CV Node
SYSCA.VDBADCT2	0	CV Printer
SYSCA.VDBADCT2	0	CV Program
SYSCA.VDBADCT2	0	CV Program Pool new Sysgen
SYSCA.VDBADCT2	0	CV Pterm
SYSCA.VDBADCT2	0	CV Queue
SYSCA.VDBADCT2	0	CV Reent Prog Pool new Sysgen
SYSCA.VDBADCT2	0	CV Resource Table
SYSCA.VDBADCT2	0	CV Rununit
SYSCA.VDBADCT2	0	CV Segment
SYSCA.VDBADCT2	0	CV Storage Pool
SYSCA.VDBADCT2	0	CV Subschema Mapping

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objectname
SYSCA.VDBADCT2	0	CV Task
SYSCA.VDBADCT2	0	CV XA Storage Pool
SYSCA.VDBADCT2	0	CV XA Storage Pool new Sysgen
SYSCA.VDBADCT2	0	DCMT Command Outstanding
SYSCA.VDBADCT2	0	DDS
SYSCA.VDBADCT2	0	Data Sharing Group Member
SYSCA.VDBADCT2	0	Data Sharing LIST Structure
SYSCA.VDBADCT2	0	Data Sharing LOCK Structure
SYSCA.VDBADCT2	0	Data Sharing Summary
SYSCA.VDBADCT2	0	Deadlock
SYSCA.VDBADCT2	0	External Rununit
SYSCA.VDBADCT2	0	Journal Status
SYSCA.VDBADCT2	0	LU
SYSCA.VDBADCT2	0	Limit
SYSCA.VDBADCT2	0	Loadlib
SYSCA.VDBADCT2	0	Lock on Area
SYSCA.VDBADCT2	0	Lock on Lterm
SYSCA.VDBADCT2	0	Log
SYSCA.VDBADCT2	0	Log Driver
SYSCA.VDBADCT2	0	MPMode
SYSCA.VDBADCT2	0	MT Queue Depth
SYSCA.VDBADCT2	0	Nucleus Reload
SYSCA.VDBADCT2	0	Reply pending
SYSCA.VDBADCT2	0	Report Class / Destination
SYSCA.VDBADCT2	0	SNA Pterm
SYSCA.VDBADCT2	0	Shared Cache
SYSCA.VDBADCT2	0	Snap
SYSCA.VDBADCT2	0	Statistics Area
SYSCA.VDBADCT2	0	Statistics Buffer
SYSCA.VDBADCT2	0	Statistics File

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objecname
SYSCA.VDBADCT2	0	Statistics Interval
SYSCA.VDBADCT2	0	Statistics Lock
SYSCA.VDBADCT2	0	Statistics Segment
SYSCA.VDBADCT2	0	Statistics System
SYSCA.VDBADCT2	0	SubTasks
SYSCA.VDBADCT2	0	Time
SYSCA.VDBADCT2	0	Transaction
SYSCA.VDBADCT2	0	UCF Terminal
SYSCA.VDBADCT2	1	Active Program Details
SYSCA.VDBADCT2	1	Active Storage Details
SYSCA.VDBADCT2	1	Active Task Details
SYSCA.VDBADCT2	1	CV Area in Segment
SYSCA.VDBADCT2	1	CV Backend in DBGroup
SYSCA.VDBADCT2	1	CV DBName including Segment
SYSCA.VDBADCT2	1	CV File in Segment
SYSCA.VDBADCT2	1	CV LTerm in Line
SYSCA.VDBADCT2	1	CV PTerm in Line
SYSCA.VDBADCT2	1	CV Rununit Detail
SYSCA.VDBADCT2	1	CV Segment in DBName
SYSCA.VDBADCT2	1	CV Subschema Mapping in DBName
SYSCA.VDBADCT2	1	File in Shared Cache
SYSCA.VDBADCT2	1	Modeent of LU
SYSCA.VDBADCT2	1	Report
SYSCA.VDBADCT2	1	Statistics Area in Segment
SYSCA.VDBADCT2	1	Statistics File in Segment
SYSCA.VDBADCT2	1	Time initiated Task
SYSCA.VDBADCU2	0	User signed on
SYSTEM.AM	2	Access Module
SYSTEM.AREA	2	Area
SYSTEM.BUFFER	2	Database Buffer

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objectname
SYSTEM.BUFFER	2	Journal Buffer
SYSTEM.COLUMN	0	Dictionary
SYSTEM.COLUMN	3	Column
SYSTEM.COLUMN	3	Column of View
SYSTEM.COLUMN	3	Parameter of Procedure
SYSTEM.COLUMN	3	Parameter of Table Procedure
SYSTEM.DBNAME	2	DBGroup in DBTable
SYSTEM.DBNAME	2	DBName
SYSTEM.DBSEGMENT	2	DBName including Segment
SYSTEM.DBSEGMENT	3	Segment in DBName
SYSTEM.DBSSC	2	Subschema Mapping
SYSTEM.DBSSC	3	Subschema Mapping in DBName
SYSTEM.DBTABLE	1	DBTable
SYSTEM.FILE	2	File
SYSTEM.FILEMAP	3	Area in File
SYSTEM.FILEMAP	3	File in Area
SYSTEM.JOURNAL	2	Archive Journal
SYSTEM.JOURNAL	2	Disk Journal
SYSTEM.JOURNAL	2	Tape Journal
SYSTEM.RESOURCEAUTH	2	Access Module granted
SYSTEM.RESOURCEAUTH	2	Access Module granted
SYSTEM.RESOURCEAUTH	2	Grantee for Schema
SYSTEM.RESOURCEAUTH	2	Schema granted
SYSTEM.RESOURCEAUTH	2	Schema granted
SYSTEM.RESOURCEAUTH	3	Grantee on Access Module
SYSTEM.RESOURCEAUTH	3	Grantee on Procedure
SYSTEM.RESOURCEAUTH	3	Grantee on Table
SYSTEM.RESOURCEAUTH	3	Grantee on Table Procedure
SYSTEM.RESOURCEAUTH	3	Grantee on View
SYSTEM.RESOURCEAUTH	3	Grantee on non SQL Table

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objecname
SYSTEM.RESOURCEAUTH,	2	Non SQL Table granted
SYSTEM.RESOURCEAUTH,	2	Non SQL Table granted
SYSTEM.RESOURCEAUTH,	2	Procedure granted
SYSTEM.RESOURCEAUTH,	2	Procedure granted
SYSTEM.RESOURCEAUTH,	2	Table Procedure granted
SYSTEM.RESOURCEAUTH,	2	Table Procedure granted
SYSTEM.RESOURCEAUTH,	2	Table granted
SYSTEM.RESOURCEAUTH,	2	Table granted
SYSTEM.RESOURCEAUTH,	2	View granted
SYSTEM.RESOURCEAUTH,	2	View granted
SYSTEM.SCHEMA	1	SQL Schema
SYSTEM.SCHEMA	2	Schema referencing Segment
SYSTEM.SCHEMA	3	Schema using Area as default
SYSTEM.SEGMENT	1	Segment
SYSTEM.SYMBOL	3	Displacement
SYSTEM.SYMBOL	3	Index
SYSTEM.SYMBOL	3	Subarea
SYSTEM.SYNTAX	3	Check Condition
SYSTEM.SYNTAX	3	View Definition
SYSTEM.TABLE	2	Procedure granted
SYSTEM.TABLE	2	Procedure granted
SYSTEM.TABLE	2	Table Procedure granted
SYSTEM.TABLE	2	Table Procedure granted
SYSTEM.TABLE	2	Table granted
SYSTEM.TABLE	2	Table granted
SYSTEM.TABLE	2	View granted
SYSTEM.TABLE	2	View granted
SYSTEM.VIEWDEP	3	View Component
SYSTEM.VIEWDEP	3	View on Table
SYSTSCHM2.SOR-046	4	Data Compression Table

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objectname
SYSTSCHM2.SOR-046	4	Record Element
SYSTSCHM2.SOR-046	4	Record Procedure
SYSTSCHM2.SOR-046	4	Record Synonym
SYSVNTWK2.AM_AREA_OPT	3	Area with Ready Mode in AM
SYSVNTWK2.AM_RCM	3	Relational Command Module in AM
SYSVNTWK2.AM_SCHEMA_MAP	3	Schema Mapping of AM
SYSVNTWK2.APPLICATION	3	Application
SYSVNTWK2.AREAPROC	3	Area Procedure
SYSVNTWK2.ATTRIBUTE	2	Attribute
SYSVNTWK2.AUTOTASK	2	Autotask
SYSVNTWK2.CLASS	1	IDD Class
SYSVNTWK2.DESTINATION	2	Destination
SYSVNTWK2.DESTINATION_DICTUS	3	User in Destination
SYSVNTWK2.DESTLTERM	3	Lterm in Destination
SYSVNTWK2.DICTUSER	1	Dictionary User
SYSVNTWK2.DICTUSER	2	Entity Type Authority
SYSVNTWK2.ELEMSDES	3	IDD Element Description
SYSVNTWK2.ELEMSDES	4	Element Description
SYSVNTWK2.ELEMSYN	4	Element Synonym
SYSVNTWK2.ELEMVALUE	3	IDD Element Value
SYSVNTWK2.ELEMVALUE	4	Element Value
SYSVNTWK2.IDDELEMSYN	3	IDD Element Synonym
SYSVNTWK2.IDDRECORD	1	IDD Record
SYSVNTWK2.IDDRECORDSYN	2	IDD Record Synonym
SYSVNTWK2.KEYTABLE	2	Key Table
SYSVNTWK2.LINE	2	Line
SYSVNTWK2.LOADLIST	2	Load List
SYSVNTWK2.LOADLIST_COMPONENT	3	Component of Loadlist
SYSVNTWK2.LOADMODULE	3	Load Module
SYSVNTWK2.MAPTYPE	2	Map Table

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objecname
SYSVNTWK2.MAPTYPE	3	Entry of Map Table
SYSVNTWK2.MBR_REC_SET_ME	4	Set Record is member
SYSVNTWK2.MBR_REC_SET_OW	4	Set owned by Record
SYSVNTWK2.NAMEDES	3	IDD Element Indexed By
SYSVNTWK2.NAMEDES	4	Element Indexed By
SYSVNTWK2.NODE	2	Node
SYSVNTWK2.NSQLAREA	2	Area in Non SQL Schema
SYSVNTWK2.NSQLATTRIB	2	Class / Attribute
SYSVNTWK2.NSQLCOMMENT	2	Comment
SYSVNTWK2.NSQLSCHEMA	1	Non SQL Schema
SYSVNTWK2.NSQLSET	2	Set
SYSVNTWK2.PRINTER	3	Printer in Destination
SYSVNTWK2.PROGRAM	2	Program
SYSVNTWK2.PTERM	3	Pterm
SYSVNTWK2.PTERM_LTERM	3	Lterm in Line
SYSVNTWK2.PTERM_LTERM	4	Lterm in Pterm
SYSVNTWK2.QUEUE	2	Queue
SYSVNTWK2.RCM	1	Relational Command Module
SYSVNTWK2.RECORDELEM	2	IDD Record Element
SYSVNTWK2.RECORDELEM	3	Record Element
SYSVNTWK2.RECORDELEM,	4	Record Element
SYSVNTWK2.RECORDPROC	3	Data Compression Table
SYSVNTWK2.RECORDPROC	3	Record Procedure
SYSVNTWK2.RECORDPROC,	4	Data Compression Table
SYSVNTWK2.RECORDPROC,	4	Record Procedure
SYSVNTWK2.RECORDSHARE	3	Structure Shared By
SYSVNTWK2.RECORDSYN	3	Record Synonym
SYSVNTWK2.RECORDSYN,	4	Record Synonym
SYSVNTWK2.RECORDTYP	2	Record
SYSVNTWK2.RECORDTYP	3	Record in Area

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objectname
SYSVNTWK2.RECORDTYP_SET_ME	3	Set Record is member
SYSVNTWK2.RECORDTYP_SET_OW	3	Set owned by Record
SYSVNTWK2.RESTABLE	2	Destination Node in Res Table
SYSVNTWK2.RESTABLE	2	Dictionary via Node in Res Table
SYSVNTWK2.RESTABLE	2	Local Dictionary in Res Table
SYSVNTWK2.RUNUNIT	2	Rununit
SYSVNTWK2.SET_RECORD_MEMBER	3	Member Record of Set
SYSVNTWK2.SET_RECORD_OWNER	3	Owner Record of Set
SYSVNTWK2.SET_REC_CONTROLKEY	4	Record Control Key
SYSVNTWK2.SET_REC_FOREIGNKEY	4	Record Foreign Key
SYSVNTWK2.STORPOOL	2	Storage Pool
SYSVNTWK2.STORPOOL	2	XA Storage Pool
SYSVNTWK2.SUBPROGAREA	4	Area used in Program
SYSVNTWK2.SUBPROGAREA	5	Area Statistics for Program
SYSVNTWK2.SUBPROGAREA_RECORD	5	Record in Area
SYSVNTWK2.SUBPROGRECORD	4	Record used in Program
SYSVNTWK2.SUBPROGRECORD	5	Record Statistics for Program
SYSVNTWK2.SUBPROGRECORD_ELEM	5	Element of Subschema Record
SYSVNTWK2.SUBPROGSET	4	Set used in Program
SYSVNTWK2.SUBPROGSET	5	Set Statistics for Program
SYSVNTWK2.SUBPROGSET_MEMBER	5	Member Record of Set
SYSVNTWK2.SUBPROGSET_OWNER	5	Owner Record of Set
SYSVNTWK2.SUBRECORD	3	Record in Subschema
SYSVNTWK2.SUBRECORDELEM	4	Element of Subschema Record
SYSVNTWK2.SUBSCHEMA	2	Subschema
SYSVNTWK2.SUBSCHEMAATTRIB	3	Class / Attribute
SYSVNTWK2.SUBSCHEMACOMMENT	3	Comment
SYSVNTWK2.SUBSCHEMA_AREA	3	Area in Subschema
SYSVNTWK2.SUBSCHEMA_PROG	3	Program using Subschema
SYSVNTWK2.SUBSCHEMA_SET	3	Set in Subschema

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objecname
SYSVNTWK2.SYSTEM	1	System
SYSVNTWK2.SYSTEM_LTERM	2	Lterm
SYSVNTWK2.SYSTEM_PRINTER	2	Printer
SYSVNTWK2.TASK	2	Task
SYSVNTWK2.U_NSQLSHEMA	2	Non SQL Schema registered
SYSVNTWK2.U_NSQLSHEMA	2	Non SQL Schema responsibility
SYSVNTWK2.U_NSQLSHEMA	2	Registree for Schema
SYSVNTWK2.U_NSQLSHEMA	2	Responsible for Schema
SYSVNTWK2.U_SUBSCHEMA	2	Subschema registered
SYSVNTWK2.U_SUBSCHEMA	2	Subschema responsibility
SYSVNTWK2.U_SUBSCHEMA	3	Registree for Subschema
SYSVNTWK2.U_SUBSCHEMA	3	Responsible for Subschema
SYSVSYST2.AM_AMDEP	3	Access Module for Procedure
SYSVSYST2.AM_AMDEP	3	Access Module for Table
SYSVSYST2.AM_AMDEP	3	Access Module for Table Procedure
SYSVSYST2.AM_AMDEP	3	Access Module for View
SYSVSYST2.AM_AMDEP	3	Access Module using non SQL Table
SYSVSYST2.AM_AMDEP	3	Non SQL Table accessed
SYSVSYST2.AM_AMDEP	3	Procedure accessed
SYSVSYST2.AM_AMDEP	3	Table Procedure accessed
SYSVSYST2.AM_AMDEP	3	Table accessed
SYSVSYST2.AM_AMDEP	3	View accessed
SYSVSYST2.CONSTRAINT	2	Constraint
SYSVSYST2.CONSTRAINT	3	Constraint Table referenced
SYSVSYST2.CONSTRAINT	3	Constraint Table referencing
SYSVSYST2.CONSTRAINT	3	Referenced Table in Constraint
SYSVSYST2.CONSTRAINT	3	Referencing Table in Constraint
SYSVSYST2.DBNAME_DBTABLE	3	DBTable including DBName
SYSVSYST2.DBNAME_SCHEMA	3	Schema referencing DBName
SYSVSYST2.DMCL_AREA	2	Area in DMCL

Table 4: Schema Table in Alphabetical Order

Schema.Table	Level	Objectname
SYSVSYST2.DMCL_AREA	3	DMCL using Area
SYSVSYST2.DMCL_FILE	2	File in DMCL
SYSVSYST2.DMCL_FILE	3	DMCL using File
SYSVSYST2.VDMCL	1	DMCL
SYSVSYST2.VDMCL	2	DMCL using DBTable
SYSVSYST2.VDMCLAREA	2	Segment.Area Override
SYSVSYST2.VDMCLFILE	2	Segment.File Override
SYSVSYST2.VDMCLSEGMENT	2	DMCL including Segment
SYSVSYST2.VDMCLSEGMENT	2	Segment in DMCL
SYSVSYST2.VDMCL_DSH_AREA	2	Shared Area in DMCL
SYSVSYST2.VDMCL_DSH_CACHE	2	Shared Cache in DMCL
SYSVSYST2.VDMCL_DSH_FILE	2	Shared File in DMCL
SYSVSYST2.VINDEX	3	Calc Key
SYSVSYST2.VINDEX	3	Index
SYSVSYST2.VINDEX	3	Index stored in Area
SYSVSYST2.VINDEX	3	Key in Procedure
SYSVSYST2.VINDEX	3	Key in Table Procedure
SYSVSYST2.VTABLE	1	Procedure
SYSVSYST2.VTABLE	1	Table
SYSVSYST2.VTABLE	1	Table Procedure
SYSVSYST2.VTABLE	1	View
SYSVSYST2.VTABLE	2	Procedure in Schema
SYSVSYST2.VTABLE	2	Table Procedure in Schema
SYSVSYST2.VTABLE	2	Table in Schema
SYSVSYST2.VTABLE	2	View in Schema
SYSVSYST2.VTABLE	3	Table stored in Area
SYSVSYST2.VTABLENSQL	1	Non SQL Table
SYSVSYST2.VTABLENSQL	2	Non SQL Table granted
SYSVSYST2.VTABLENSQL	2	Non SQL Table granted
SYSVSYST2.VTABLENSQL	2	Non SQL Table in Schema

Uninstalling and Reinstalling

Uninstalling the CA-IDMS Visual DBA PC Client component

Uninstalling CA-IDMS Visual DBA 1.0

CA-IDMS Visual DBA 2.0 coexists with 1.0, so there is no need to uninstall 1.0 prior to installing 2.0. However, if at any time you wish to uninstall version 1.0 on your PC, go to the Control Panel, select Add/Remove Programs, select CA-IDMS Visual DBA 1.0, and click Add/Remove.

Uninstalling CA-IDMS Visual DBA 2.0

To uninstall the PC Client component of Visual DBA 2.0, go to the Control Panel, select Add/Remove Programs, select CA-IDMS Visual DBA 2.0, and click Add/Remove.

Uninstalling the mainframe CA-IDMS Visual DBA SQL Definitions

Uninstalling CA-IDMS Visual DBA 1.0

You can use CA-IDMS Visual DBA 2.0 to uninstall the 1.0 mainframe SQL definitions as follows:

1. Open an SQL Command Console session of CA-IDMS Visual DBA 2.0 on the node (ODBC data source) that contains the dictionaries you want to uninstall.
2. Select the dictionary that you want to uninstall from the dictionary drop-down menu.
3. Open the VDB1UNIN.SQL script and execute it.

4. Repeat the previous two steps for each dictionary you want to uninstall.

Note: VDB1UNIN.SQL is found in the Mainfram folder of the CA-IDMS Visual DBA 2.0 product folder. This folder is created during the installation of the CA-IDMS Visual DBA 2.0 on the PC. See the **Installing and Starting CA-IDMS Visual DBA** chapter for more details.

Alternately, you can upload the VDB1UNIN.SQL file to the mainframe and run an IDMSBCF batch job to execute the script. The IDMSBCF batch job must be run against each dictionary for which CA-IDMS Visual DBA 1.0 SQL definitions are to be uninstalled.

Uninstalling CA-IDMS Visual DBA 2.0

You can use CA-IDMS Visual DBA 2.0 to uninstall the mainframe SQL definitions of an existing 2.0 version as follows:

1. Open an SQL Command Console session of CA-IDMS Visual DBA 2.0 on the node (ODBC data source) that contains the dictionaries you want to uninstall.
2. Select the dictionary that you want to uninstall from the dictionary drop-down menu.
3. Open the appropriate VDB2UNxx.SQL script and execute it:
 - VDB2UN12.SQL for CA-IDMS Rel 12.0
 - VDB2UN14.SQL for CA-IDMS Rel 14.x
 - VDB2UN15.SQL for CA-IDMS Rel 15.x
4. Repeat the previous two steps for each dictionary you want to uninstall.

Note: The VDB2UNxx.SQL scripts are found in the Mainfram subfolder of the CA-IDMS Visual DBA 2.0 product folder.

Alternately, you can upload the VDB2UNxx.SQL file to the mainframe and run an IDMSBCF batch job to execute the script. The IDMSBCF batch job must be run against each dictionary for which CA-IDMS Visual DBA 2.0 SQL definitions are to be uninstalled.

Reinstalling CA-IDMS Visual DBA 2.0

In order to reinstall CA-IDMS Visual DBA 2.0 on top of an existing 2.0 installation, you must first uninstall both the PC Client and the mainframe SQL definitions of the existing version.

Using the Demo20.cfg File

The CA-IDMS Visual DBA product folder contains an example saved configuration file named `demo20.cfg`. This file may be used to

- Verify the successful installation of CA-IDMS Visual DBA on the PC.
- Familiarize yourself with the CA-IDMS Visual DBA tree and its objects without having to establish a mainframe connection.

To use this saved configuration file, select Open from the File menu and select `demo20.cfg` from the product folder. To avoid possible SQL errors when using this saved configuration, you must establish a dummy connection to the ODBC data source "DummyConnect". This DummyConnect data source is predefined in the Node window of `demo20.cfg`.

The Dictionary object in the saved configuration has 3 instances: APPLDICT, SYSDICT and SYSTEM, but only the dictionary SYSDICT has cached objects (instances). The saved configuration contains cached instances for most of the objects in the CA-IDMS Visual DBA object tree.

You may view the Detail Information pane and the dialogs for Alter, Create, Drop, Grant, Revoke, Register and Responsibility. You may also invoke the online help for these dialogs.

If you attempt to display information that has not been cached in this configuration file, CA-IDMS Visual DBA will attempt to establish a connection to the data source that was used to create this saved configuration. If you have not established the "dummy connection", you will receive an SQL error. You may also receive an SQL error if you attempt to alter, create, drop, grant, or revoke an entity without having established the "dummy connection". If this happens, click OK to continue using `demo20.cfg`

To see the result of any Alter, Create, Drop, Grant, or Revoke dialog while using the dummy connection, select the View Syntax boxes in the preference settings for the OK Action preferences. Because there is no real connection, the syntax will not be executed.

Note: Once you have made a (dummy) connection to “DummyConnect”, all database requests from CA-IDMS Visual DBA will return no instances or will return null attributes for all nodes. You must exit and restart CA-IDMS Visual DBA before any real connections can be made.

Index

A

Application window, 2-6

Assigning privileges

- create, alter and drop object instances, 2-15
- granting and revoking, 2-15
- to objects, 2-14
- to update object instances, 2-14

B

Bar tab, 3-27

C

CA-IDMS

- objects you can manage, 1-2
- related documentation, 1-9
- viewing and maintaining multiple CA-IDMS systems, 1-7
- what you need to know, 1-8

CA-IDMS Visual DBA

- Application window, 2-6
- customizing the user interface, 1-8
- features, 1-1
- installing on the Mainframe, 2-2
- installing on the PC, 2-1
- Reinstalling, B-2
- securing objects, 2-12
- software requirements, 2-1
- starting your DBA session, 2-6
- uninstalling on the mainframe, B-1

Uninstalling on the PC, B-1

Command consoles, 1-7

D

Database Object Manager window, 2-6, 3-1

- Detail Information pane, 3-2
- Object Tree pane, 3-2
- opening, 2-11

Datasource

- adding, 2-10
- connecting to, 2-9

Detail Information pane, 3-2

Detail Information Pane tabs

- Bar, 3-27
- Grantee, 3-23
- Pie, 3-25
- Privilege, 3-24
- Property, 3-18
- Raw Prop, 3-20
- Rows, 3-22

Display Options

- setting, 3-29

Documentation, related, 1-9

DOM Window toolbar, 3-3

E

Environments
creating, opening, saving, 3-30

G

Getting help, 1-8
Grantee tab, 3-23
Granting and revoking object privileges, 2-15

I

Installing
CA-IDMS Visual DBA on the Mainframe, 2-2
CA-IDMS Visual DBA on the PC, 2-1

M

Managing objects using drag and drop, 1-7

N

Nodes window, 2-6
Nodes Window toolbar, 2-8

O

Object syntax
viewing, executing, and logging, 3-13
Object tree
expanding and collapsing, 3-4
refreshing, 3-13
Object Tree pane, 3-2

Objects
assigning privileges to, 2-14
Displaying attributes, 3-16
finding, 3-12
restricting access, 2-13
selecting and copying, 3-12
you can manage, 1-2

Opening a Database Object Manager window, 2-11

P

Pie tab, 3-25
Preferences
setting, 3-29
Privilege tab, 3-24
Property tab, 3-18

R

Raw Prop tab, 3-20
Reinstalling
CA-IDMS Visual DBA, B-2
Related documentation, 1-9
Restart from Position, 3-6
Rows tab, 3-22

S

Scratchpad, 3-9
Securing CA-IDMS Visual DBA objects, 2-12, 2-13
Software requirements, 2-1
Starting your DBA session, 2-6

T

Tear-Out Window, 3-7

Toolbars

DOM Window, 3-3

Nodes Window, 2-8

Workspace, 2-7

Tree Objects

manipulating, 3-10

U

Uninstalling

CA-IDMS Visual DBA on the mainframe, B-1

CA-IDMS Visual DBA on the PC, B-1

User interface

customizing, 1-8

V

Viewing and maintaining multiple CA-IDMS systems, 1-7

W

Workspace Environment, 2-6

Workspace toolbar, 2-7