

CA IDMS™ Visual DBA

User Guide

Version 18.0.00



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or 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 YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2011 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA IDMS™

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Contents

Chapter 1: Introduction to CA IDMS Visual DBA 9

Features.....	10
CA IDMS Objects.....	11
Manipulate CA IDMS Objects	24
Manage Objects Using Drag-and-Drop.....	24
Enhanced Object Security	24
Multiple CA IDMS Systems Viewing and Maintenance	25
Increase Productivity	25
Command Consoles.....	25
User Interface Customization.....	25
Prerequisites	26
Help Information	26

Chapter 2: Enhanced Object Security 27

Overview	28
Enable and disable Enhanced Object Security	28
Enable Enhanced Object Security for an IDMS CV	29
Disable Enhanced Object Security for an IDMS CV	29
Define the VIDMSR17 attribute in USER or SYSTEM PROFILE	29
Syntax.....	29
Parameters.....	30
Usage.....	30
Examples	31
Define the Visual DBA profile.....	31
Syntax.....	31
Parameters.....	32
Usage.....	34
Examples	35
VDBA r16 compatibility	35
Programmer	37
End User	38
Operator.....	39
Security Administrator	40
DC Administrator.....	41
Data Administrator.....	42
Default.....	44

CA IDMS Visual DBA Objects Security	45
Grant Access to Objects	46
Privileges to View Object Instances	46
Privileges to Update Object Instances and Use Command Console	46
Privileges to Create, Alter, and Drop Object Instances	47
Privileges to Grant and Revoke Object Privileges	47

Chapter 3: Start Your CA IDMS Visual DBA Session **49**

The Application Window	49
The Workspace Toolbar	50
The Nodes Window Toolbar	52
Connect to a Datasource/Node	53
Add a New Datasource/Node	53
Open a Database Object Manager Window	56
What's Next	57

Chapter 4: A Tour of the Database Object Manager Window **59**

The DOM Window	60
Use the DOM Window Toolbar	62
Expand and Collapse the Object Tree	63
Expand the Tree	63
Expand One Level at a Time	64
Expand Multiple Levels	64
Infinite Drill	64
Collapse the Tree	65
Change the Tree Structure	65
Restart from Position	65
Tear-Out Window	67
Scratchpad	68
Manipulate Tree Objects	70
Find Objects	71
Multiple Objects Selection	72
Object Syntax Viewing, Execution, and Logging	72
Refresh the Tree	73
Connect to Multiple CA IDMS Systems	74
Display Object Attributes in the Detail Information Pane	75
Tab Examples	78
Set Display Options for Your Session	96
Create, Open, and Save Environments	97
Try It Out Yourself	97

Chapter 5: Command Console	101
Command Console Workspace	102
Command Console Toolbar	102
Run a Command Script	105
Use the SQL Assistant.....	107
Set Command Console Preferences	111
Using the Editor	113
Editing functions and Accelerator Keys.....	114
Drag-and-Drop Support.....	114
Appendix A: Use the Demo180.cfg File	115
Appendix B: Third Party Software Acknowledgements	117
Index	119

Chapter 1: Introduction to CA IDMS Visual DBA

CA IDMS Visual DBA is a robust, comprehensive tool that enables you to view and manage CA IDMS objects from a single, object-oriented graphical user interface (GUI). Because of the use of customizable profiles, it allows an end-user, a programmer, a security administrator, a DC administrator, a data administrator, and a database administrator (DBA) to perform essential, everyday tasks quickly with 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 view and/or manage your database objects, such as learning the syntax for multiple online compilers, DCMT commands, or the SQL language. 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.

This section contains the following topics:

[Features](#) (see page 10)

[CA IDMS Objects](#) (see page 11)

[Manipulate CA IDMS Objects](#) (see page 24)

[Manage Objects Using Drag-and-Drop](#) (see page 24)

[Enhanced Object Security](#) (see page 24)

[Multiple CA IDMS Systems Viewing and Maintenance](#) (see page 25)

[Increase Productivity](#) (see page 25)

[Command Consoles](#) (see page 25)

[User Interface Customization](#) (see page 25)

[Prerequisites](#) (see page 26)

[Help Information](#) (see page 26)

Features

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

- Display all CA IDMS objects in a hierarchical tree. You can create, alter, drop, display objects, and assign privileges for each object in the tree.
- Define user profiles which limit the exploration of CA IDMS objects to those needed by the user.
- Connect to multiple CA IDMS systems simultaneously (r16, r17 or Version 18.0.00) during one CA IDMS Visual DBA session.
- View and search the CA IDMS log to obtain important information about a CA IDMS central version and its runtime objects.
- Increase productivity and decrease your learning curve because you no longer need to know the syntax of multiple online compilers or DCMT commands or the SQL language.
- 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.
- Use a PC text editor with support for copy, cut, paste, undo, redo, find, replace, drag-and-drop capability with shortcut keys for common edit functions.

Note: Pie charts for monitoring program and storage pool use, and for the database space use are available, similar to how bar diagrams are available for statistical information and table/record space distribution.

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

- Drill-down icons.
- Pop-up 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.
- Auto hiding panes.
- Tab bar to quickly access a window.
- Tooltips with extensive description.
- Extensive context sensitive help.

CA IDMS Objects

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

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

The following list shows the objects that can be explored using CA IDMS Visual DBA.

Object Id and Name

1	Dictionary
2	SQL Schema
3	Table in Schema
4	Index
5	View on Table
6	Constraint Table referenced
7	Constraint Table referencing
8	Access Module for Table
9	Calc Key
10	Column
11	Check Condition
12	Grantee on Table

13-24	Grantees
25	View in Schema
26	View Component
27	Access Module for View
28	Column of View
29	View Definition
30	Grantee on View
31-41	Grantees
42	Table Procedure in Schema
43	Key in Table Procedure
44	Parameter of Table Procedure
45	Access Module for Table Procedure
46	Grantee on Table Procedure
47-57	Grantees
58	Procedure in Schema
59	Key in Procedure
60	Parameter of Procedure
61	Routine Body of Procedure
62	Access Module for Procedure
63	Grantee on Procedure
64-74	Grantees
75	Function in Schema
76	Parameter of Function
77	Routine Body of Function
78	Access Module for Function
79	Grantee on Function
80-90	Grantees
91	Non SQL Table in Schema
92	Column of non SQL Table
93	Access Module using non SQL Table
94	Grantee on non SQL Table
95-100	Grantees
101	Constraint
102	Referencing Table in Constraint
103	Referenced Table in Constraint
104	Access Module
105	Relational Command Module in AM
106	Schema Mapping of AM
107	Area with Ready Mode in AM
108	Table accessed
109	View accessed
110	Table Procedure accessed
111	Procedure accessed
112	Function accessed
113	Non SQL Table accessed
114	Grantee on Access Module
115-120	Grantees
121	Grantee for Schema
122-127	Grantees

128	Table Like
129	Table
130	View
131	Table Procedure
132	Procedure
133	Function
134	Non SQL Table
135	Relational Command Module
136	Non SQL Schema
137	Area in Non SQL Schema
138	Area Procedure
139	Record in Area
140	Record
141	Record Procedure
142	Data Compression Table
143	Record Synonym
144	Record Element
145	Element Synonym
146	Element Description
147	Comment
148	Definition
149	Index Key
150	Olq Header
151	Culprit Header
152	Occurs Depending On
153	Element Value
154	Value
155	External Picture
156	Edit Table Value
157	Code Table Value
158	Element Indexed By
159	Cobol Indexed By
160	Set owned by Record
161	Set Record is member
162	Structure Shared By
163	Set
164	Owner Record of Set
165	Member Record of Set
166	Record Procedure
167	Data Compression Table
168	Record Synonym
169	Record Element
170	Set owned by Record
171	Set Record is member
172	Record Control Key
173	Record Foreign Key
174	Subschema
175	Area in Subschema
176	Record in Subschema

177	Element of Subschema Record
178	Set in Subschema
179	Load Module
180	Program using Subschema
181	Area used in Program
182	Area Statistics for Program
183	Record in Area
184	Record used in Program
185	Record Statistics for Program
186	Element of Subschema Record
187	Set used in Program
188	Set Statistics for Program
189	Owner Record of Set
190	Member Record of Set
191	Class/Attribute
192	Comment
193	Registree for Subschema
194	Registree for All
195	Registree for .Update
196	Registree for ..Delete
197	Registree for ..Modify
198	Registree for ...Display
199	Registree for Public Access
200	Responsible for Subschema
201	Responsible for Creation
202	Responsible for Deletion
203	Responsible for Update
204	Responsible for None
205	Class/Attribute
206	Comment
207	Grantee for Use
208	Registree for Schema
209	Registree for All
210	Registree for .Update
211	Registree for ..Delete
212	Registree for ..Modify
213	Registree for ...Display
214	Registree for Public Access
215	Responsible for Schema
216	Responsible for Creation
217	Responsible for Deletion
218	Responsible for Update
219	Responsible for None
220	IDD Class & Record & Module
221	Class
222	Attribute
223	IDD Record
224	IDD Record Synonym
225	IDD Record Element

226	IDD Element Synonym
227	IDD Element Description
228	IDD Comment
229	IDD Definition
230	IDD Index Key
231	IDD Olq Header
232	IDD Culprit Header
233	IDD Occurs Depending On
234	IDD Element Value
235	IDD Value
236	IDD External Picture
237	IDD Edit Table Value
238	IDD Code Table Value
239	IDD Element Indexed By
240	IDD Cobol Indexed By
241	Module
242	Module Text
243	Assembler Module
244	Cobol Module
245	Culprit Module
246	DC Module
247	OCF Module
248	OLQ Module
249	PL/I Module
250	Process Module
251	Segment
252	File
253	Area in File
254	DMCL using File
255	Area
256	File in Area
257	DMCL using Area
258	Schema using Area as default
259	Table stored in Area
260	Index stored in Area
261	Symbolic Parameter
262	Subarea
263	Displacement
264	Index
265	Grantee on Area
266	Grantee for DBAREAD
267	Grantee for DBAWRITE
268	Grantee for Use
269	DMCL including Segment
270	DBName including Segment
271	Schema referencing Segment
272	Grantee for Segment
273-279	Grantees
280	DBTable

281	Subschema Mapping
282	DBName
283	Segment in DBName
284	Subschema Mapping in DBName
285	Schema referencing DBName
286	DBTable including DBName
287	Grantee for DBName
288-294	Grantees
295	DMCL using DBTable
296	DBGroup in DBTable
297	Grantee for DBTable
298-303	Grantees
304	DMCL
305	Segment in DMCL
306	Segment.File Override
307	Segment.Area Override
308	File in DMCL
309	Area in DMCL
310	Shared File in DMCL
311	Shared Area in DMCL
312	Shared Cache in DMCL
313	Buffer
314	Database Buffer
315	Journal Buffer
316	Journal
317	Archive Journal
318	Disk Journal
319	Tape Journal
320	Grantee for DMCL
321-326	Grantees
327	System
328	Load List
329	Component of Loadlist
330	Program
331	Queue
332	Autotask
333	Task
334	Storage Pool
335	XA Storage Pool
336	Line
337	Pterm in Line
338	Lterm in Pterm
339	Lterm in Line
340	Pterm
341	Lterm
342	Printer
343	Destination
344	Lterm in Destination
345	Printer in Destination

346	User in Destination
347	Node
348	Resource Table
349	Local Dictionary in Res Table
350	Dictionary via Node in Res Table
351	Destination Node in Res Table
352	Rununit
353	Map Table
354	Entry of Map Table
355	Key Table
356	Application
357	SQL Cache
358	Except Connect Name
359	TCP/IP
360	Stack
361	Except
362	System Resource & Profile
363	Category
364	Access Module in Category
365	Load Module in Category
366	Program in Category
367	Queue in Category
368	Rununit in Category
369	Task in Category
370	Grantee for Execute
371	Activity
372	Grantee for Execute
373	System Id
374	Signon System Profile.User
375	Grantee for System Id
376-381	Grantees
382	System Profile
383	Attribute
384	Signon System Id.User
385	Grantee on System Profile
386-390	Grantees
391	User Profile
392	Attribute
393	Grantee on User Profile
394-398	Grantees
399	Group
400	User in Group
401	Privilege for Group
402	Access Module granted
403-408	Privileges
409	Activity granted
410	Area granted
411-413	Privileges
414	Category granted

415 DBName & Segment granted
416-422 Privileges
423 DBTable granted
424-429 Privileges
430 DCADMIN granted ?
431 DMCL granted
432-437 Privileges
438 Group granted
439-443 Privileges
444 Non SQL Schema granted
445 Privilege for Use
446 Non SQL Table granted
447-452 Privileges
453 Schema granted
454-459 Privileges
460 SYSADMIN granted ?
461 System Id granted
462-467 Privileges
468 System Profile granted
469-473 Privileges
474 Table granted
475-486 Privileges
487 Table Procedure granted
488-498 Privileges
499 Procedure granted
500-510 Privileges
511 Function granted
512-522 Privileges
523 User granted
524-528 Privileges
529 User Profile granted
530-534 Privileges
535 View granted
536-546 Privileges
547 Grantee on Group
548-552 Grantees
553 Central User
554 Group User belongs
555 Signon System Id.System Profile
556 Privilege for User
557 Access Module granted
558-563 Privileges
564 Activity granted
565 Area granted
566-568 Privileges
569 Category granted
570 DBName & Segment granted
571-577 Privileges
578 DBTable granted

579-584 Privileges
585 DCADMIN granted ?
586 DMCL granted
587-592 Privileges
593 Group granted
594-598 Privileges
599 Non SQL Schema granted
600 Privilege for Use
601 Non SQL Table granted
602-607 Privileges
608 Schema granted
609-614 Privileges
615 SYSADMIN granted ?
616 System Id granted
617-622 Privileges
623 System Profile granted
624-628 Privileges
629 Table granted
630-641 Privileges
642 Table Procedure granted
643-653 Privileges
654 Procedure granted
655-665 Privileges
666 Function granted
667-677 Privileges
678 User granted
679-683 Privileges
684 User Profile granted
685-689 Privileges
690 View granted
691-701 Privileges
702 Grantee on User
703-707 Grantees
708 Dictionary User
709 Unity Type Authority
710 Authority on Password
711 Authority on Culprit
712 Authority on OLQ
713 Authority on ADS
714 Authority on Load Module
715 Authority on Class Attribute
716 Authority on Class
717 Authority on Attribute
718 Authority on DC
719 Authority on Destination
720 Authority on Line
721 Authority on Logical-terminal
722 Authority on Map
723 Authority on Message

724	Authority on Panel
725	Authority on Physical-terminal
726	Authority on Queue
727	Authority on Task
728	Authority on IDD
729	Authority on Element
730	Authority on Entry Point
731	Authority on File
732	Authority on Module
733	Authority on Process
734	Authority on Qfile
735	Authority on Table
736	Authority on Program
737	Authority on Record
738	Authority on Report
739	Authority on Transaction
740	Authority on System
741	Authority on User
742	Authority on Idms
743	Authority on Schema
744	Authority on Subschema
745	Authority on All
746	Registration for Dictionary User
747	Non SQL Schema registered
748	Registration for All
749	Registration for .Update
750	Registration for ..Delete
751	Registration for ..Modify
752	Registration for ...Display
753	Registration for Public Access
754	Subschema registered
755	Registration for All
756	Registration for .Update
757	Registration for ..Delete
758	Registration for ..Modify
759	Registration for ...Display
760	Registration for Public Access
761	Responsibility of Dictionary User
762	Non SQL Schema responsibility
763	Responsibility for Creation
764	Responsibility for Deletion
765	Responsibility for Update
766	Responsibility for None
767	Subschema responsibility
768	Responsibility for Creation
769	Responsibility for Deletion
770	Responsibility for Update
771	Responsibility for None
772	Grantee for Administration

773	Grantee for DCADMIN
774	Grantee for SYSADMIN
775	CV DBTable
776	CV Subschema Mapping
777	CV DBName
778	CV Segment in DBName
779	CV Subschema Mapping in DBName
780	CV DBGroup
781	CV Backend in DBGroup
782	CV DMCL
783	CV Segment
784	CV File in Segment
785	CV Area in Segment
786	CV DBName including Segment
787	CV File
788	CV Area
789	SMP Space Report
790	SMP Avail. Space Distribution
791	SMP File Space Report
792	SMP Avail. Space Distribution
793	Full Space Report
794	Avail. Space Distribution
795	Record Space Distribution
796	File Space Report
797	Avail. Space Distribution
798	CV Database Buffer
799	CV Journal Buffer
800	CV Journal
801	CV System
802CV	Load List
803CV	Program
804CV	Program Module
805CV	Program Pool new Sysgen
806CV	Reent Prog Pool new Sysgen
807CV	Queue
808CV	Task
809CV	Storage Pool
810CV	XA Storage Pool
811CV	XA Storage Pool new Sysgen
812CV	Line
813CV	Pterm in Line
814CV	Lterm in Line
815CV	TCP/IP Info
816CV	Line new Sysgen
817CV	Pterm
818CV	Lterm
819CV	Printer
820CV	Destination
821CV	Node

822CV	Resource Table
823CV	Rununit
824CV	Rununit Detail
825CV	SQL Cache
826CV	Except Connect Name
827CV	TCP/IP
828	Services
829	Stacks
830	SYSIDMS
831	Activity
832A	Active Program
833	Active Program Detail
834	Active Queue
835	Active Storage
836	Active Storage Detail
837	Active Task
838	Active Task Detail
839	Lterm using Scratch
840	User signed on
841	Search Log
842	Message in Log
843	Snap/Dump in Log
844	Trace in Log
845	Central Version
846	Autotune
847	Change Tracking
848	Change Tracking Files
849	Data Sharing Summary
850	Data Sharing Group Member
851	Data Sharing LIST Structure
852	Data Sharing LOCK Structure
853	Distributed Resource Manager
854	Dist Resource Manager Detail
855	Distributed Transaction
856	Dist Transaction Detail
857	Deadlock
858	DCMT Command Outstanding
859	External Rununit
860	Journal Status
861	Pending Transactions
862	Lock on Area
863	Lock on Lterm
864	Log
865	Log Driver
866	Shared Cache
867	File in Shared Cache
868	SQL Cache Content
869	SQL Cache Entry
870	Trace DB

871	Transaction
872	Transaction Sharing
873	DC
874	ADSO
875	DDS Lines
876	DDS PTerms
877	PTerms Attributes
878	Limit
879	Loadlib
880	LU
881	Modeent of LU
882	MPMode
883	MT Queue Depth
884	Nucleus Reload
885	Reply pending
886	Report Class/Destination
887	Report
888	Scratch
889	SNA Pterm
890	Snap
891	SubTask
892	TCP/IP Sockets
893	Time
894	Time initiated Task
895	Trace System
896	UCF Terminal
897	Statistics
898	Statistics Segment
899	Statistics File in Segment
900	Statistics Area in Segment
901	Statistics File
902	Statistics Area
903	Statistics Buffer
904	Statistics Interval
905	Statistics Roll
906	Statistics Lock
907	Statistics System
908	Statistics TCP/IP

Notes:

- The profile assigned to a user usually limits the set of accessible objects.
- Grantee and Privilege objects have been collapsed to limit the size of the list.

Manipulate 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 pop-up menu that enables you to create, alter, drop the object, and assign privileges to it. Alternatively, you can use the Edit menu or the buttons on the toolbar.

Manage 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 subbranch or from the Detail Information Pane to any other subbranch 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 subbranch 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 technique greatly simplifies tasks like test and production database maintenance.

Enhanced Object Security

You can optionally setup IDMS Central Version to use enhanced object security for CA IDMS Visual DBA access. With this option selected, CA IDMS Visual DBA will require the existence of a CA IDMS Visual DBA profile for the user accessing the Central Version.

CA IDMS Visual DBA uses the value of the session attribute VIDMSR17 to locate and retrieve an IDD module that contains the CA IDMS Visual DBA profile to be used for the CA IDMS Visual DBA session. The IDD module contains the root entries, optionally with filters and permissions that the user is presented after a connection has been established. Only those root objects and associated child objects can be viewed and manipulated in accordance with the defined permissions. The IDD module also contains a list of command console processors with associated dictionaries that can be run by the user.

Multiple CA IDMS Systems Viewing and Maintenance

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.

Increase Productivity

With CA IDMS Visual DBA, you can manage your CA IDMS objects 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. The results can be dragged and dropped to external tools, such as word or document processors and spreadsheets.

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

User Interface Customization

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 views of the tree optimized for different tasks. 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.

Prerequisites

This guide assumes that you are already familiar with the concepts and features associated with your role as a CA IDMS user.

For more information about CA IDMS , see the *CA IDMS* documentation.

In addition to an understanding of the CA IDMS concepts and features for your role, this guide assumes that you are familiar with Windows terminology and navigational techniques. This includes how to work with standard Windows items like menus, dialogs, 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.

Help Information

Online help is provided, 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 dialogs and active windows.

Chapter 2: Enhanced Object Security

This chapter explains Enhanced Object Security, a feature that can optionally be activated for an IDMS Central Version. Because Enhanced Object Security is defined at the IDMS CV side, a client CA IDMS Visual DBA can access a mix of IDMS CVs that are either configured with or without Enhanced Object Security.

Only IDMS Central Versions that control access from CA IDMS Visual DBA through Enhanced Object Security require the additional configuration steps described in this chapter.

At any time CA IDMS Visual DBA access control through Enhanced Object Security can be disabled or enabled for an IDMS CV.

At the end of this chapter, security aspects of the SQL objects used by CA IDMS Visual DBA are discussed.

This section contains the following topics:

[Overview](#) (see page 28)

[Enable and disable Enhanced Object Security](#) (see page 28)

[Enable Enhanced Object Security for an IDMS CV](#) (see page 29)

[Disable Enhanced Object Security for an IDMS CV](#) (see page 29)

[Define the VIDMSR17 attribute in USER or SYSTEM PROFILE](#) (see page 29)

[Define the Visual DBA profile](#) (see page 31)

[CA IDMS Visual DBA Objects Security](#) (see page 45)

Overview

To streamline access control to CA IDMS through CA IDMS Visual DBA, to provide an additional level of object security, and to present to the user a DOM tree customized for his tasks, Enhanced Object Security is available. This optional feature is based on CA IDMS Visual DBA profiles that are stored in CA IDMS on the mainframe. Setting up Enhanced Object Security consists of configuring the mainframe component of CA IDMS Visual DBA. This involves defining CA IDMS Visual DBA profiles and assigning profiles to users and/or groups.

Linking or associating profiles with users in CA IDMS Visual DBA is based on the CA IDMS session profile: when a user connects to an IDMS CV a session is established with an associated profile. An IDMS session profile can have multiple user defined attributes. The use of Enhanced Object Security requires the existence of an attribute with name VIDMSR17. The value assigned to this attribute is the dictionary and module name of an IDD module that contains the CA IDMS Visual DBA profile for the current session.

As any session attribute, the VIDMSR17 session attribute is created during signon of a user if the user's *user profile* or the *system profile* contains a VIDMSR17 attribute definition.

Implementing Enhanced Object Security through the use of CA IDMS Visual DBA profiles is the easiest way to control access to CA IDMS by CA IDMS Visual DBA.

The SQL objects defined for use by CA IDMS Visual DBA can be used by any CA IDMS client and therefore access to these objects should also be controlled.

Enable and disable Enhanced Object Security

The use of Enhanced Object Security is optional and is controlled by the presence of the parameter `NO_SERVER_ROLE` in the definition of the table procedure `SYSCA.VDBA_VERSION5` in the dictionary specified in the Virtual Node or ODBC data source definition for the IDMS Central Version. If the parameter `NO_SERVER_ROLE` is present, Enhanced Object Security is not active, otherwise it is.

Enable Enhanced Object Security for an IDMS CV

Submitting the following SQL DDL statements against the dictionary specified in the Virtual Node or ODBC data source definition for the IDMS CV will enable Enhanced Object Security.

```
drop table procedure SYSCA.VDBA_VERSION5;
create table procedure SYSCA.VDBA_VERSION5 (
CA_IDMS_VDBA_V_5_0 char(5)
, VDBA_EXE_BUILD_1 int
, VDBA_DLL_BUILD_1 int
, VDBA_VIEW_BUILD_1 int
) EXTERNAL NAME VDBAVER;
```

Disable Enhanced Object Security for an IDMS CV

Submitting the following SQL DDL statement against the dictionary specified in the Virtual Node or ODBC Datasource definition for the IDMS CV will disable Enhanced Object Security.

```
alter table procedure SYSCA.VDBA_VERSION5
add (NO_SERVER_ROLE int);
```

Note: The installation of CA IDMS Visual DBA on the mainframe requires the execution of an SQL script that contains the definition of the above table procedure SYSCA.VDBA_VERSION5. The default is to disable Enhanced Object Security but this can have been overwritten through configuration of the SQL script.

Define the VIDMSR17 attribute in USER or SYSTEM PROFILE

An attribute is a keyword and an associated value for the keyword. Attributes are defined for user profiles through the CREATE USER PROFILE or the ALTER USER PROFILE statements. Attributes for system profiles are defined through the CREATE SYSTEM PROFILE or the ALTER SYSTEM PROFILE statements.

The syntax rules for the attribute-specification of the CA IDMS Visual DBA VIDMSR17 attribute in either user or system profiles are:

Syntax

```
>— VIDMSR17 — = — ‘ ———— dict-name — . ———— module-name — ‘ —><
```

Parameters

dict-name

Specifies the dictionary from which the CA IDMS Visual DBA profile module is retrieved.

module-name

Specifies the name of the CA IDMS Visual DBA profile module.

Usage

The dictionary from which the CA IDMS Visual DBA module is retrieved can be any dictionary with a DDLML area. The dictionary can, but does not need to have the CA IDMS Visual DBA components installed. The dictionary does not need to be defined as a CA IDMS Visual DBA accessible dictionary.

A user who is using the CA IDMS Visual DBA profile defined in the specified module must have IDD DISPLAY access for that module. To prevent a user from modifying his CA IDMS Visual DBA profile he must not have any type of update access to the specified module.

The VIDMSR17 attribute can be created either in user or system profiles. The option to use one or the other depends on your current use of profiles. If you are already using user profiles, it seems natural to simply add the VIDMSR17 attribute to the different profiles specifying the location and name of the IDD module with the CA IDMS Visual DBA profile as appropriate for the user profile.

If user profiles are not yet used, either user profiles can be created that correspond to the CA IDMS Visual DBA profiles or system profiles can be updated to include the VIDMSR17 attribute.

Note: For more information about defining user profiles and attributes and associating users with profiles, see the *CA IDMS Security Administration Guide*. For more information about defining system profiles and attributes, see the *CA IDMS System Tasks and Operator Commands Guide*.

Examples

In the following example, the user profile ENDUSER is first created and then altered to include the attribute VIDMSR17. The attribute value specifies SYSTEM as the dictionary name and VIDMS_ENDUSER as the module name for the IDD module that contains the CA IDMS Visual DBA profile. Finally the profile is associated with users Adam and Joe.

```
create user profile ENDUSER;
alter user profile ENDUSER
  attribute VIDMSR17 = 'SYSTEM.VIDMS_ENDUSER' OVERRIDE NO;
alter user Adam profile ENDUSER;
alter user Joe profile ENDUSER;
```

In this example, a system profile PROGRAMMER is first created and then altered to include the attribute VIDMSR17. The attribute value specifies SYSTEM as the dictionary name and VIDMS_PROGRAMMER as the module name for the IDD module that contains the CA IDMS Visual DBA profile. Finally the system profile is associated with users Julie and Nancy when they sign on to the system identified through resource SYSPROD.

```
create system profile PROGRAMMER;
alter system profile PROGRAMMER
  attribute VIDMSR17 = 'SYSTEM.VIDMS_PROGRAMMER' OVERRIDE NO;

grant signon on system SYSPROD profile PROGRAMMER to Julie, Nancy;
```

Note: The file vidmsv18.bcf in the CA IDMS Visual DBA product folder has many examples of CA IDMS Visual DBA profile modules, user profiles and attributes and syntax to associate users with user profiles.

Define the Visual DBA profile

The rules for defining the CA IDMS Visual DBA profile are described as follows. The rules specify the content of the IDD module that is being referred to by the VIDMSR17 attribute of CA IDMS Visual DBA user.

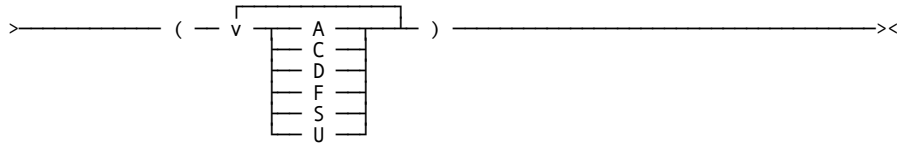
Syntax

```
>- v _____><
   |-----|
   | dom-object-profile-line |
   | cmd-console-profile-line |
   | * _____ |
   |-----| comment |><
```

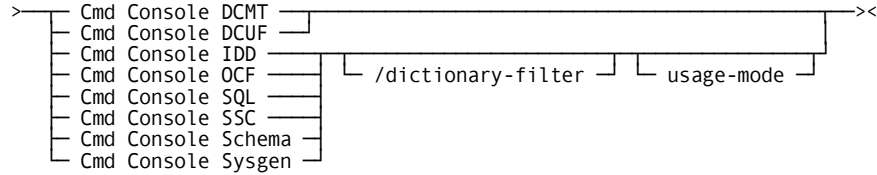
Expansion of dom-object-profile-line

```
>- dom-object _____><
   |-----| /filter | permitted-ops |><
   | v-[parent-instance] |><
```

Expansion of permitted-ops



Expansion of cmd-console-profile-line



Expansion of usage-mode



Parameters

dom-object-profile-line

Specifies a root branch of the DOM tree, an optional instance filter and also optional permitted operations. Each dom-object-profile-line must be completely coded on one, separate text line of the CA IDMS Visual DBA profile module. The order of the dom-object-profile-line lines is also the order of the root branches in the DOM tree. Space characters, except when used before or in the dom-object, indicate the start of comment text.

cmd-console-profile-line

Specifies the availability of a command console processor and optionally a filter to limit the CA IDMS Visual DBA list of available dictionaries. Each cmd-console-profile-line must be completely coded on one, separate text line of the CA IDMS Visual DBA profile module. Space characters, except when used before or in the starting 'Cmd Console' string, indicate the start of comment text.

dom-object

This is the name of the object that becomes a root branch of the DOM tree.

The name is formed of all characters starting with the first non-blank character up to either an open square bracket or “[”, an open parenthesis or “(”, a slash (“/”) or the end of the line. The name must match an object from the CA IDMS Visual DBA tree as listed in CA IDMS Objects in Chapter 1 “Introduction to CA IDMS Visual DBA” with either 0, 1, 2, or 3 level parents. An object must have as many parent instances specified as required by its parent level.

parent-instance

This is the instance of a parent object of the object defined as a root branch of the DOM tree. If more than one parent-instance needs to be specified because the parent level of the DOM object is greater than one, the parent-instance of the parent at the highest level must be specified first, then the one at the next level, and so on.

filter

Specifies a pattern (character string) that must be matched by the instances of the DOM object. The following wildcard characters are available:

? (question-mark)

Represents a single character.

*** (asterisk)**

Represents any string of zero or more characters

permitted-ops

Specifies the operations that are permitted on the DOM object. View or display is automatically permitted. The additional permissions are listed in the table:

Code	Operation	Description
A	Alter	The Alter dialog can be invoked and executed
C	Create	The Create dialog can be invoked and executed
D	Drop	The Drop dialog can be invoked and executed
F	Full	All dialogs can be invoked and executed, equivalent to “SU”
S	Security	The Grant, Revoke, Register, and Responsibilities dialogs can be invoked and executed. Object expansion contains Grantee, Registree, and Responsible where applicable.
U	Update	The Alter, Create, and Drop dialogs can be invoked and executed, equivalent to “ACD”

Note: In order to Alter, Create or Drop a Central User, a Group or a Dictionary User, the S(Security) permission is also required.

Cmd Console DCMT

Cmd Console DCUF

Cmd Console IDD

Cmd Console OCF

Cmd Console SQL

Cmd Console SSC

Cmd Console Schema

Cmd Console Sysgen

Specifies a command console and processor that can be executed.

dictionary-filter

Specifies a pattern (character string) that must be matched by the dictionary selected in the command console, in order to be able to execute the commands from the associated command console processor. The following wildcard characters are available:

? (question-mark)

Represents a single character.

*** (asterisk)**

Represents any string of zero or more characters.

(U)

If specified the dictionary is accessed in update mode, else in retrieval mode.

Usage

Dictionary access in Command Console

While the use of a dictionary-filter controls the dictionary against which a command stream is executed, it does not prohibit the user from entering CONNECT commands in an OCF command stream or SIGNON commands in IDD, Schema, SSC or SYSGEN command streams against any dictionary.

Similarly the absence of a usage-mode specification, implying a usage mode retrieval, does not prohibit the user from entering a SET SESSION READ WRITE command in OCF or SIGNON USAGE UPDATE command in IDD, Schema, SSC, or Sysgen.

Command consoles for the compiler processors IDD, Schema, SSC, and Sysgen should only be enabled if the access to all dictionaries is properly secured.

Note: When using the SQL processor it is not possible to issue a successful connect within the SQL command stream limiting the SQL commands to be executed against the specified dictionary only. Furthermore, while the SQL command stream can contain a SET SESSION READ WRITE command, it will not allow the user to execute SQL update statements if the dictionary access mode is not update.

Examples

The file VIDMSR18.bcf located in the CA IDMS Visual DBA product folder can be used as a template for defining CA IDMS Visual DBA profiles, users, and user profiles and for associating users with user profiles.

The following are examples of different CA IDMS Visual DBA profiles.

VDBA r16 compatibility

The following profile module can be used to provide compatibility with VDBA r16. The dom-object-profile-line's specify the same root branches as used by VDBA r16 with full permissions.

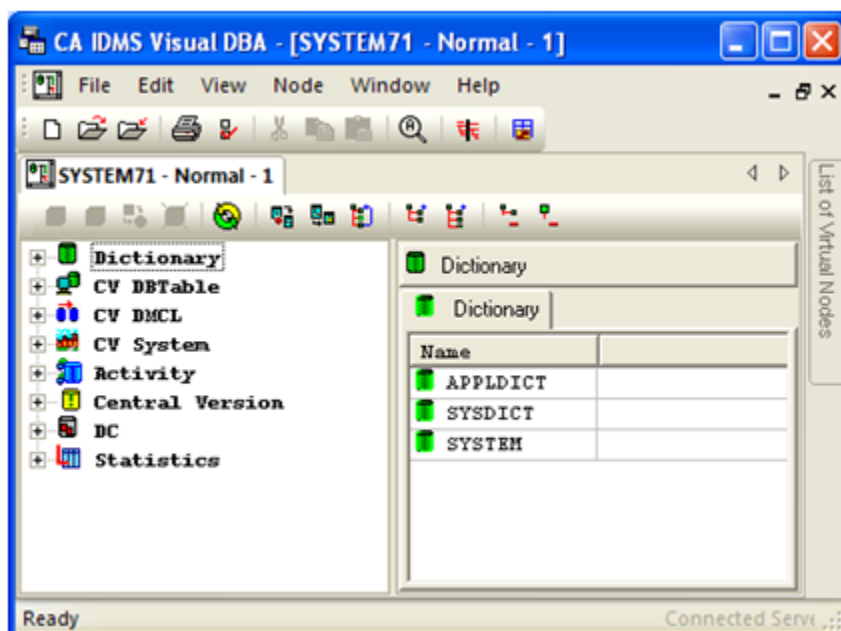
The cmd-console-profile-line's specify, where applicable a pattern that does not filter any dictionaries.

```
add module name is VIDMS_VDBAR16 version is 1
    include user JOESIXP registered for all
    public access allowed for DISPLAY
    module source follows
```

```
Dictionary(F)
CV DBTable(F)
CV DMCL(F)
CV System(F)
Activity(F)
Central Version(F)
DC(F)
Statistics(F)
```

```
*  
Cmd Console DCMT  
Cmd Console DCUF  
Cmd Console IDD/*(U)  
Cmd Console OCF/*(U)  
Cmd Console Schema/*(U)  
Cmd Console SQL/*(U)  
Cmd Console SSC/*(U)  
Cmd Console Sysgen/*(U)  
msend.
```

The CA IDMS Visual DBA tree after a successful connection looks like this, showing the same DOM tree as with VDBA r16.



Programmer

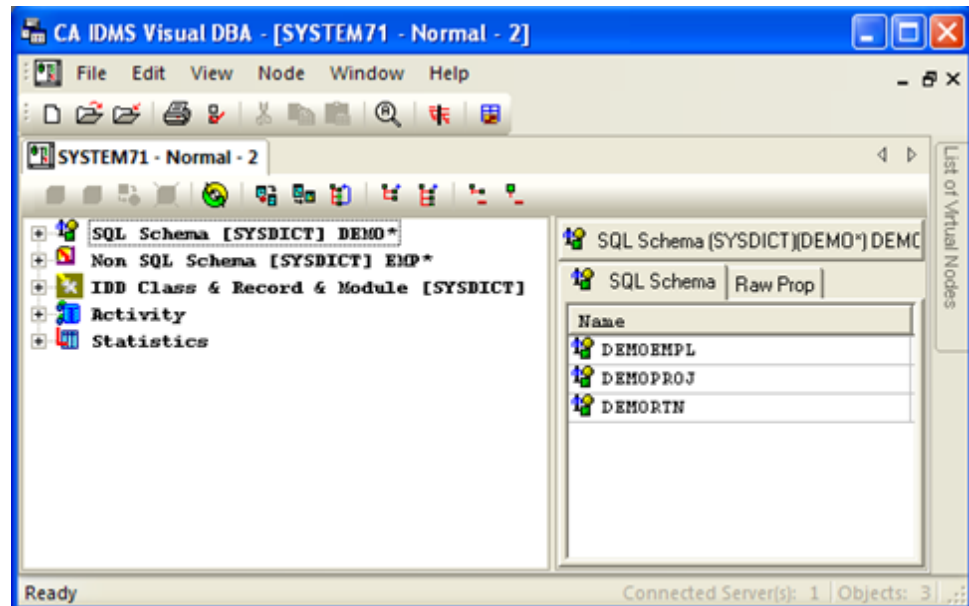
The following profile could be used for programmers that have only read access to SQL schema's in dictionary SYSDICT that start with DEMO and to Non-SQL Schema's in dictionary SYSDICT that start with EMP .

This profile does allow full control of all Records and Modules in dictionary SYSDICT.

Command consoles are not available when using this profile. Through the (F) specification for Activity, this profile allows for creating searches in to the online IDMS log.

```
add module name is VIDMS_PROGRAM_DEMO version is 1
      include user JOESIXP registered for all
      public access allowed for DISPLAY
      module source follows
SQL Schema[SYSDICT]/DEMO*
Non SQL Schema[SYSDICT]/EMP*
IDD Class & Record & Module[SYSDICT](F)
Activity(F)
Statistics
      msend.
```

The CA IDMS Visual DBA tree after a successful connection looks like this:



End User

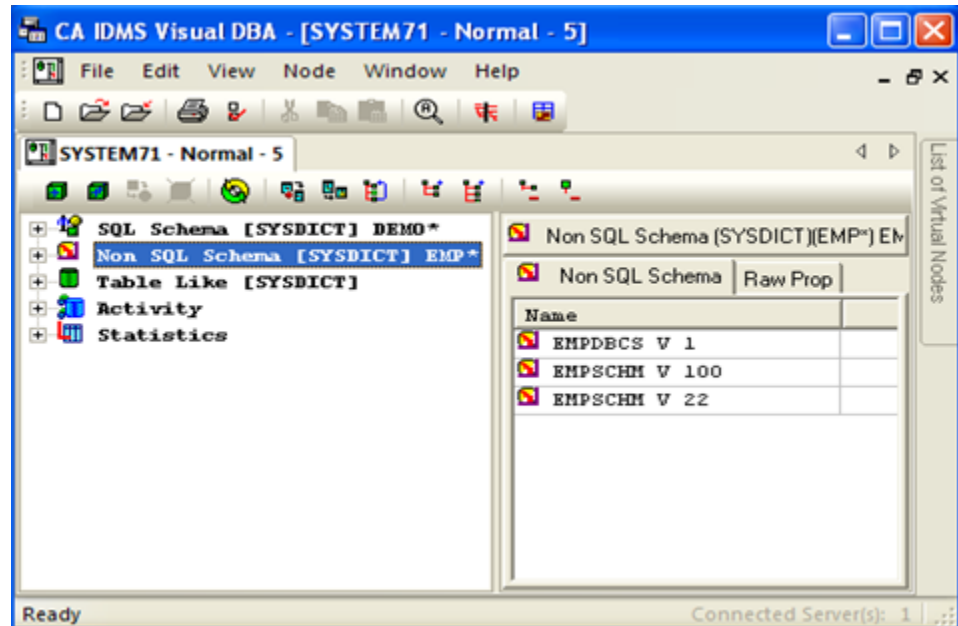
The following profile could be used by an end user who has full control of SQL Schema's that start with DEMO in dictionary SYSDICT and to Non-SQL Schema's in dictionary SYSDICT that start with EMP .

In addition this end user has read access to all Table Like objects in dictionary SYSDICT.

The only command console processor that can be used in the profile is SQL and only against dictionary SYSDICT. Through the (F) specification for Activity, this profile allows for creating searches in the online IDMS log.

```
add module name is VIDMS_ENDUSER version is 1
      include user JOESIXP registered for all
      public access allowed for DISPLAY
      module source follows
SQL Schema[SYSDICT]/DEMO*(F)
Non SQL Schema[SYSDICT]/EMP*(F)
Table Like[SYSDICT]
Activity(F)
Statistics
*
Cmd Console SQL/SYSDICT
      msend.
```

After a successful connection, expansion of SQL Schema [SYSDICT] DEMO* and selection of Non-SQL Schema [SYSDI CT] EMP*, the CA IDMS Visual DBA tree looks like:



Operator

The following profile could be used for an operator. Only run time objects Activity and Statistics are available. Assigning (F) Full control to Activity allows the operator to create searches in the log and to change max tasks of the CV.

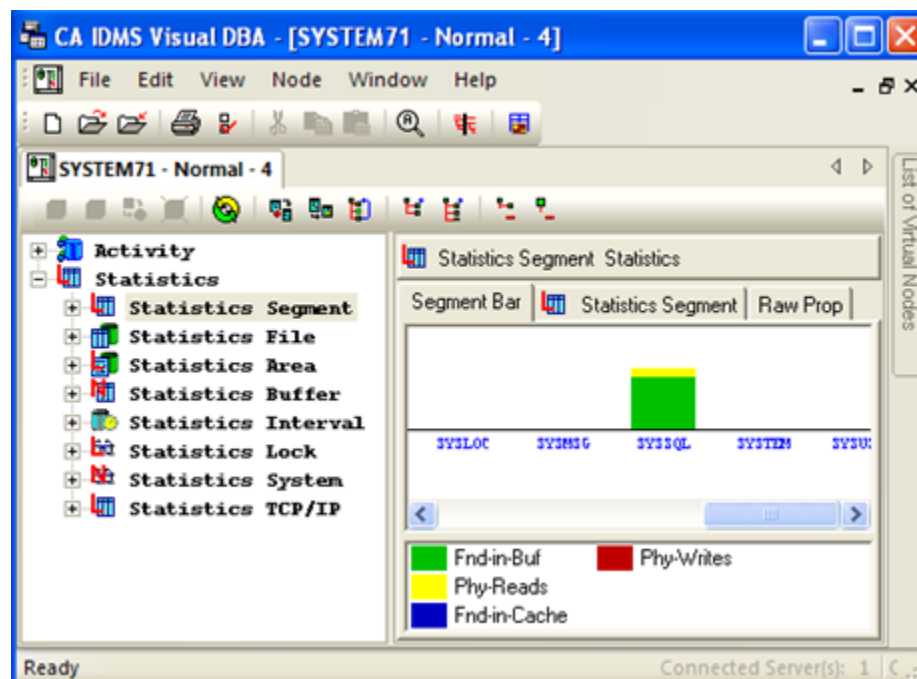
```
add module name is VIDMS_OPERATOR version is 1
    include user JOESIXP registered for all
    public access allowed for DISPLAY
    module source follows
```

Activity(F)

Statistics

msend.

The CA IDMS Visual DBA tree after a successful connection and expanding Statistics looks like this:



Security Administrator

The following profile could be used by a security administrator. Read access is given to all objects in all dictionaries. Security related objects can be fully operated on. Also full control is given to Search Log to allow searches in the online log.

Activity can be viewed. Command consoles are not available.

```
add module name is VIDMS_SECADMIN version is 1
    include user JOESIXP registered for all
    public access allowed for DISPLAY
    module source follows
```

Dictionary

System Resource & Profile[SYSDICT](F)

System Resource & Profile[SYSTEM](F)

Group[SYSDICT](F)

Central User[SYSDICT](F)

Dictionary User[SYSDICT](F)

Grantee for Administration[SYSDICT](F)

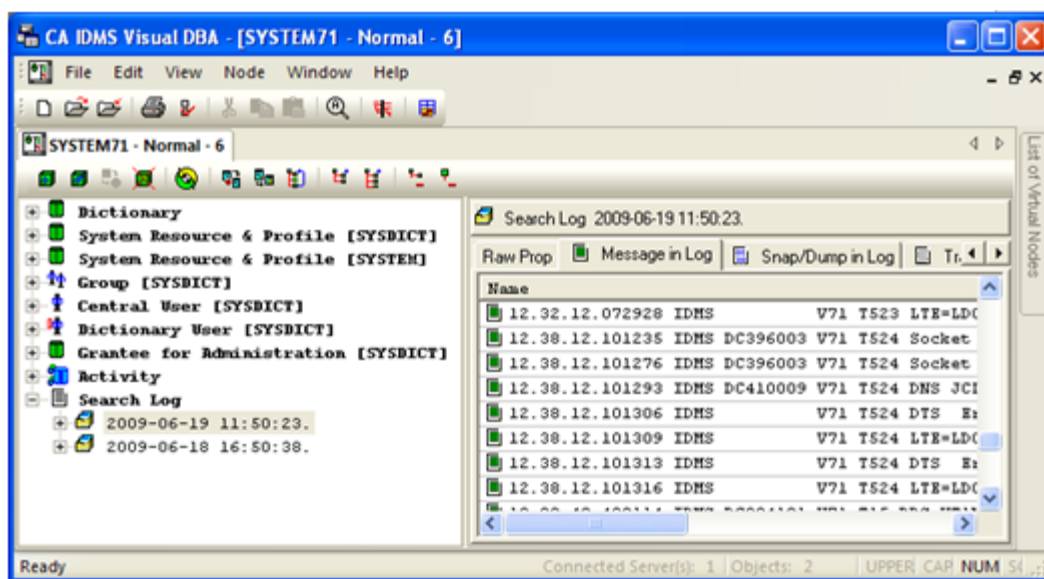
Activity

Search Log(F)

*

msend.

After successfully connecting and creating Search Log instances, the CA IDMS Visual DBA Tree and Detail Information Pane (DIP) look like this:

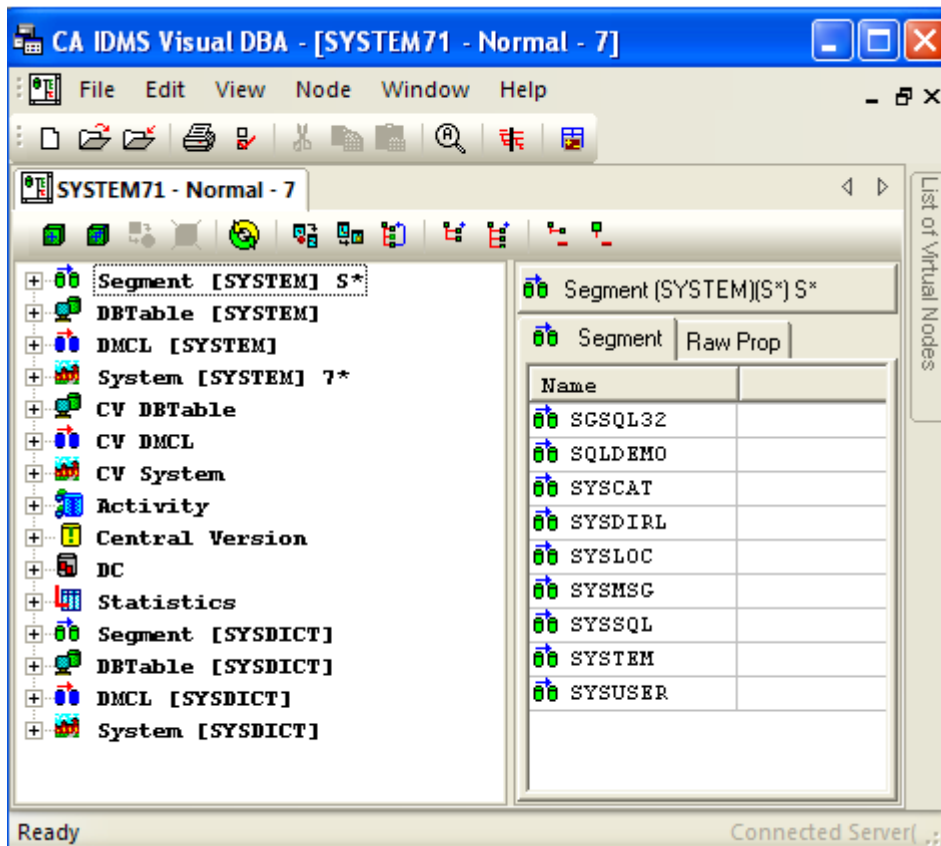


DC Administrator

The following profile could be used for a DC administrator. It allows full control in dictionary SYSTEM to Segment's that start with 'S', to System's that start with 7 and to all DBTable and DMCL objects. Full control is also given on the run time objects CV DBTable, CV DMCL, CV System, Activity, Central Version, DC, and Statistics. Read access is allowed on all Segment, DBTable, DMCL, and System objects in dictionary SYSDICT. Finally in a command console DCMT and DCUF processors are available.

```
add module name is VIDMS_DCADMIN version is 1
    include user JOESIXP registered for all
    public access allowed for DISPLAY
    module source follows
Segment[SYSTEM]/S*(F)
DBTable[SYSTEM](F)
DMCL[SYSTEM](F)
System[SYSTEM]/7*(F)
CV DBTable(F)
CV DMCL(F)
CV System(F)
Activity(F)
Central Version(F)
DC(F)
Statistics(F)
Segment[SYSDICT]
DBTable[SYSDICT]
DMCL[SYSDICT]
System[SYSDICT]
Cmd Console DCMT
Cmd Console DCUF
msend.
```

After successful connection and selecting Segment [SYSTEM] S*:

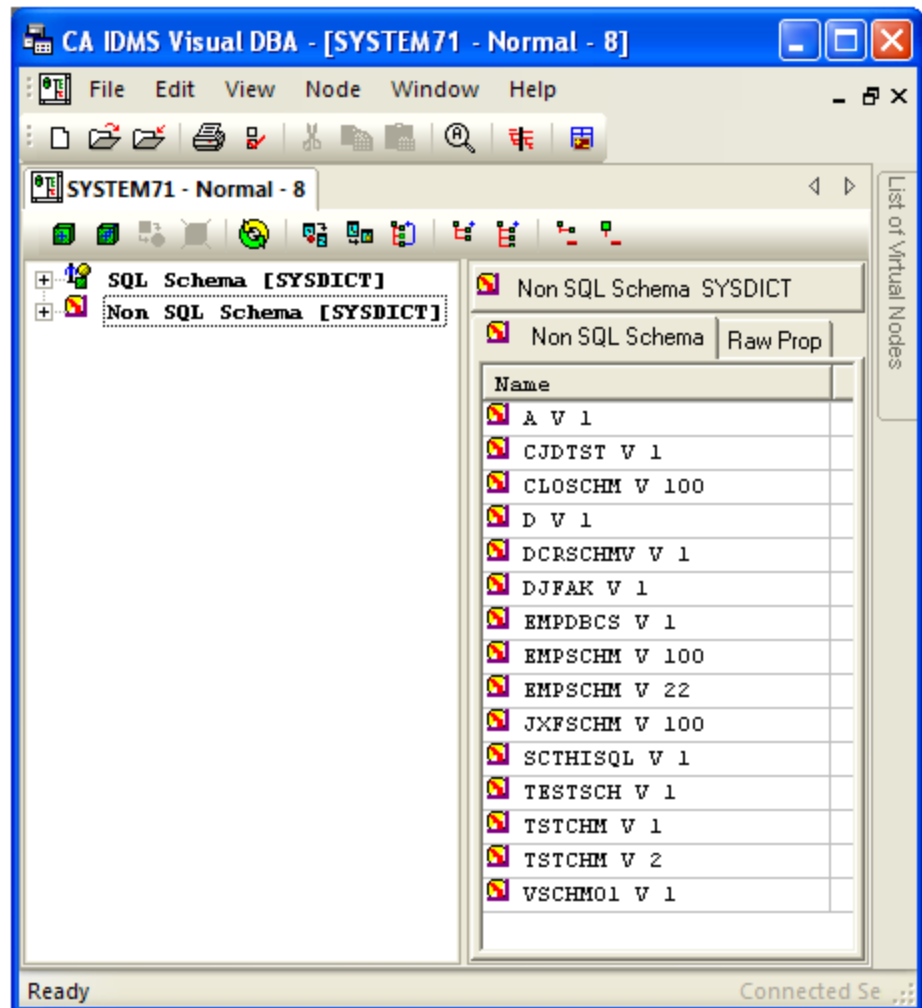


Data Administrator

The following profile could be used for a data administrator. Full control is available for all SQL Schema and Non-SQL Schema objects in dictionary SYSDICT.

```
add module name is VIDMS_DATAADMIN version is 1
    include user JOESIXP registered for all
    public access allowed for DISPLAY
    module source follows
SQL Schema[SYSDICT](F)
Non SQL Schema[SYSDICT](F)
    msend.
```

After a successful connection the CA IDMS Visual DBA tree would look like this:



Default

The following profile is used as a default profile. Users with this profile are given read access only to all objects in dictionary SYSDICT and to all run time objects in CV DBTable, CV DMCL, CV System, and Activity. An SQL command console is granted for access to dictionary SYSDICT.

```
add module name is VIDMS_DEFAULT version is 1
    include user JOESIXP registered for all
    public access allowed for DISPLAY
    module source follows
```

Dictionary/SYSDICT

CV DBTable

CV DMCL

CV System

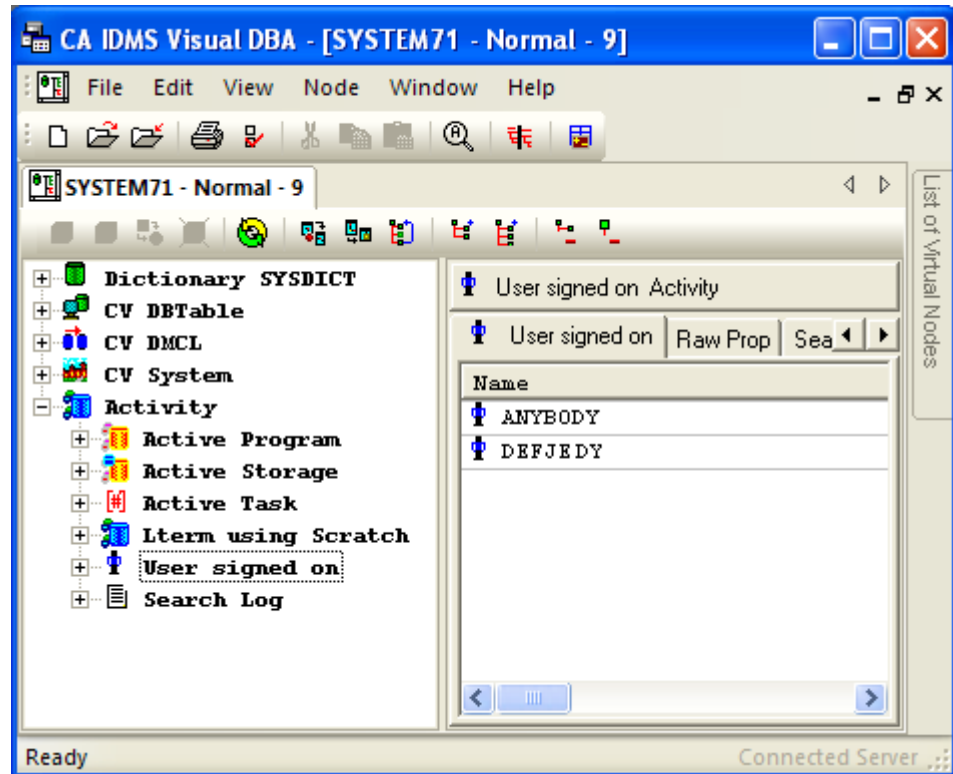
Activity

*

Cmd Console SQL/SYSDICT

msend.

After a successful connection of user ANYBODY and expansion of User signed on, the DOM tree and Detail Information Pane (DIP) looks like this:



CA IDMS Visual DBA Objects Security

Setting up appropriate CA IDMS Visual DBA user profiles is the easiest way to control access to CA IDMS dictionaries and run time data by CA IDMS Visual DBA users.

The remaining of this chapter gives information on the SQL objects defined for use by CA IDMS Visual DBA. Access to these SQL objects and their associated resources by other CA IDMS client applications must be controlled and optionally secured. Because CA IDMS Visual DBA is an SQL client application, its users also need to be granted privileges for all the accessed resources.

The names of the resources, which are of type TABL, are in the form schema-name.table-name. The objects in the following schema's are for exclusive use by CA IDMS Visual DBA:

- SYSVNTWK5 contains views on objects in the DDLDDL area of dictionary
- SYSVSYST5 contains views on objects in the DDLCAT area of dictionary
- SYSTSCHM5 is used for access to the Non-SQL schema IDMSNTWK (non Sysgen)
- SYSTSYST5 is used for access to the Non-SQL schema IDMSNTWK (Sysgen)
- IDMSSECS5 is used for access to the Non-SQL schema IDMSSECS
- IDMSSECU5 is used for access to Non-SQL schema IDMSSECU

The following table procedures and functions in schema SYSCA are also for use by CA IDMS Visual DBA:

- SYSCA.VDBA_DICTIONARY5 definition only
- SYSCA.VDBA_VERSION5 definition only
- SYSCA.VDBAIDD5 for executing BCF/OCF, IDD, Schema, SSC, and Sysgen compilers
- SYSCA.VDBADCT5 for executing DCMT commands
- SYSCA.VDBADCU5 for executing DCUF commands
- SYSCA.VDBALOG5 for viewing the LOG database
- SYSCA.VDBAQNC5 for viewing columns of Non-SQL Schema's
- SYSCA.VDBAGET5 generic function
- SYSCA.VDBA_CONV2CHAR5 generic function

Grant Access to Objects

If access is granted to objects for non-SQL schema's, you also need to grant Use privilege for the corresponding non-SQL schema.

Note: For more information, see the SQL definitions in the CA IDMS Visual DBA members VDB5R16X, VDB5R17X or VDB5V18X in your CA IDMS source library or in the CA IDMS Visual DBA IdmsR16x, IdmsR17x or IdmsV18x folders.

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.

Privileges to Update Object Instances and Use Command Console

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. Normal CA IDMS security applies for running these compilers and for executing DCMT commands.

CA IDMS Visual DBA runs the compilers and DCMT with the help of table procedures SYSCA.VDBAIDD5, SYSCA.VDBADCT5 and SYSCA.VDBADCU5. To perform any updates with CA IDMS Visual DBA or to execute commands and scripts in a Command Console, you need the SELECT and the INSERT privilege on resource names SYSCA.VDBAIDD5, SYSCA.VDBADCT5, and SYSCA.VDBADCU5.

Similarly, to globally prohibit updates of any of the CA IDMS Visual DBA objects or execution of commands and scripts in a Command Console, you can revoke the SELECT and INSERT privileges on resource name SYSCA.VDBAIDD5, SYSCA.VDBADCT5, and SYSCA.VDBADCU5 for a user.

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.

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.

Chapter 3: Start Your CA IDMS Visual DBA Session

You can begin your work session by starting CA IDMS Visual DBA and connecting to a CA IDMS system.

To start CA IDMS Visual DBA

1. Click Start, Programs.
2. Click CA menu.
3. Click the CA IDMS Visual DBA submenu.
4. Select CA IDMS Visual DBA.

This section contains the following topics:

[The Application Window](#) (see page 49)

[The Workspace Toolbar](#) (see page 50)

[The Nodes Window Toolbar](#) (see page 52)

[Connect to a Datasource/Node](#) (see page 53)

[Add a New Datasource/Node](#) (see page 53)

[Open a Database Object Manager Window](#) (see page 56)

[What's Next](#) (see page 57)

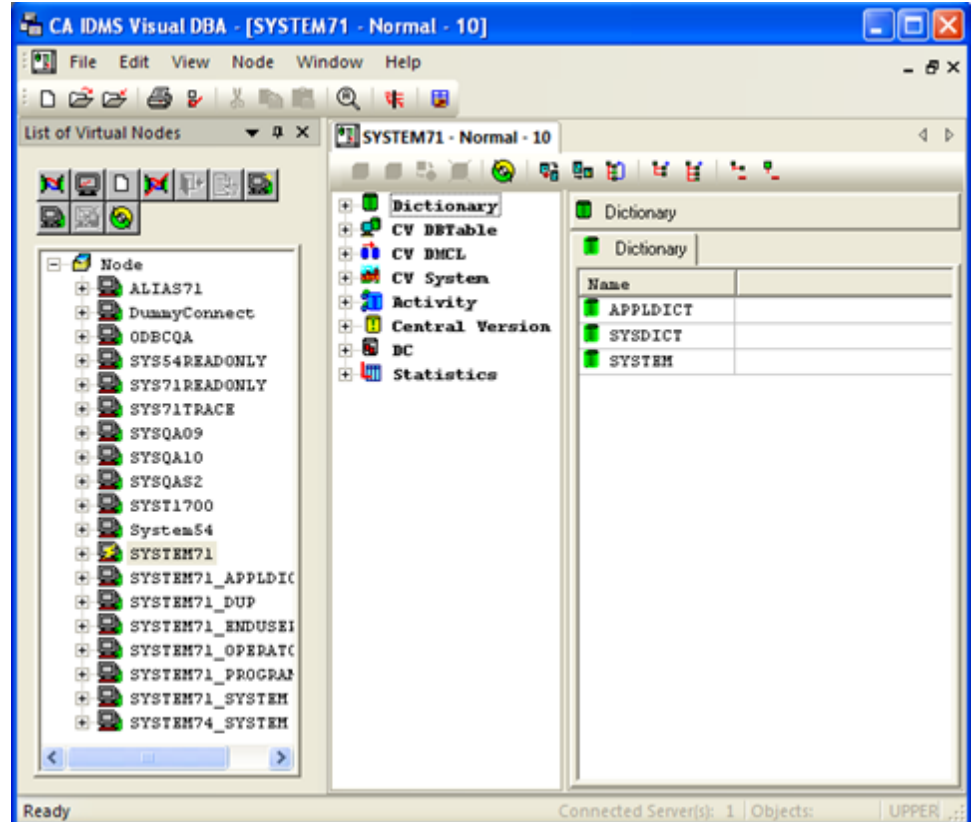
The Application Window

CA IDMS Visual DBA displays 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 subwindows:

- The *Nodes window*, by default located on the left is used to display and manage the nodes or ODBC data sources. There is only one Nodes window per CA IDMS Visual DBA instance. It is a movable window. It can be made floating, it can be docked and auto-hiding. Because it is only used for connection management, it usually is most efficient to dock it to the right side and set it to auto-hiding. This setup gives the most workspace to the DOM window(s). The position and visibility attributes of the Nodes window are preserved after exiting CA IDMS Visual DBA.
- The right subwindow 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.

Note: If instead of the Dom tree in the DOM window a dialog box appears with the message "Failed to create empty document", the IDMS CV has Enhanced Object Security enabled and the user connected to the CA IDMS system has not been assigned a profile or the profile does not contain any valid root nodes. Review and correct the configuration of the mainframe component of CA IDMS Visual DBA.











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 the following brief:




The Workspace Toolbar

The Workspace toolbar enables you to perform the following functions for which a description is also available as a tooltip:













Button	Description
	<p>Creates a new workspace environment.</p> <p>If the current environment has never been saved or has been changed during the session, you are prompted to save it. When you end your session, the current workspace configuration is closed—closing all windows, terminating connections, and setting default system settings.</p>
	<p>Opens a workspace that was previously saved.</p> <p>If a password was previously specified for the workspace, a dialog displays prompting you to provide the password.</p> <p>When the operation is completed, the workspace environment displays in the CA IDMS Visual DBA window.</p>
	<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
	<p>Prints the information in the active window.</p>
	<p>Establishes your preferences. Modifies system-wide parameters, such as fonts, session preferences, printer setup options, refresh settings, OK action, and command console options.</p>
	<p>Removes text from its current location in the Command console, and copies it to the Clipboard.</p>
	<p>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.</p>
	<p>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.</p>
	<p>Finds and jumps to an object or object category in the currently selected Database Object Manager window.</p>
	<p>Arranges all open windows so that they are side-by-side with no overlap.</p>

Button	Description
	Lets you browse the error history log.

The Nodes Window Toolbar

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 and closes all opened windows on the selected node.
	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.

Connect to a Datasource/Node

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

To connect to a Datasource or Node


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.

Add a New Datasource/Node

You can use the following steps to add a new datasource or node.

To add a new Datasource or node

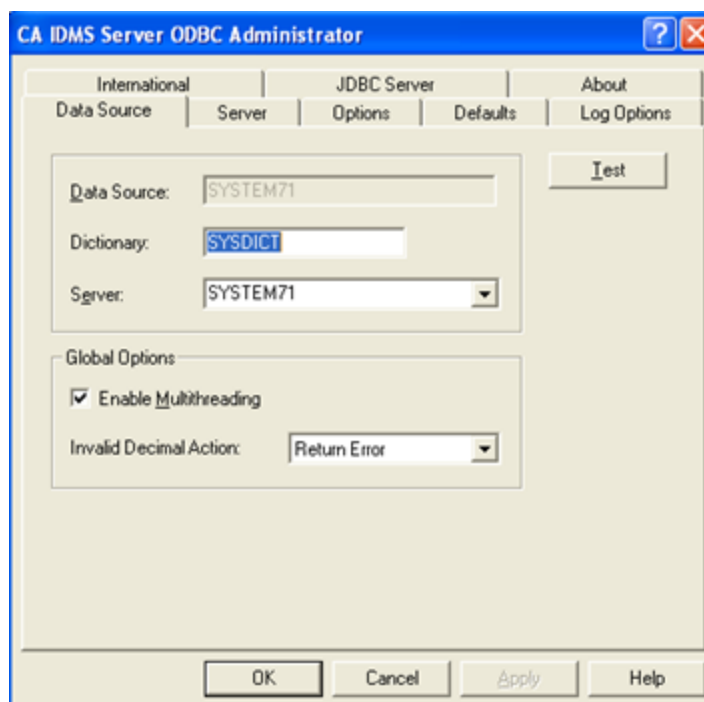
1. Select the Add function from the Node window or click the Add button in the Nodes window toolbar , or right-click in the nodes window to use the Node menu.

A dialog displays prompting you to determine if the new data source must be system or user. A system data source is available to all users of the PC, a user data source only to the defining user. Defining a system data source requires more operating system privileges than a user data source.

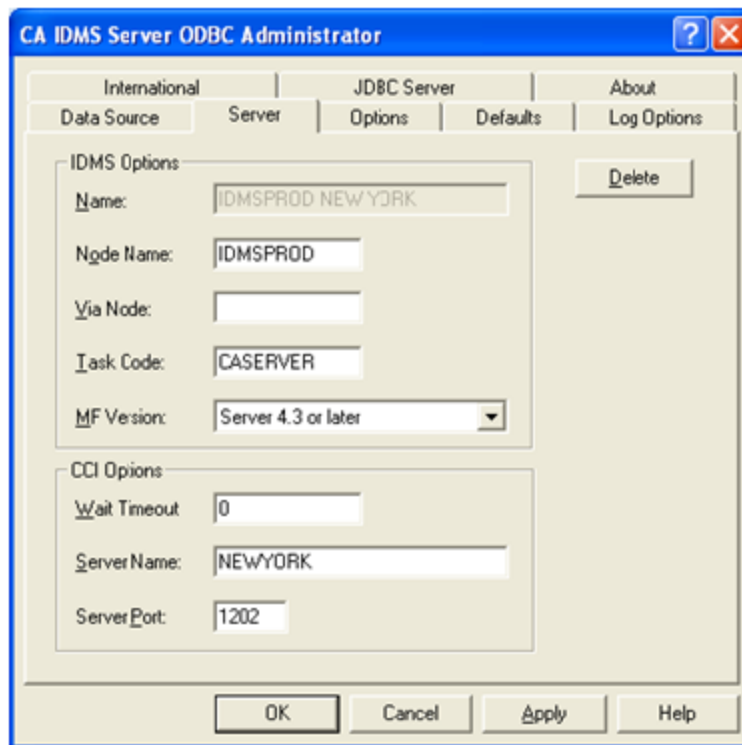
2. Respond to the prompt.
The CA IDMS ODBC Administrator is invoked.
3. In the Data Source field, enter a name for the new data source.

4. In the Dictionary field, enter the name of the dictionary in which you installed the mainframe component of CA IDMS Visual DBA.
5. From the Server drop-down combo box, select a CA IDMS system.

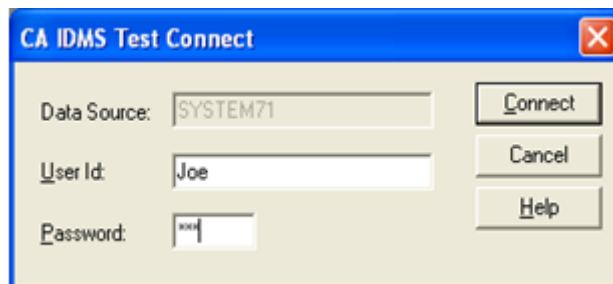
The following example shows SYSDICT as the Dictionary and SYSTEM71 as the Server name:



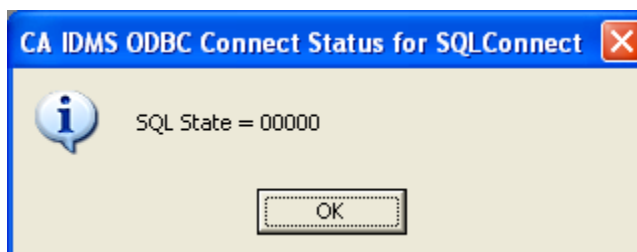
If the Server drop-down list box is empty or no usable server name is available a new Server must be defined. The name entered in the Server combo box is only an abstract name that corresponds to an IDMS CV with all the communication parameters required to connect to that IDMS CV. In the example we entered IDMSPROD NEW YORK in the Server combo box and then clicked the Server tab. This will show a new dialog that allows for specifying all the communication parameters of the abstract server IDMSPROD NEW YORK. Enter in the Node Name field the IDMS node name of the IDMS CV, this is the name specified in the SYSTEM ID clause of the IDMS CV system definition, that is IDMSPROD. The default values for Via Node, Task Code and MF Version are in general acceptable. In The CCI Options group box, the CCI Server Name can be entered. This is the TCPIP hostname where the IDMS CV is running, that is, NEWYORK. The CCI server port is usually 1202.



When all parameters are entered, click Apply, select the Data Source tab and click the Test button to execute an ODBC test connection. This will show the following dialog. Enter user id and password to signon to the IDMS CV and click the Connect button.



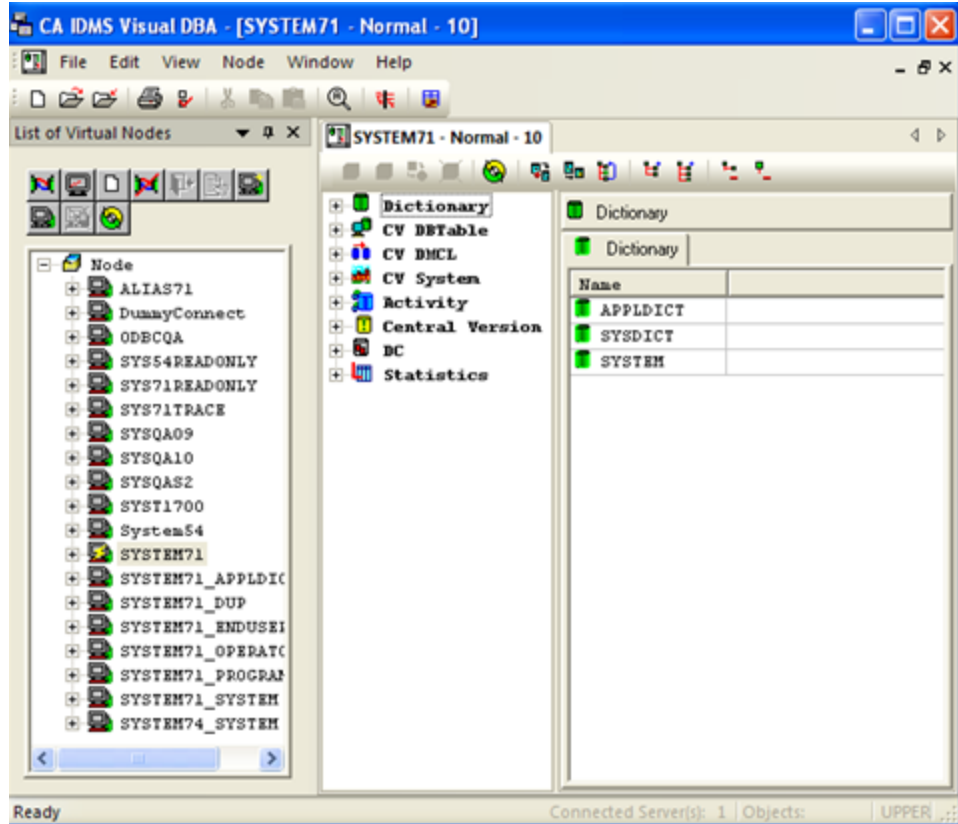
A successful connection is indicated as follows:



Note: For more information on defining ODBC data sources see the *CA IDMS Server User Guide*, which is also available in the CA IDMS Server product folder.

Open a Database Object Manager Window

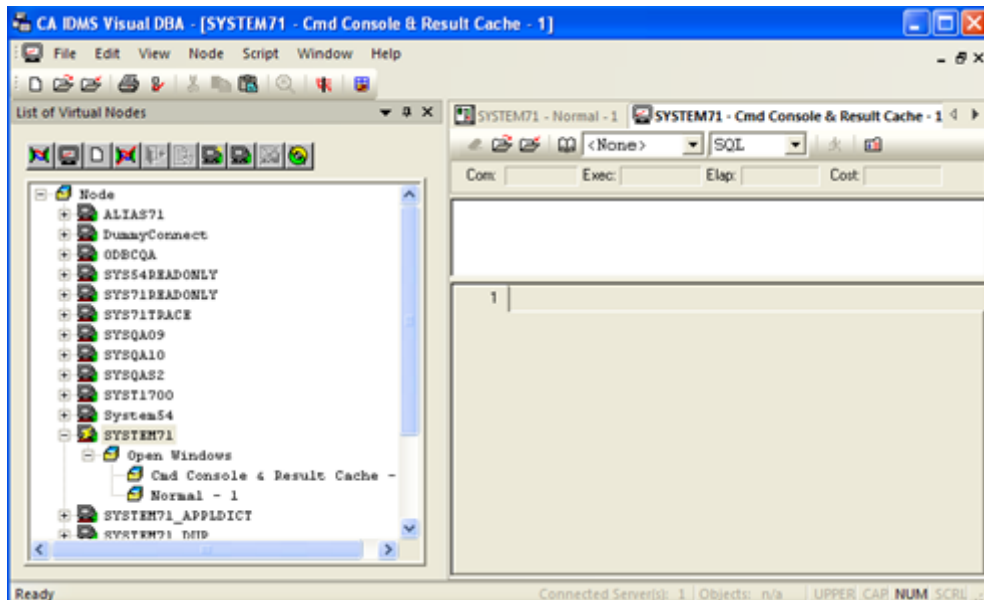
After you select a data source or node and connect to it, a Database Object Manager window is displayed, which looks similar to the following sample window.



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 space of the existing one. Use the functions in the Window menu to manage the windows. You can use the tabs below the main toolbar to quickly navigate from one window to the other.

In the following example, the same data source is used multiple times. The expansion of the connected node in the Node window shows that it is used in both Normal and Command Console modes.



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, you can perform a variety of database administration tasks. The next chapter introduces you to the features of the Database Object Manager window.

Chapter 4: A Tour of the Database Object Manager Window

This section contains the following topics:

[The DOM Window](#) (see page 60)

[Expand and Collapse the Object Tree](#) (see page 63)

[Change the Tree Structure](#) (see page 65)

[Manipulate Tree Objects](#) (see page 70)

[Find Objects](#) (see page 71)

[Multiple Objects Selection](#) (see page 72)

[Object Syntax Viewing, Execution, and Logging](#) (see page 72)

[Refresh the Tree](#) (see page 73)

[Connect to Multiple CA IDMS Systems](#) (see page 74)

[Display Object Attributes in the Detail Information Pane](#) (see page 75)

[Set Display Options for Your Session](#) (see page 96)

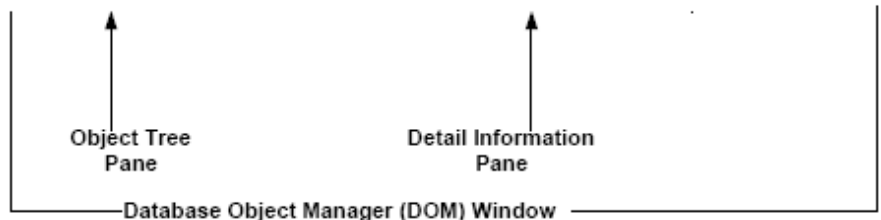
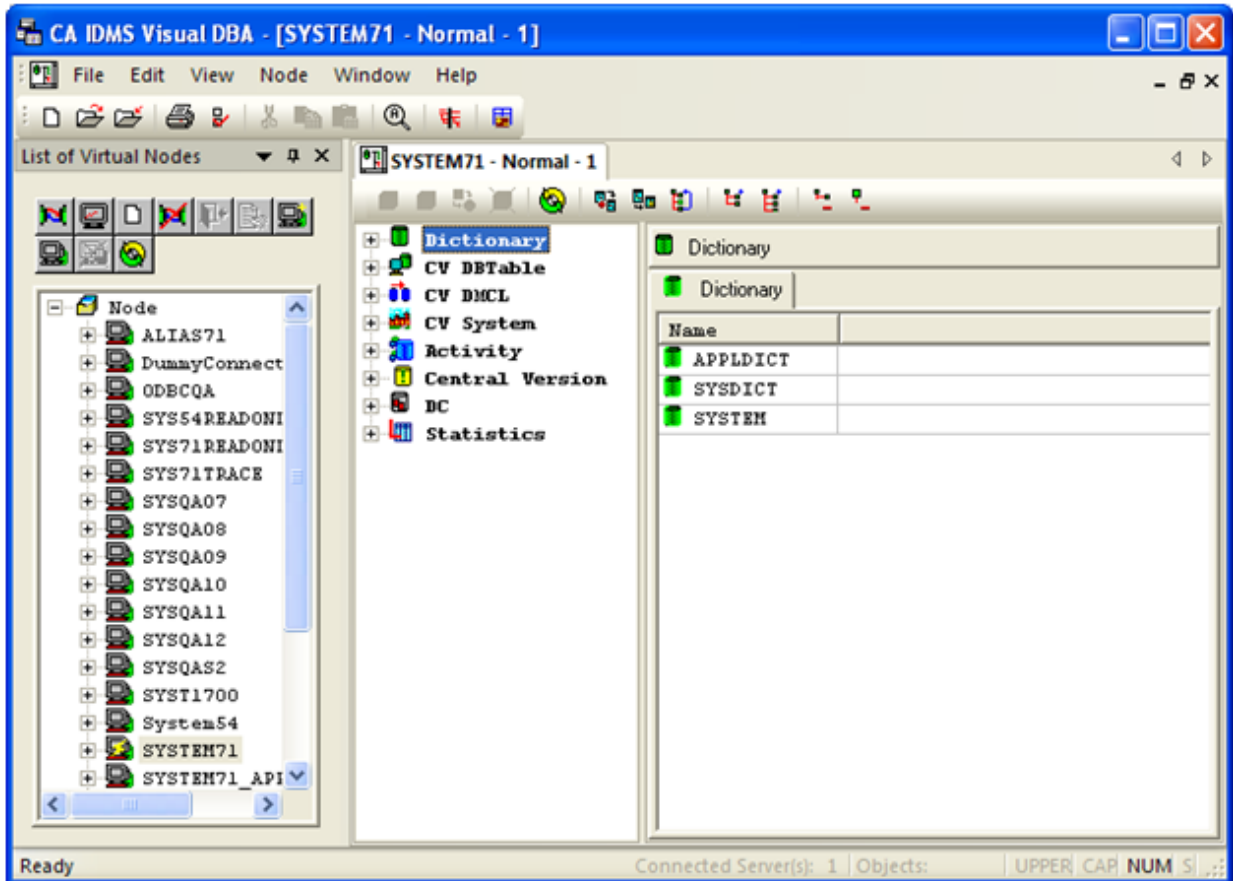
[Create, Open, and Save Environments](#) (see page 97)

[Try It Out Yourself](#) (see page 97)

The DOM Window

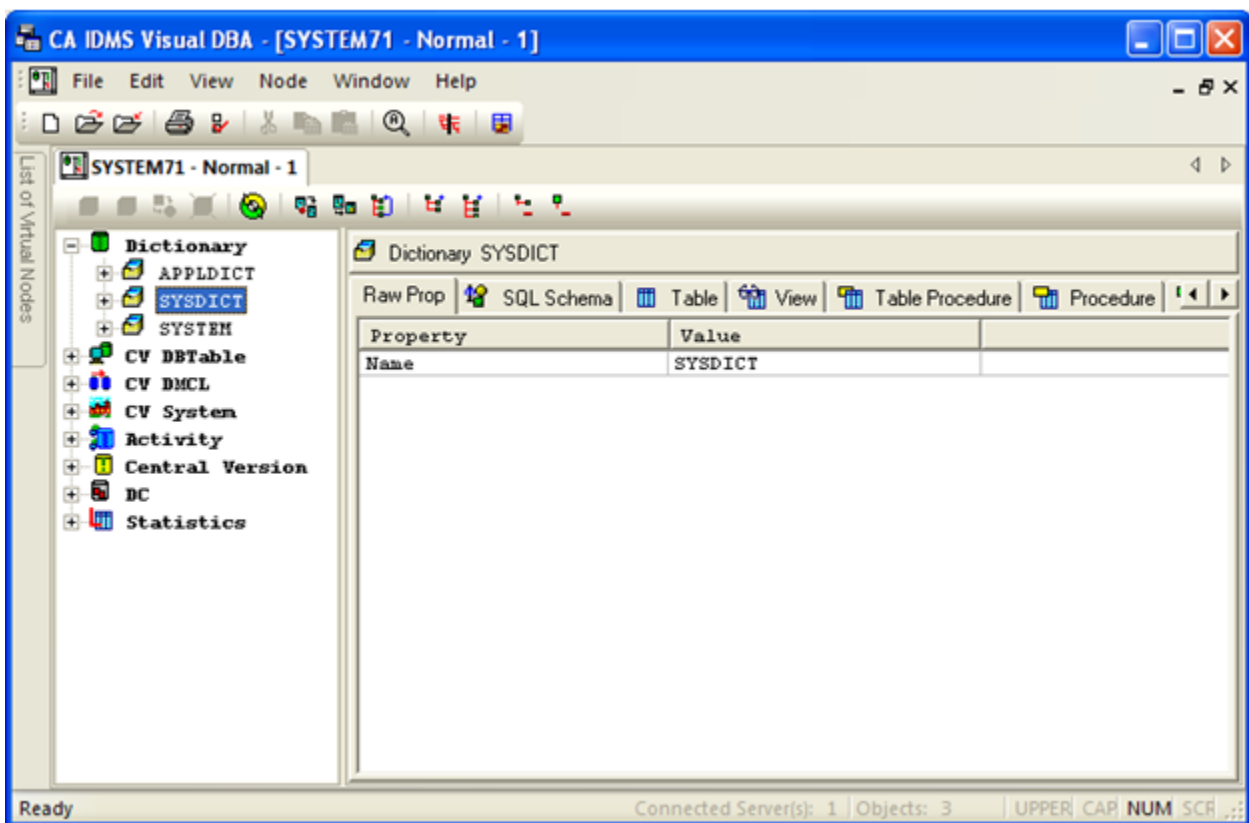
Inside a CA IDMS Visual DBA, the Database Object Manager provides a convenient and organized way to view and manipulate the database information that is currently stored on a particular server.

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 according to the profile of the current user. In the above example, the Dictionary object type is the first root object in the Object Tree pane. Because it is the first object in the tree, it is selected and highlighted and the level-one instances of the Dictionary object (in this example APPLDICT, SYSDICT, and 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 objects, and assign privileges to them. Additionally, you can open and simultaneously work with multiple databases and/or servers in multiple windows.











Note: If you are connected to a CA IDMS server that is older than the most recent release of the product, you may not be able to manipulate some objects in the DOM. When expanded, these objects display the message <No *OBJECT*>, where *OBJECT* is the name of the object you are trying to view. Similarly, these objects have null values when viewed in the Raw Property table or the Property dialog.



Use the DOM Window Toolbar

The DOM window toolbar enables you to perform the various functions such as creating and dropping an object, changing the attributes of an existing object, and so on.

To use the DOM window toolbar



Icon	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 subbranches associated with the current branch. Note: This command fully expands only those subbranches that are not associated with a cross-referenced object (so that the tree does not expand infinitely).
	Expands and displays all the branches and subbranches of all the root objects in the current DOM window. Note: This command fully expands only those subbranches that are not associated with a cross-referenced object (so that the tree does not expand infinitely).

Icon	Description
	Collapses the subbranches of the current branch. Note: When this command is used, CA IDMS Visual DBA remembers the expansion state of the lower-level subbranches (that is, the subbranches of the subbranches, and so on). When the current branch is subsequently expanded, the previous expansion state is duplicated.
	Collapses <i>all</i> subbranches associated with every root object category branch.

Expand and Collapse 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, double-click 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 displays 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.



Expand 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.

Expand One Level at a Time

Button	Description
	<p>Expands a branch at one level when you click its expansion button ().</p> <p>This displays the subbranches that exist one level down within the selected branch.</p> <p>Note: Only expandable branches have this button.</p> <p>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.</p>

Expand Multiple Levels

Button	Description
	<p>Expands all the subbranches under a single branch when you click the Expand Branch toolbar button. You can also use the toolbar, shortcut keys, and the View menu. Simply select the branch and click the Expand Branch toolbar button, press the Multiply (*) key, or use the Expand Branch command from the View menu.</p>
	<p>Expands all subbranches for <i>every</i> branch, when you click the Expand All Branches toolbar button. You can select the Expand All Branches toolbar button, press Ctrl + *, or choose the Expand All Branches command from the View menu.</p>





Infinite Drill

One of the very 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 see that nested within most new subbranches are lower-level subbranches 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* subbranch into a root branch from which you can drill down.

Collapse the Tree

The following table discusses various ways in which you can collapse the tree branches in the Database Object Manager window.


Button	Description
	Collapses a single subbranch, when you 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.
	Collapses all the subbranches under a single branch, when you use the toolbar, shortcut keys, and the View menu. Select the branch and click the Collapse Branch toolbar button, press the Minus (-) key, or use the Collapse Branch command from the View menu.
	Collapses all subbranches for <i>every</i> branch, when you click the Collapse All Branches toolbar button, press Ctrl + / , or choose the Collapse All Branches command from the View menu.

Change the Tree Structure

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

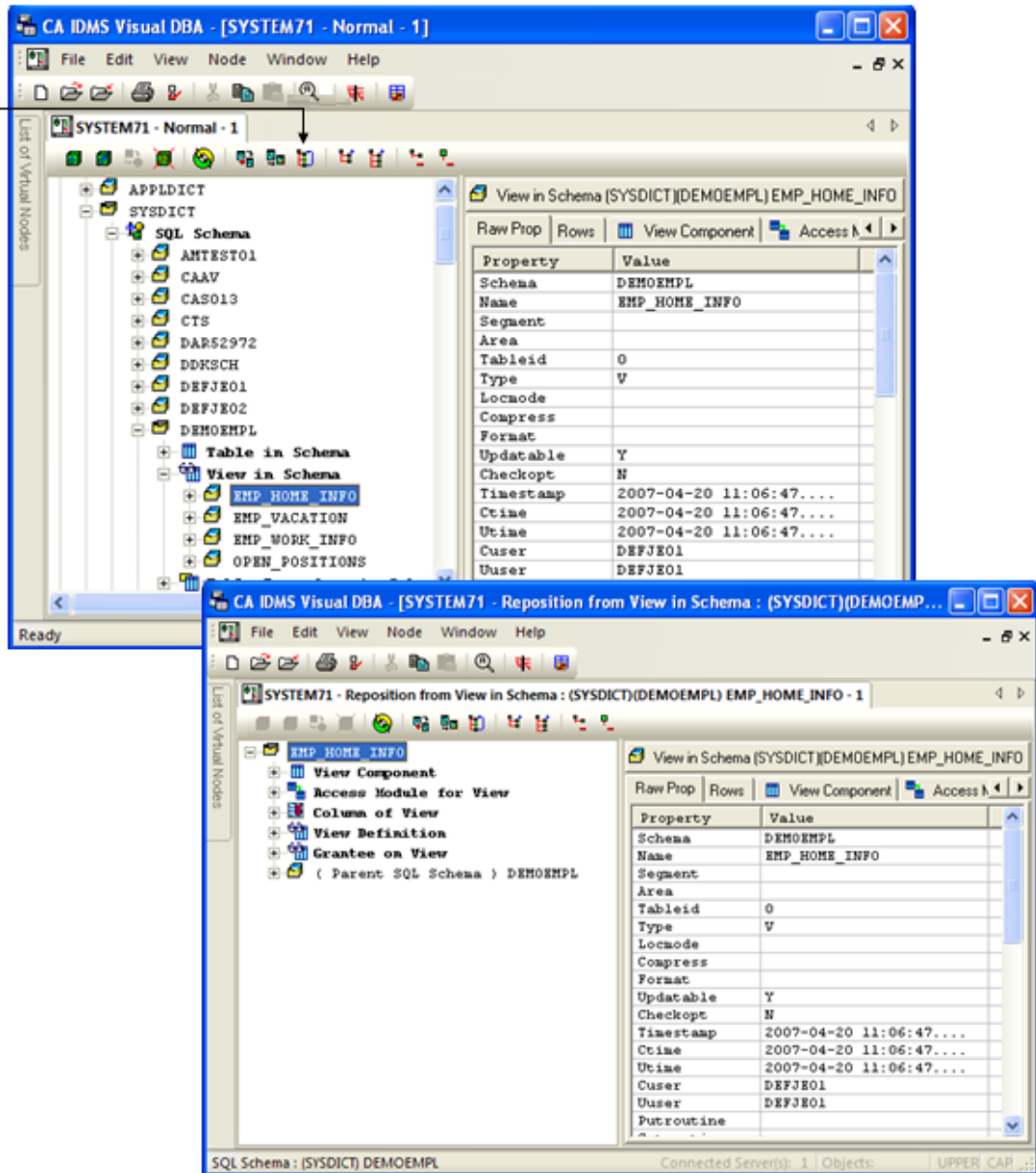
Restart from Position

The information you need often begins on a level far beneath the root object category branch. The layers of subbranches 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.

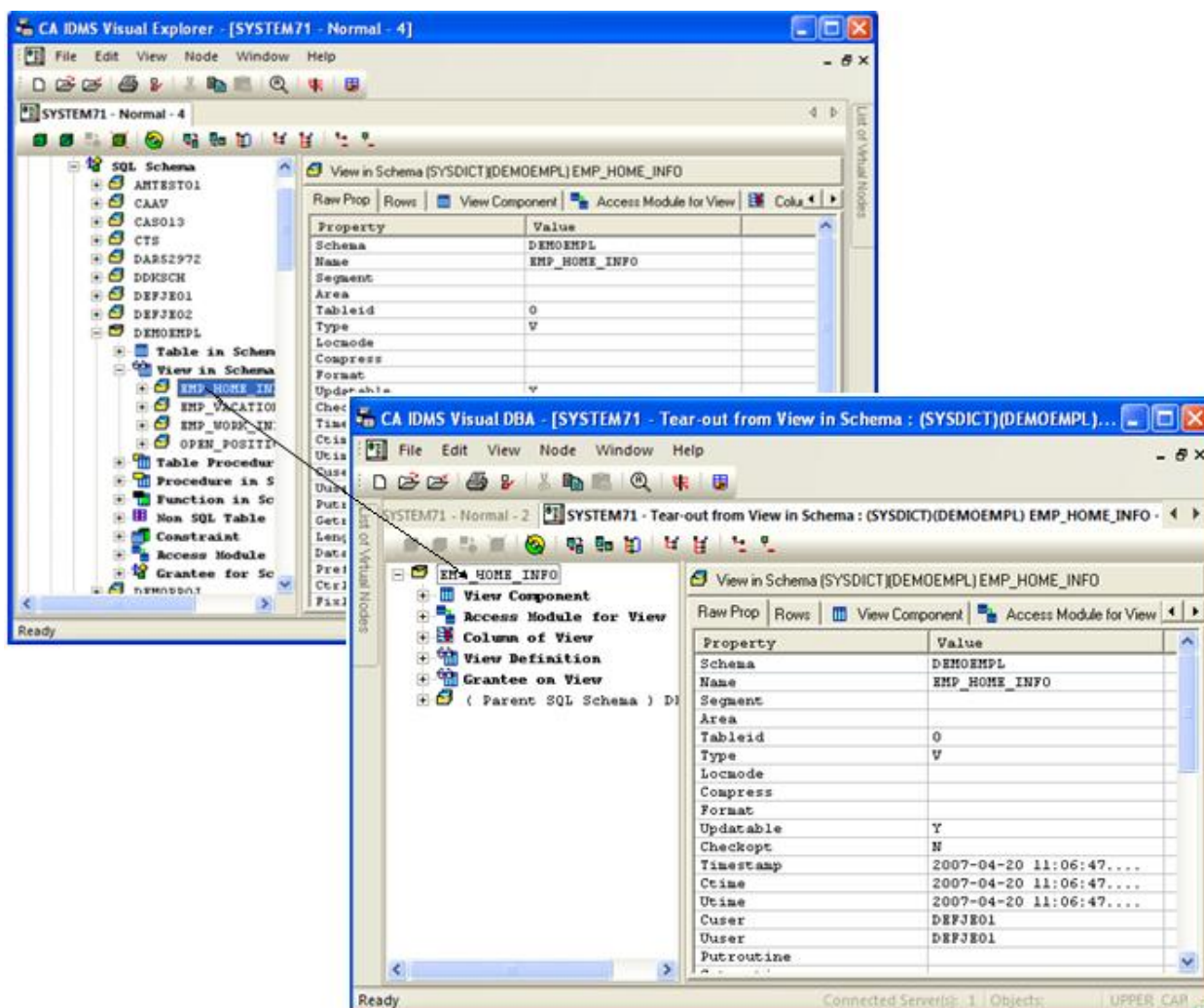
Tear-Out Window

You can use tear-out windows to customize your view of the hierarchical object tree display.

If you want to view a branch in a *new* window, you may use the Tear-out feature. By

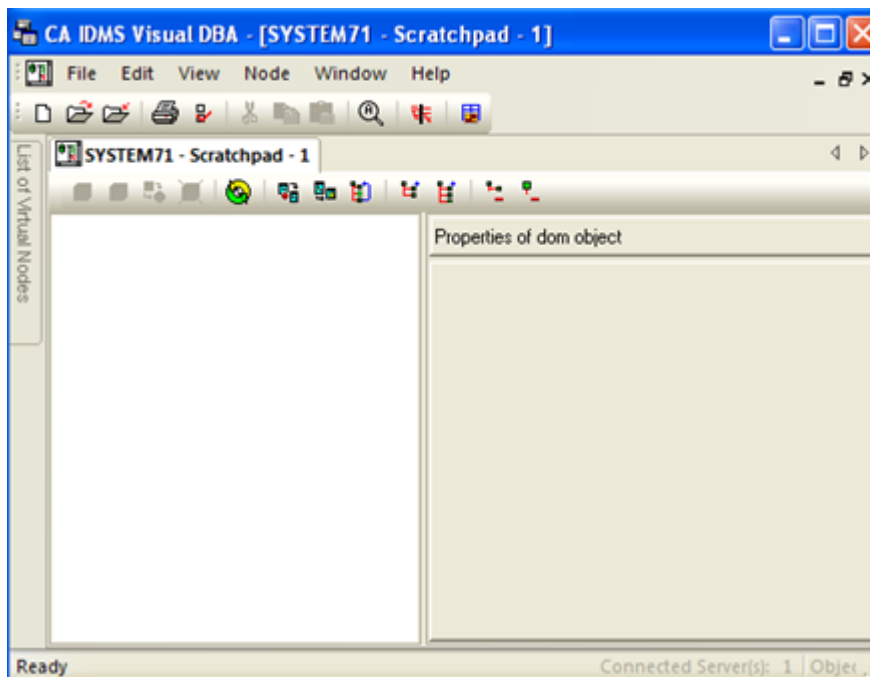
selecting a branch and clicking the Tear-Out toolbar button  (or choosing the Tear-Out command from the Window menu), you can create a new window, while leaving the original window in tact. The branch you select becomes the root branch in the new window and may be expanded to reveal all existing subbranches.

As an example, you can 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:

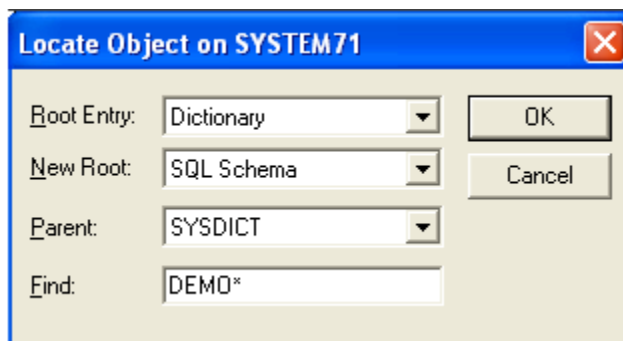


Scratchpad

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.

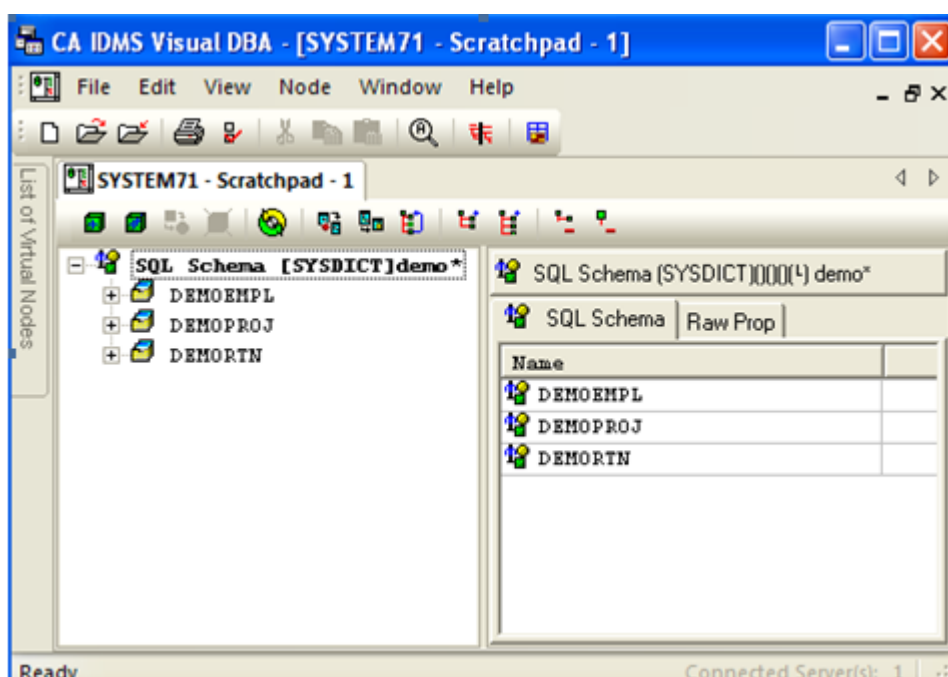


CA IDMS Visual DBA allows you to easily find any object in the tree associated with your profile on the current server and display it as the root branch in the Scratchpad window. To do this, open a Scratchpad window and then choose the Edit Locate menu command, which invokes the Locate Object dialog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



In the Root Entry drop-down list box, choose from the root entries defined in your profile, the root entry from which you wish to select an object type as a new root (Dictionary in the above example). In the New Root drop-down list box select the object type that becomes a new root (SQL Schema in the example). If the newly defined root object requires a parent object, select the parent from the Parent drop-down list box (SYSDICT in the example). Finally you can filter the object instances using the Find edit control (in the example only SQL Schema's whose name starts with DEMO makes up the new root entry). Leave only an asterisk (*) if you do not want to filter the object instances.

Expanding this branch reveals the branches for SQL schemas in dictionary SYSDICT whose name starts with DEMO:



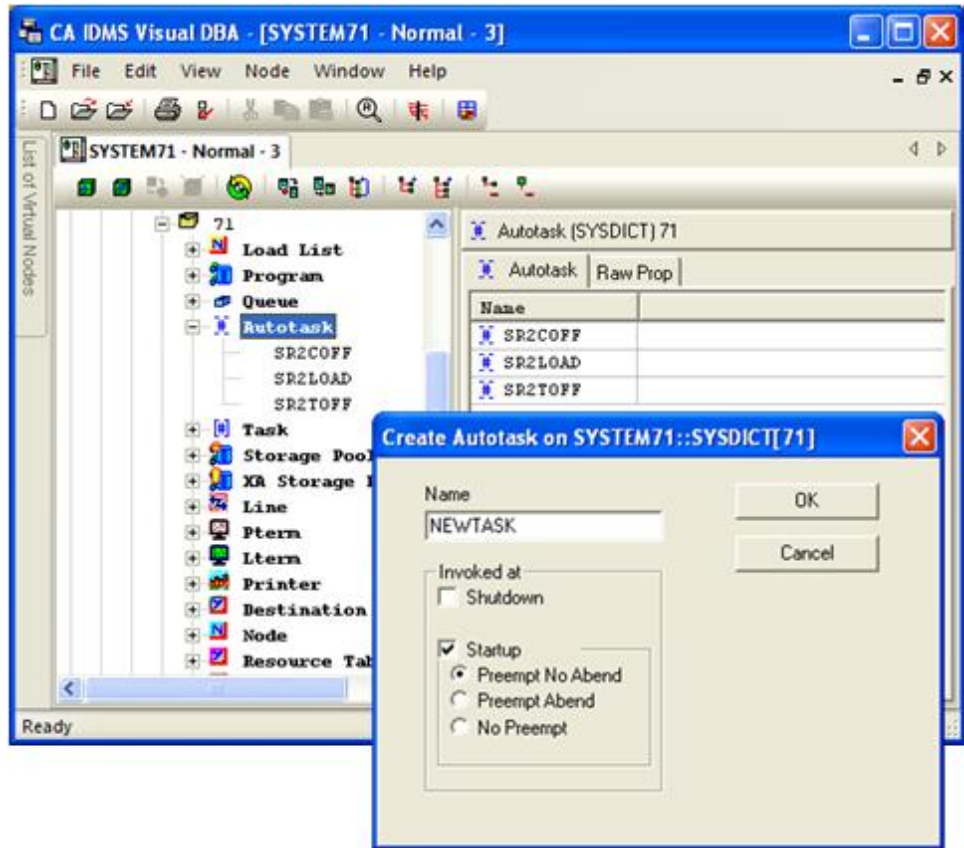
The Window Restart From Position, Tear Out, and Scratchpad menu commands all enable you to construct your own Database Object Manager windows that focus precisely on the information *you* need within the scope allowed by your profile.

Manipulate 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.

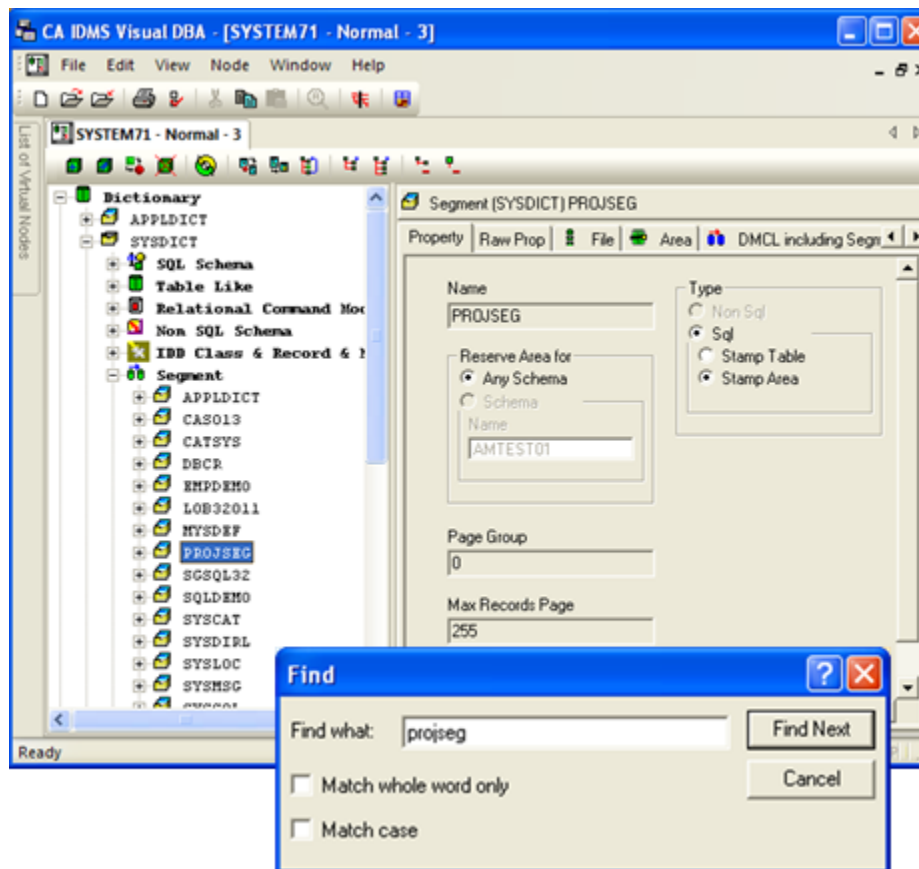
To do these things, select the object and right-click. A pop-up menu appears which displays the actions allowed for the object. Alternatively, you can select the object and use the commands on the Edit menu or use the *DOM toolbar*.

In the following figure, the Autotask system object is selected. The pop-up 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.



Find 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). A dialog displays prompting you for the search string and other search criteria. When the segment is found, it is highlighted in the Object Tree pane and associated information displays in the Detail Information Pane.



Multiple Objects Selection

You can select multiple objects in the same branch of a tree or in the child tab of the Detail Information Pane.

To select multiple objects

- 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 subbranch or from the Detail Information Pane to any other subbranch 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 IDMS tasks, especially for test and production database maintenance.

Object Syntax Viewing, Execution, and Logging


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.

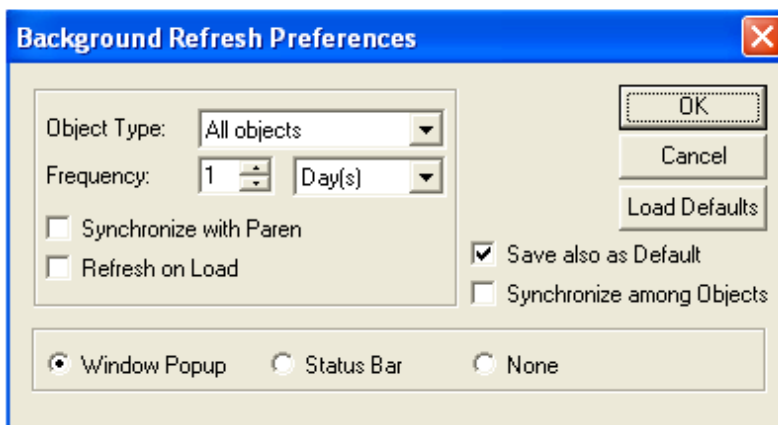
By default, CA IDMS Visual DBA executes the object syntax and displays the syntax only if an error occurs. If you select View Syntax only, it displays, but does not execute the syntax. If you select View Syntax and Exec Syntax, it displays the syntax and asks if you want to execute it.

If you select Log Syntax, a log file is created with an extension that designates the compiler and all object syntax for that compiler is logged 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.

Refresh 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 refreshes the data at a specified frequency, using the parameters defined in the Background Refresh Preferences dialog invoked by the File Preferences command.

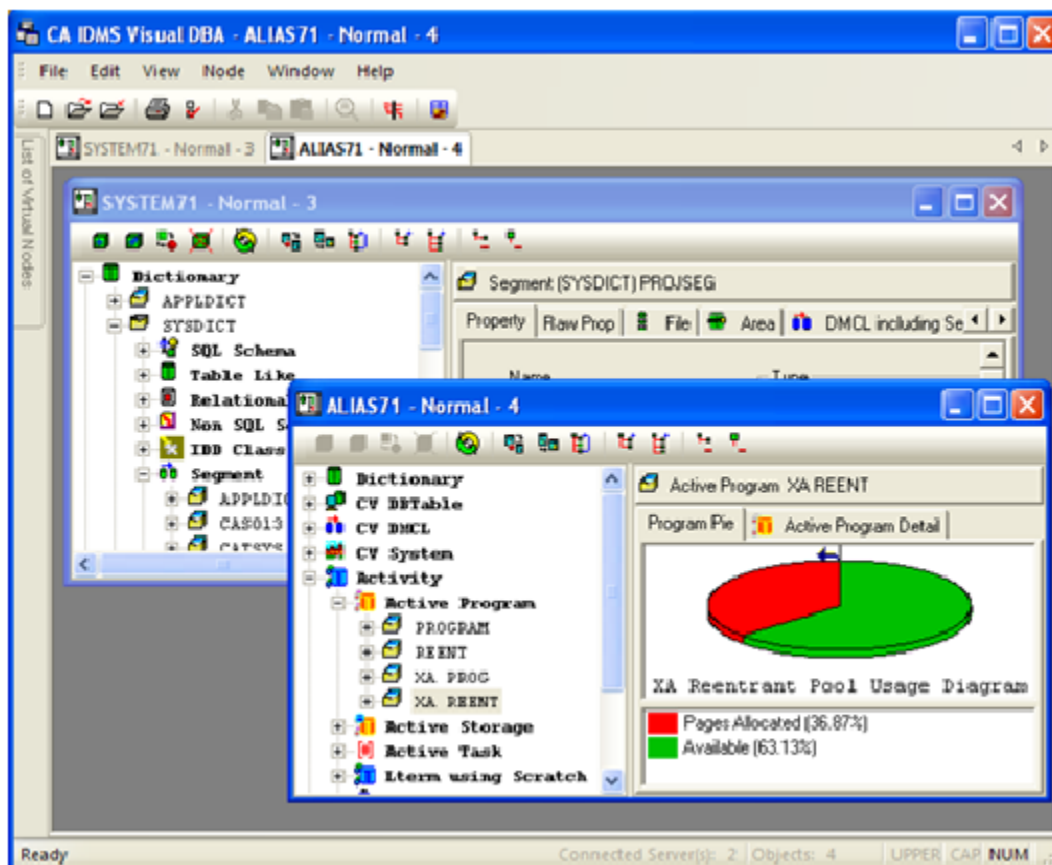


You can set the Background Refresh Preferences so that different object types are refreshed at different frequencies. You can also save your refresh setting as default settings to use with current and new workspace environment configurations.

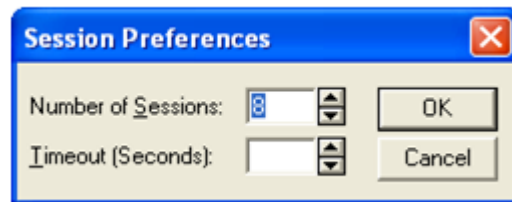
Connect to Multiple CA IDMS Systems

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

Another Database Object Manager window opens, which displays the object tree data for the CA IDMS system to which you just connected. 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.



Display 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. Some tabs allow you to execute a CA IDMS utility on a selected database object. The different categories of information are listed as follows:

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

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.

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.

Columns

Displays a table where each rows represent a column of an SQL table and contains the name, the datatype, the nullability, and the default specification of the column.

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.

Module Text

Displays a dialog with Update and Reset buttons, an input pane for editing a dictionary module, and a result pane to hold the status of the last Update execution.

Parameters

Displays a table where each row represents a parameter of an SQL table procedure, procedure or function and contains the name, the type, the size, the default specification, and a test value for the parameter.

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 the pie slice representing actual usage, CA IDMS Visual DBA displays detailed statistics.

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. You need to double-click the column to reorder the table.

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. CA IDMS 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, CA IDMS 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.

Search Log

Allows you to view messages in the CA IDMS log that relate to selected CV objects. You can search within a specific time period and specify a filter so that only the messages you want are retrieved.

SQL Routine Body

Displays a dialog with Update and Reset buttons, an input pane for editing the body of an SQL routine, and a result pane to hold the status of the last Update execution.

Test

Displays the result of invoking an SQL-invoked function or procedure.

Tune Index

Instructs CA IDMS to walk an index and tune the index using the parameters specified in the dialog. One of the optimizations is the adoption of orphaned indexed records. By eliminating orphans, runtime database performance is improved when traversing from an indexed record to its associated index entry.

View Index

Allows you to view the structure of system-owned indexes and indexed sets. You can query the index based on a number of criteria, such as the segment, output, level, and so on.

View Page

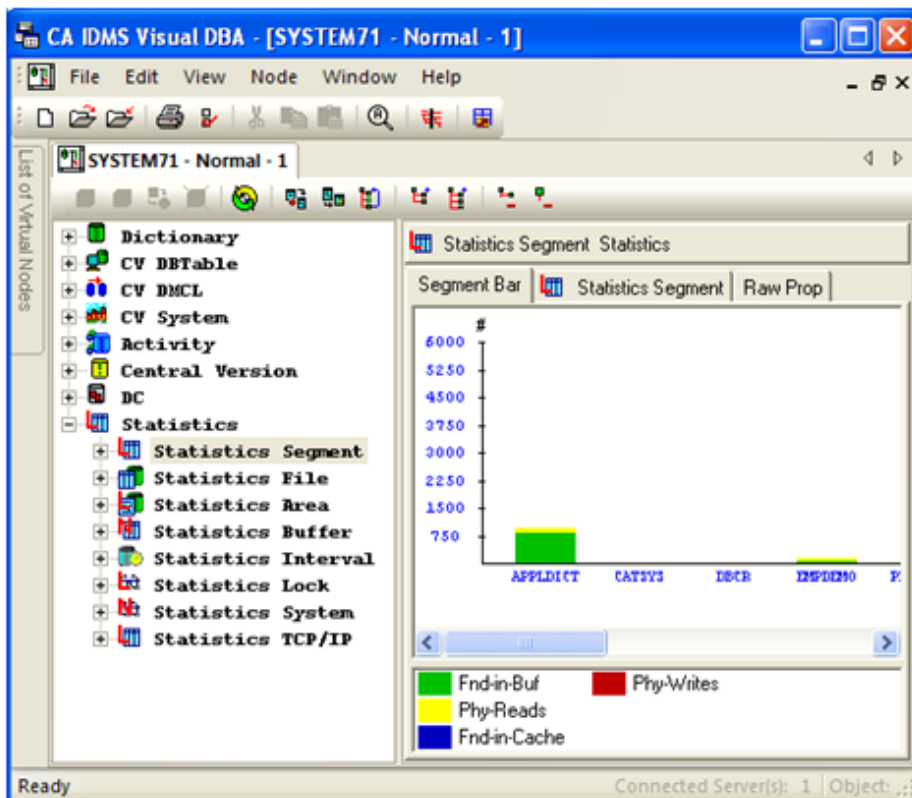
Allows you to view the contents of database pages in decimal format, hexadecimal format, or both. You can request information for a specific CALC key, subarea, or page range.

Tab Examples

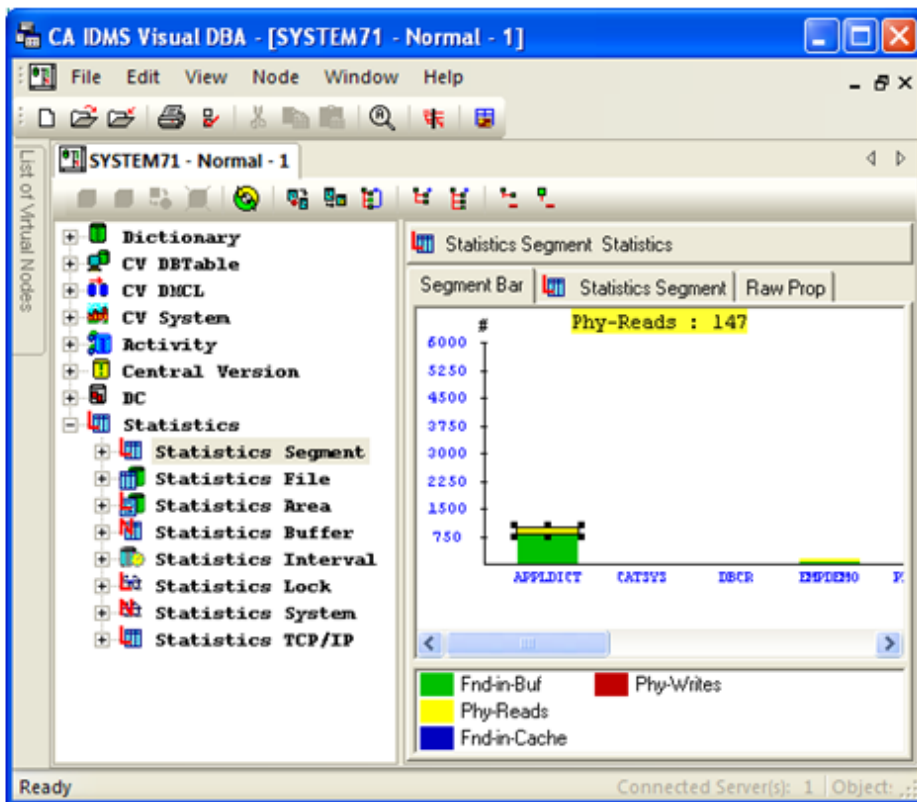
The following examples illustrate different types of tabs.

Bar

In the following example, statistical information is displayed in bar chart format for Statistics Segment. Note that the information is color coded.

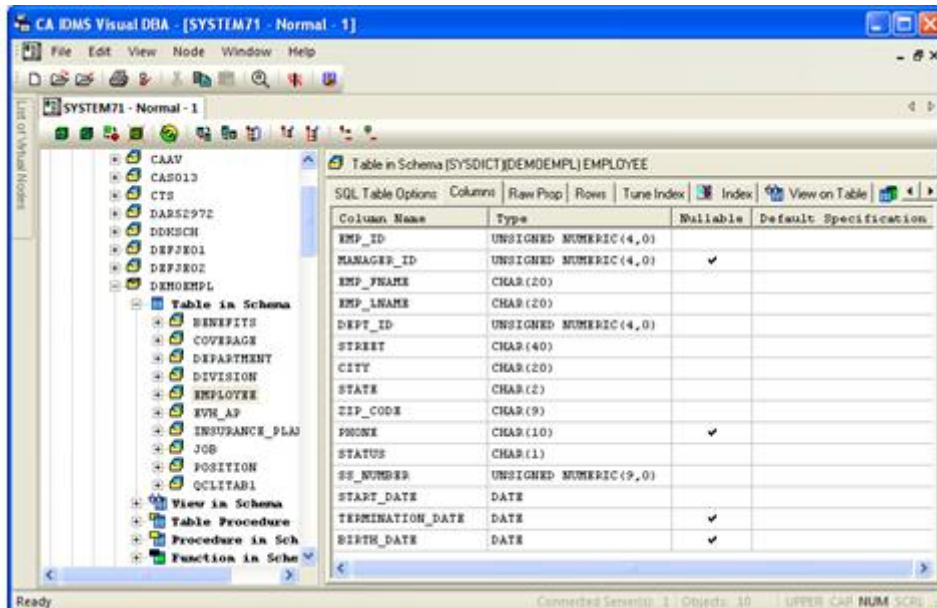


When you click a portion of the bar, the actual statistics for that segment display. In the following example, we click the yellow portion of the bar for APPLDICT. The actual number of physical reads (Phy-Reads) displays at the top of the pane. When you select any of the names shown along the X-axis,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



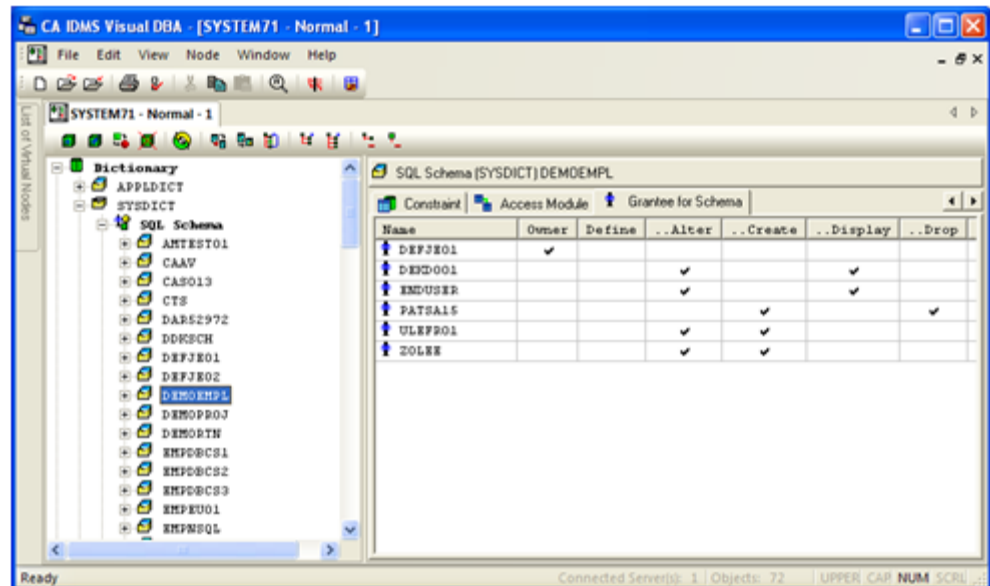
Columns

In the following example, the columns for table DEMOEMPL.EMPLOYEE are shown in the Columns tab.



Grantee

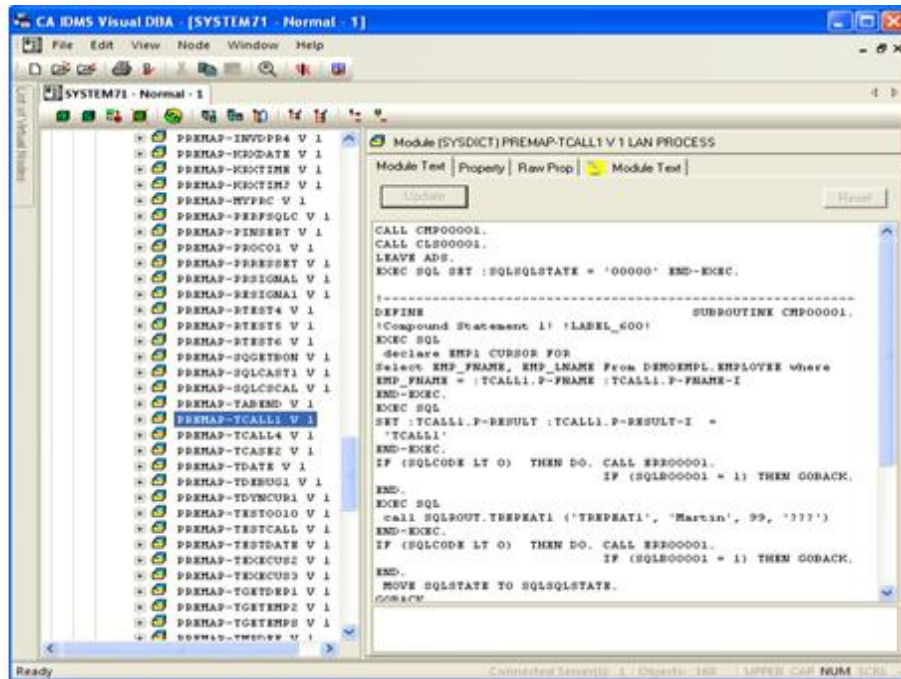
In the following example, grantee information is displayed for the Table in Schema DEMOEMPL. Note that the table includes a column for each privilege type and a row for each User ID.



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

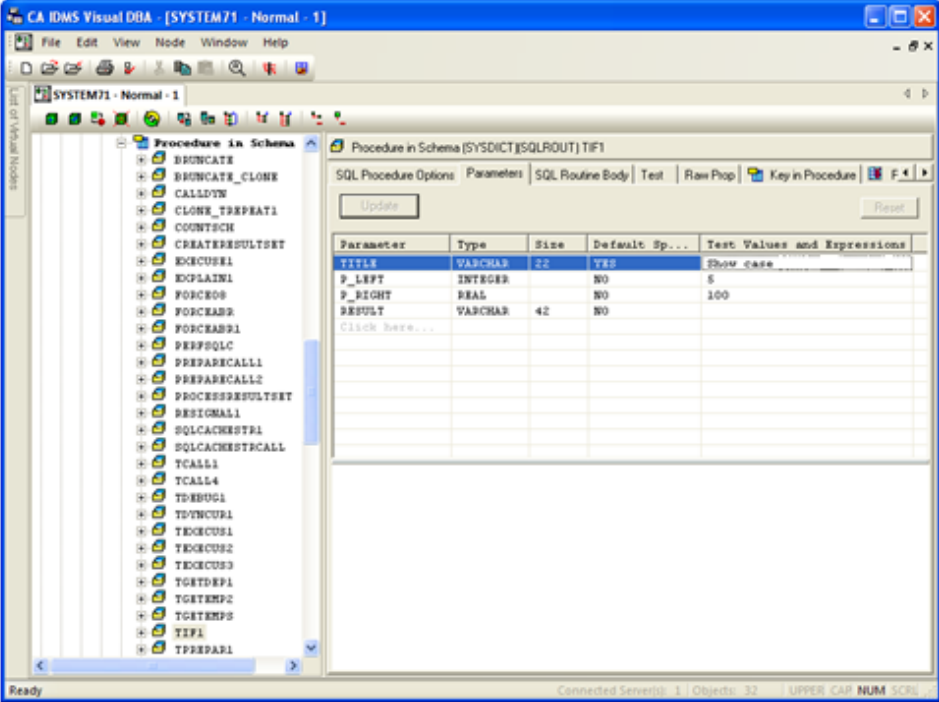
Module Text

In the following example, the content of dictionary module SYSDICT.PREMAP-TIF1 Version 1 is shown in the Module Text tab.



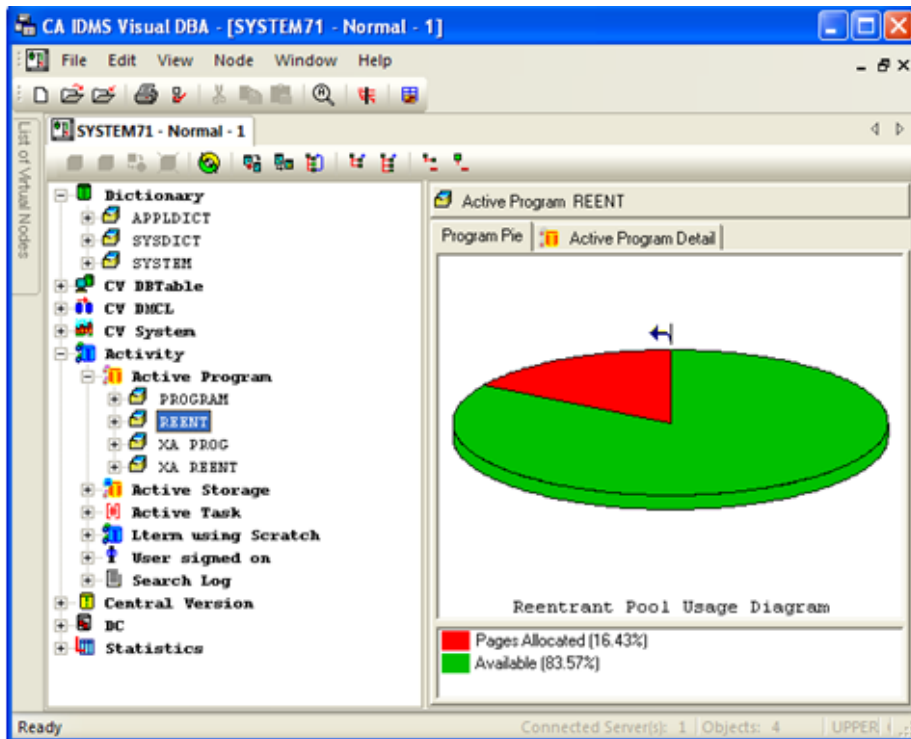
Parameters

In the following example, the parameters for the procedure SQLROUT.TIF1 are shown in the Parameters tab.

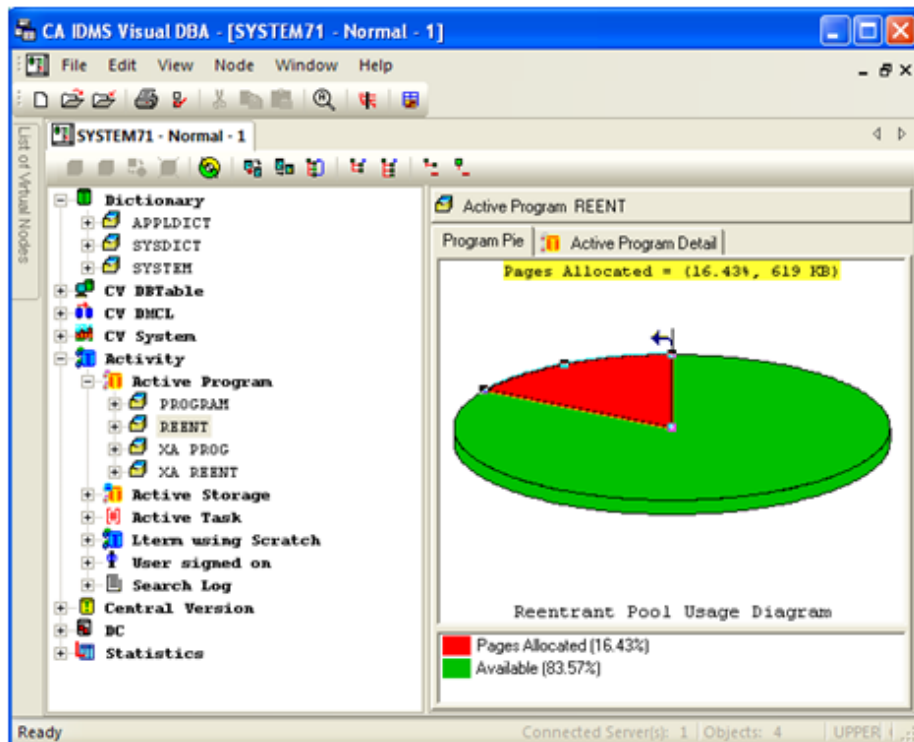


Pie

In the following example, Reentrant Pool usage is displayed in pie chart format.

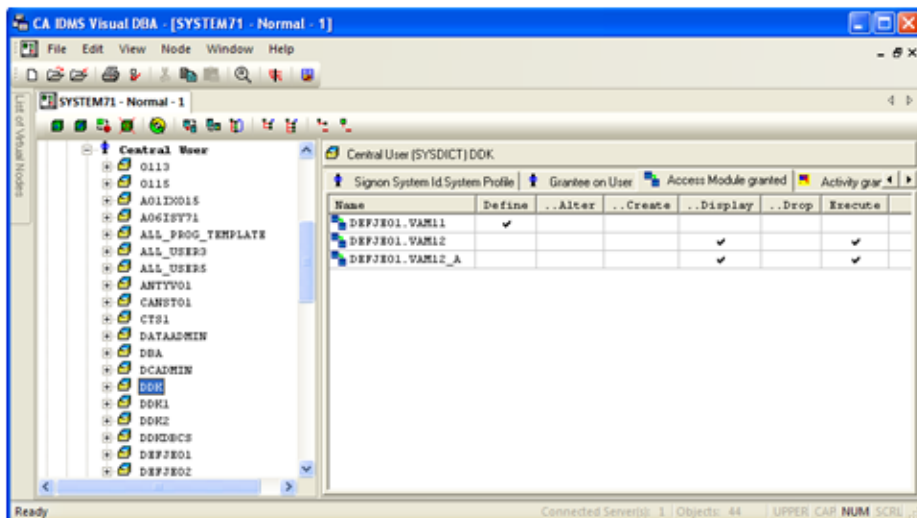


When you click a pie segment, the actual usage statistic is displayed, as shown:



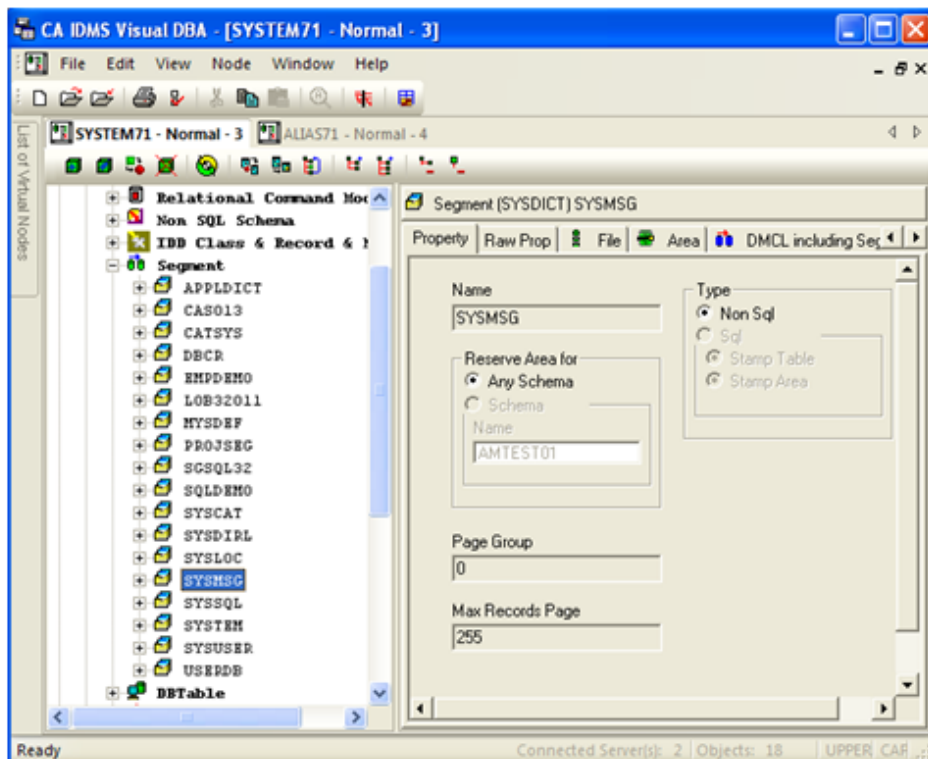
Privilege

In the following example, Privilege information is displayed for Central User DDK. 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.

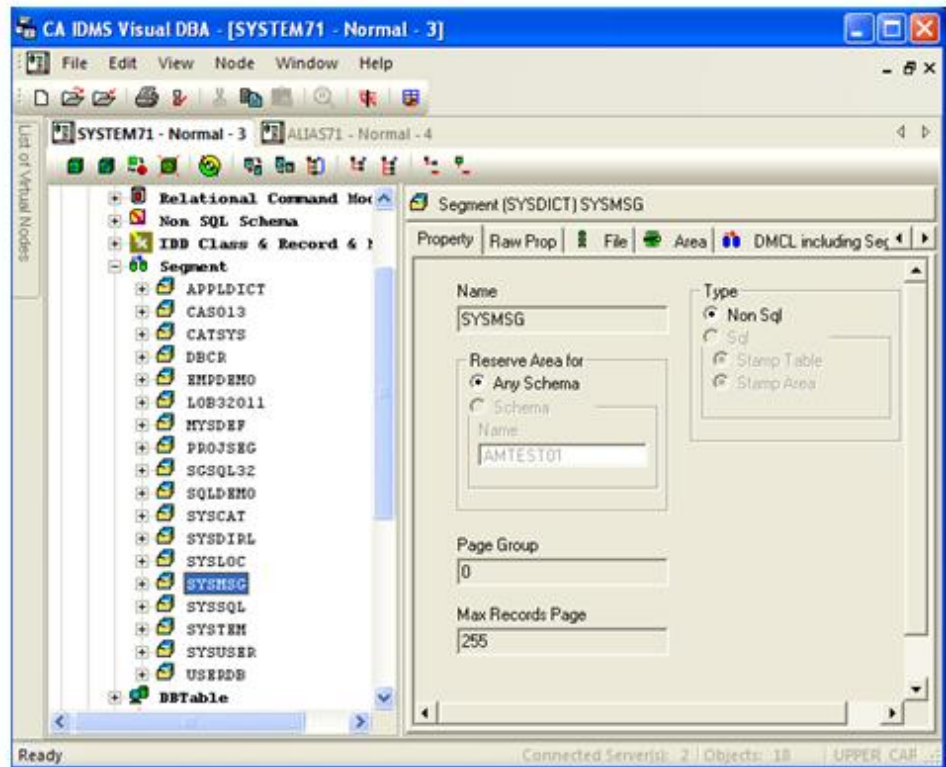


Property

In the following example, Property tab information is displayed for the SYSDICT Segment, EMPDEMO. Note that the caption in the right pane repeats the identification information of the object and its icon. Following the caption, a series of labeled tabs provides different types of attribute information.

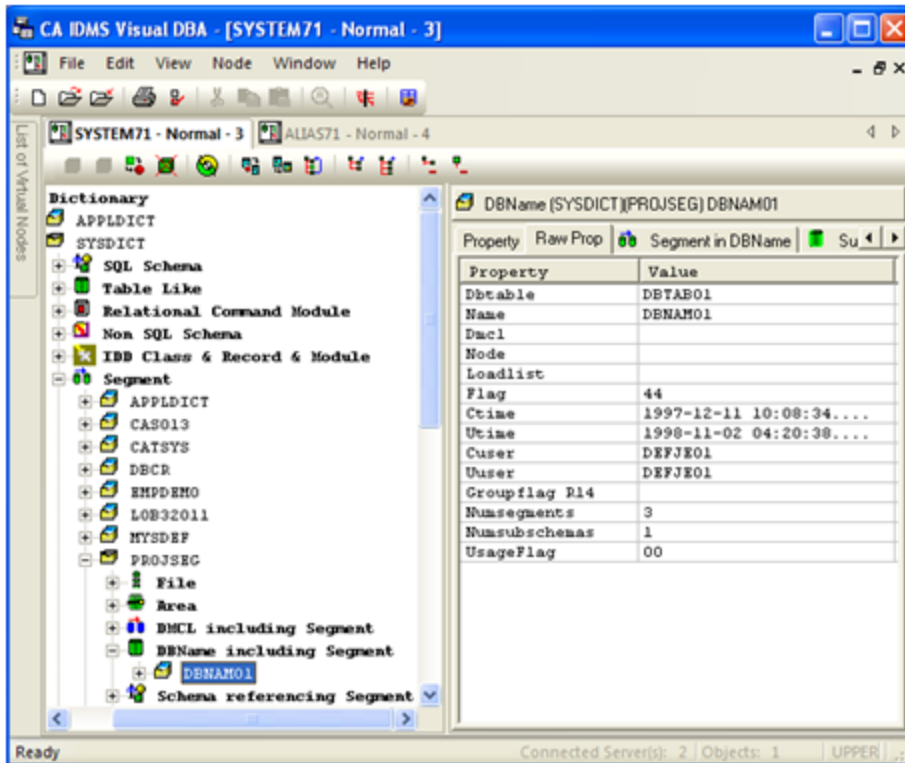


As illustrated in the next example, we can display Property information for another instance of the Segment object class by simply selecting that object from the tree. The tab selection is retained, enabling you to easily make comparisons between object classes and instances.

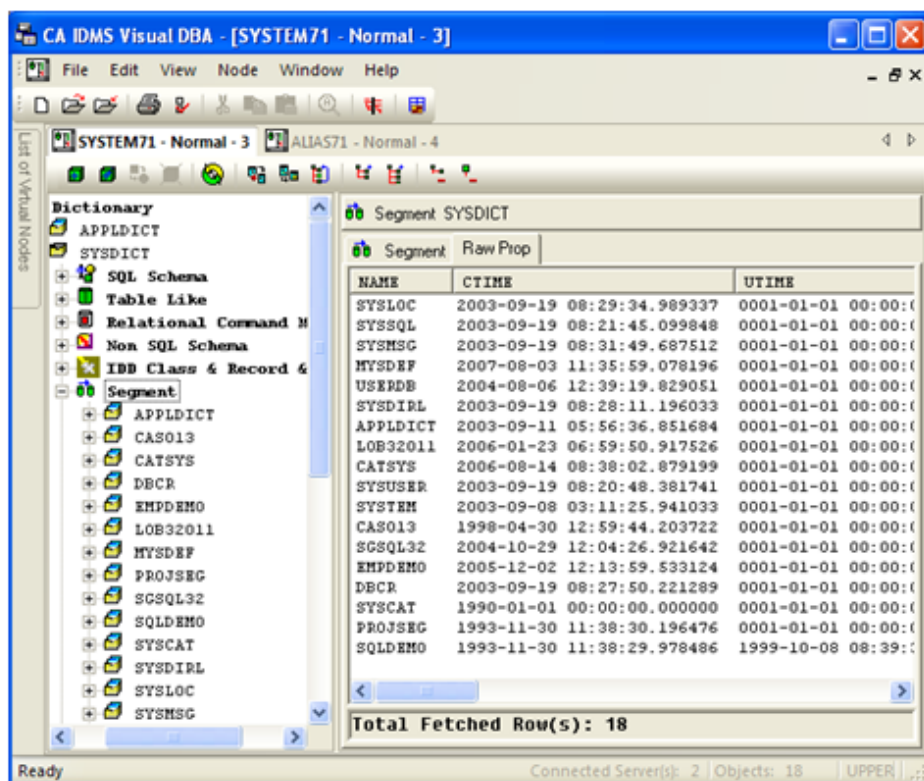


Raw Prop

In the following example, Raw Prop information is displayed for the DBNAM01 instance of the Object Class DBName including Segment.

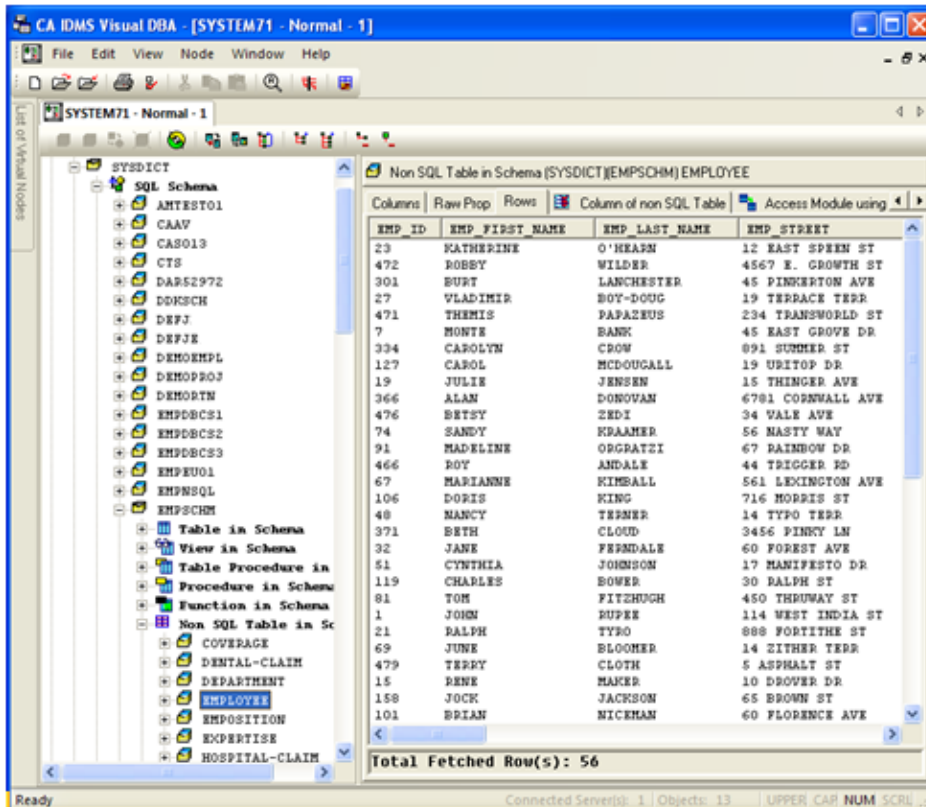


When you select an Object Class or static object, a table is displayed with a column for each attribute and a row for each object instance.



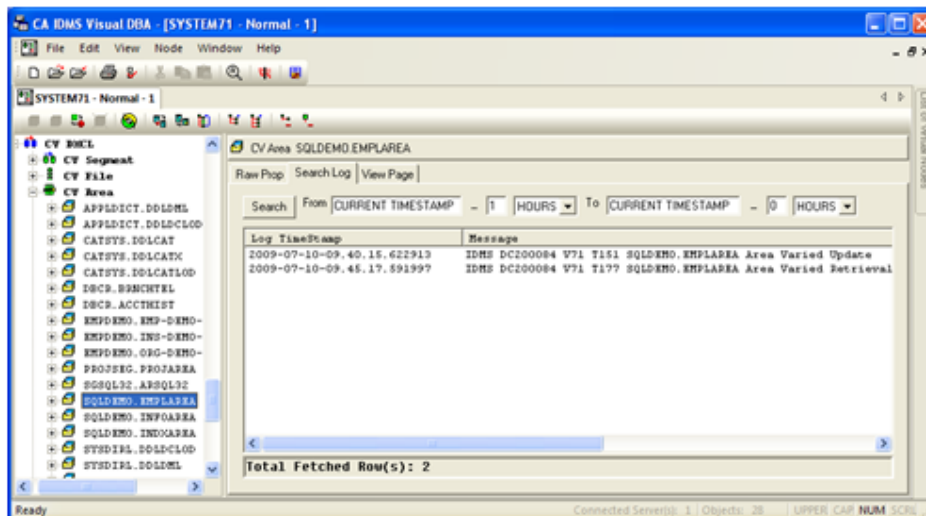
Rows

In the following example, row information is displayed for EMPSCHEM.EMPLOYEE, an instance of a Table Like object in the Non-SQL Table in Schema Object Class.



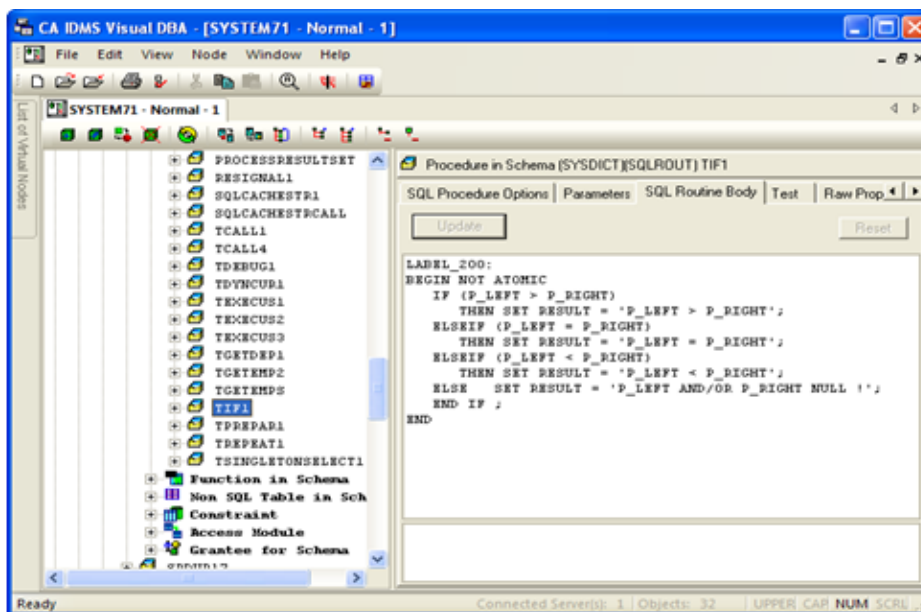
Search Log

In the following example, information in the CA IDMS log related to the CV Area SQLDEMO.EMPLAREA has been retrieved.



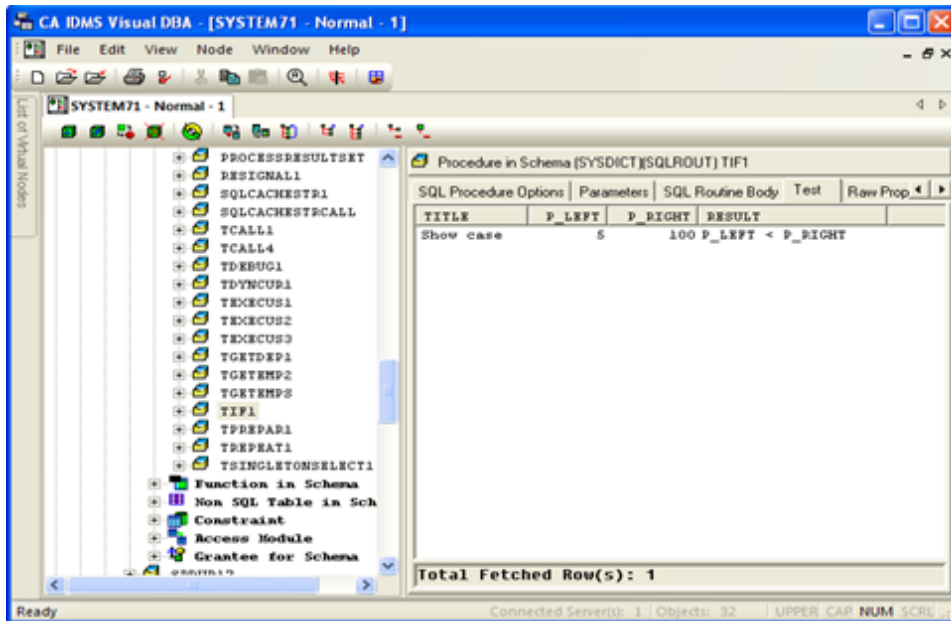
SQL Routine Body

In the following example, the SQL routine body for the procedure SQLROUT.TIF1 is shown in the SQL Routine Body tab.



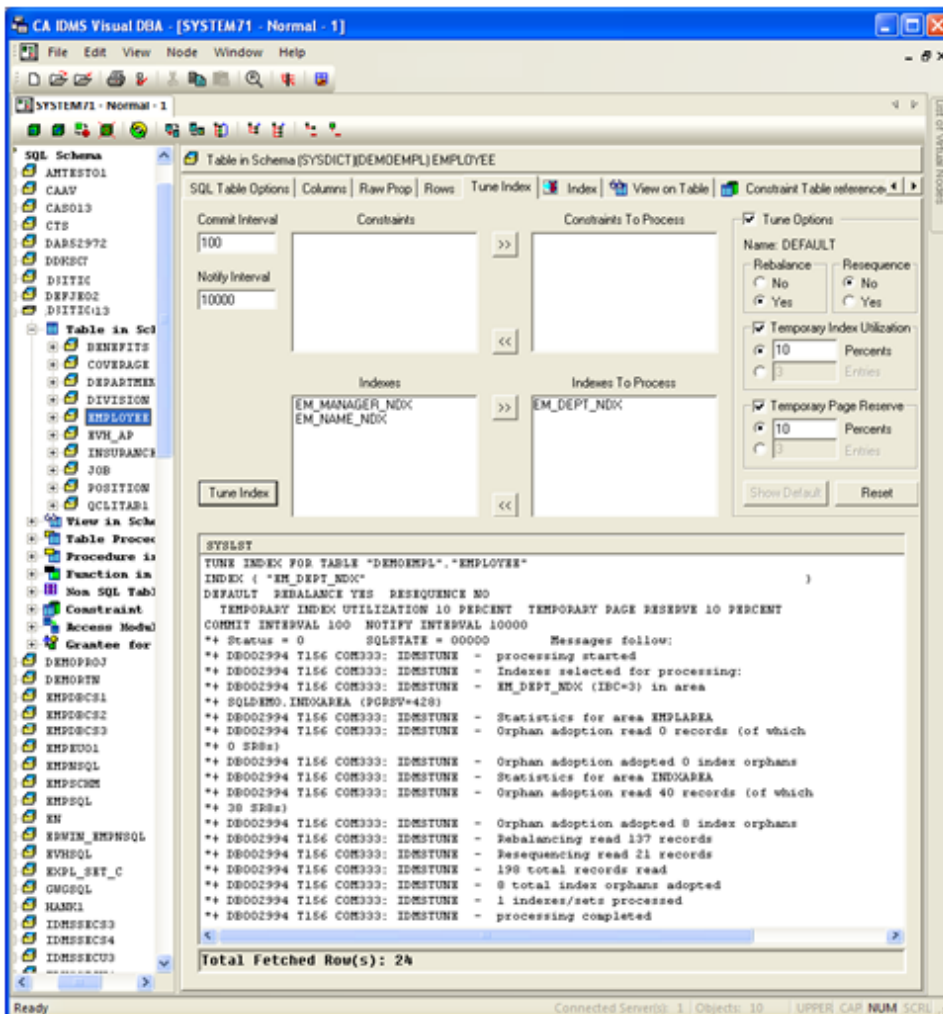
Test

In the following example, the result of executing SQL procedure SQLROUT.TIF1 is shown in the Test tab.



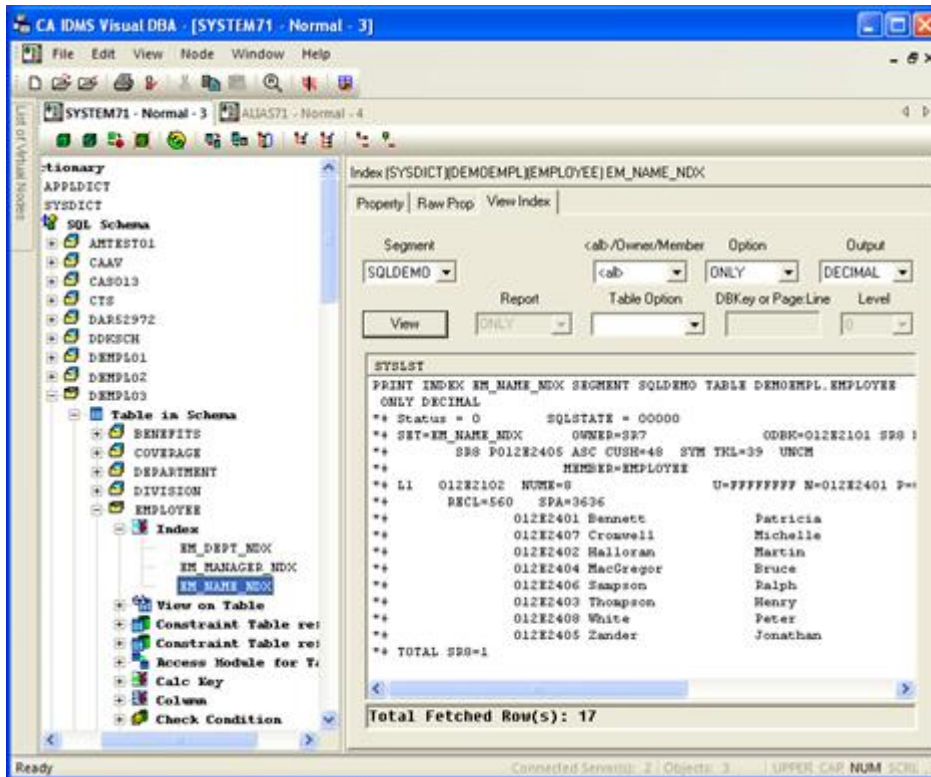
Tune Index

In the following example, the Tune Index utility has been run on the index of the EMPLOYEE Table in Schema.



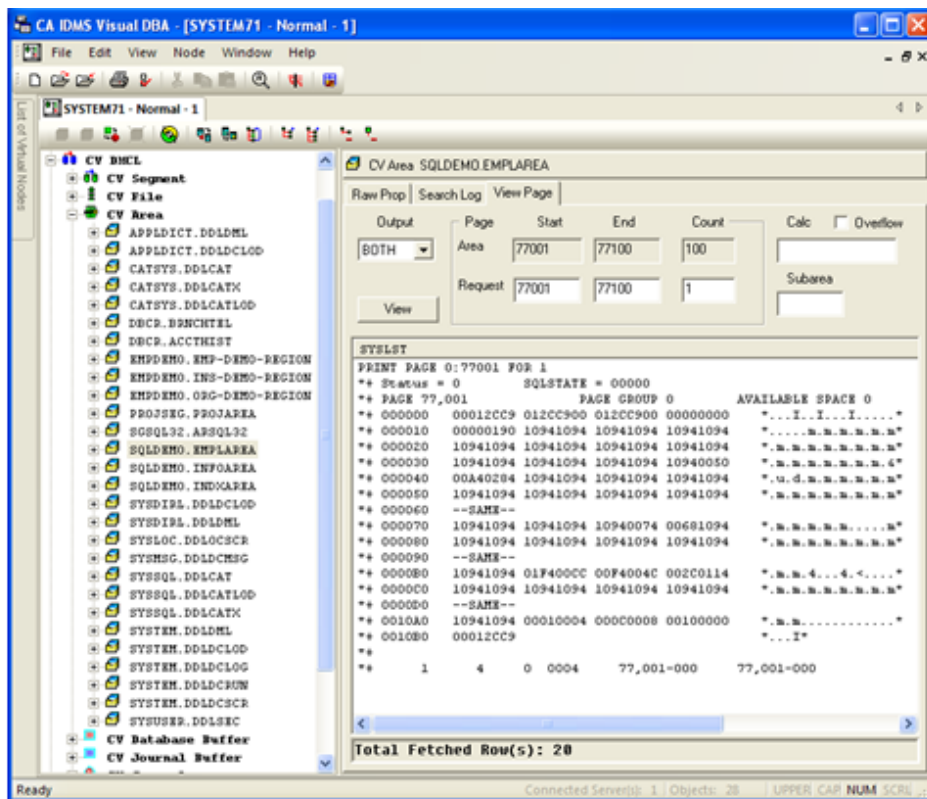
View Index

In the following example, index information is displayed for the EM_NAME_NDX index of the Table in Schema EMPLOYEE.



View Page

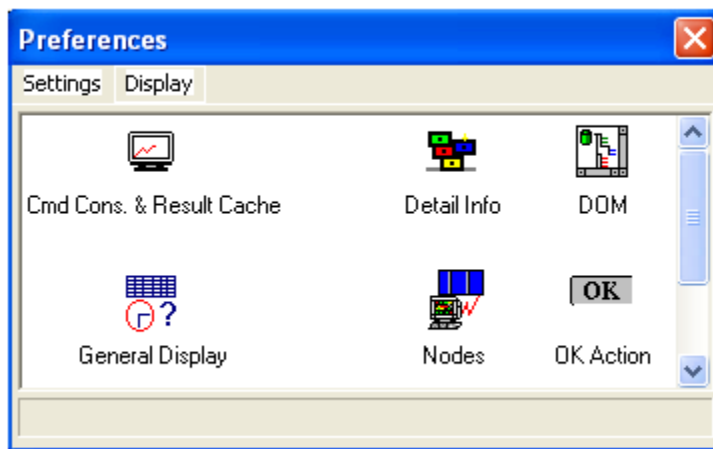
In the following example, database page information is displayed for the CV Area SQLDEMO.EMPLAREA.








Set Display Options for Your Session





You can use the Display options to customize the CA IDMS Visual DBA environment.

The Display options are found under File Preferences option.



These include the following options:

Option	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.

Option	Description
 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.
 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.

Create, Open, and Save Environments

CA IDMS Visual DBA gives you so many ways to customize your session options and object tables. It lets you save your customized environments so that you do not have to redefine 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.

Try It Out Yourself

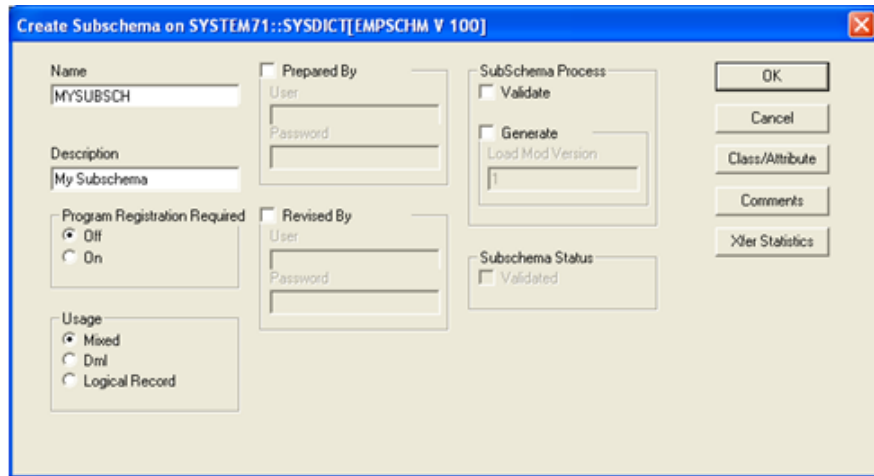
You can now apply the features of the Database Object Manager to work. In this example, we create and modify a non-SQL subschema based on an existing subschema definition. To do this, we use the following:

- 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 the CA IDMS installation. Make sure that your profile includes viewing and updating the Commonwealth demo database. In this example we assume that you are granted update access to the Dictionary object. After logging on, drill down the Dictionary branch to the subschema object by double-clicking the object or simply clicking the + icon that prefixes the object and instance, as follows:

- Dictionary object
- The APPLDICT dictionary instance
- Non-SQL schema object
- The EMPSCHM V 100 instance

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:



The Database Object Manager is refreshed and the name of the subschema you just created is displayed.

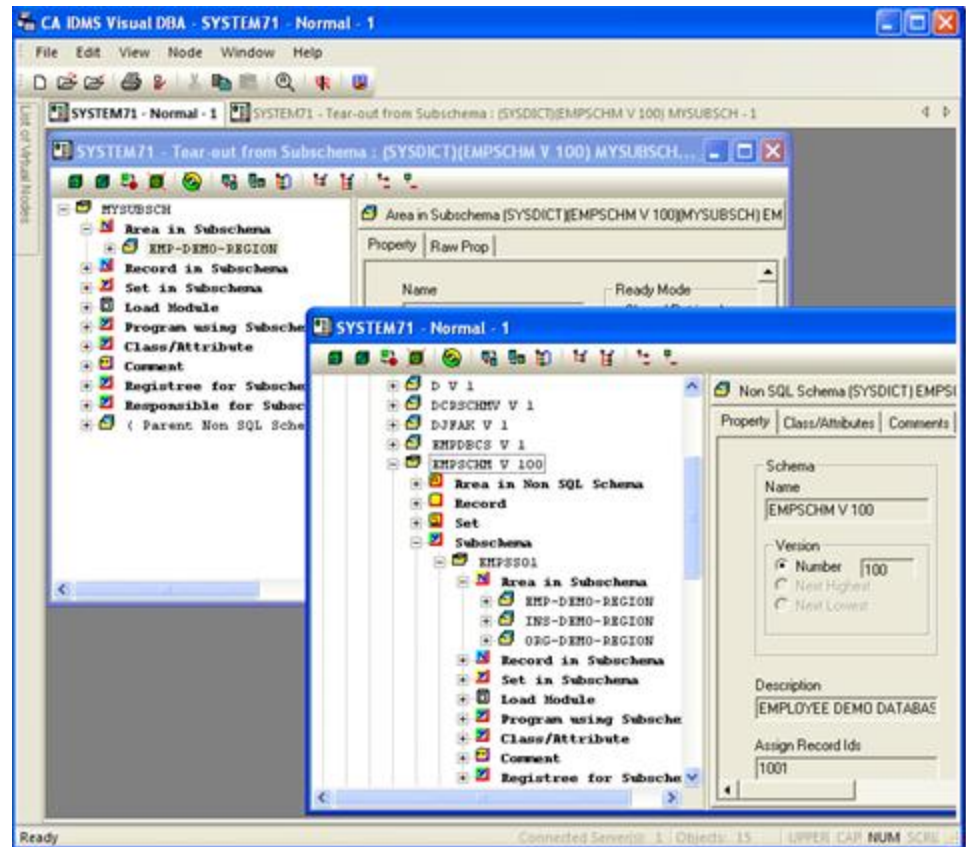
You can now populate  the new subschema with information copied from an existing subschema. To make it easier to view, first 'tear-out' the subschema you created and display it in a new window. To do this, select the subschema, MYSUBSCH, and click Tear Out from the Window menu. A new window appears with 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.

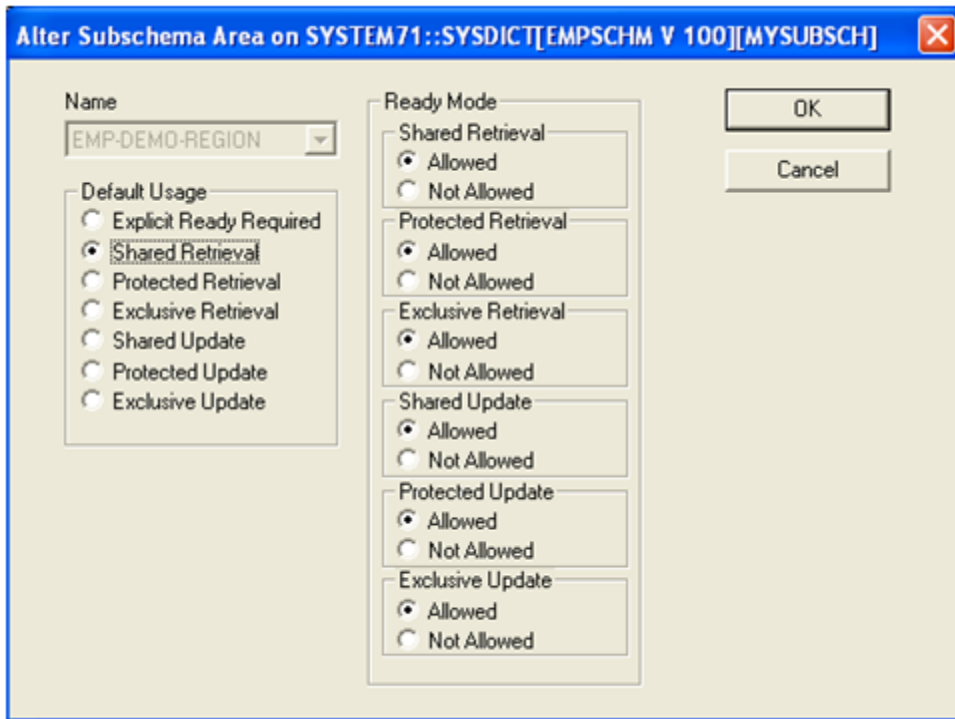
To copy the EMP-DEMO-REGION area to the new subschema, drag-and-drop EMP-DEMO-REGION from the original window to the Area in Subschema object of the 'Tear-Out' window.

Note: The mouse cursor changes to the Area icon when it is positioned over Areas in Subschema.

If you successfully copied the EMP-DEMO-REGION area, your window should look something like this:



Finally, you can modify the default usage of the EMP-DEMO-REGION area in the new subschema. To do this, select it and right-click. From the pop-up 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.

Chapter 5: 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


This section contains the following topics:

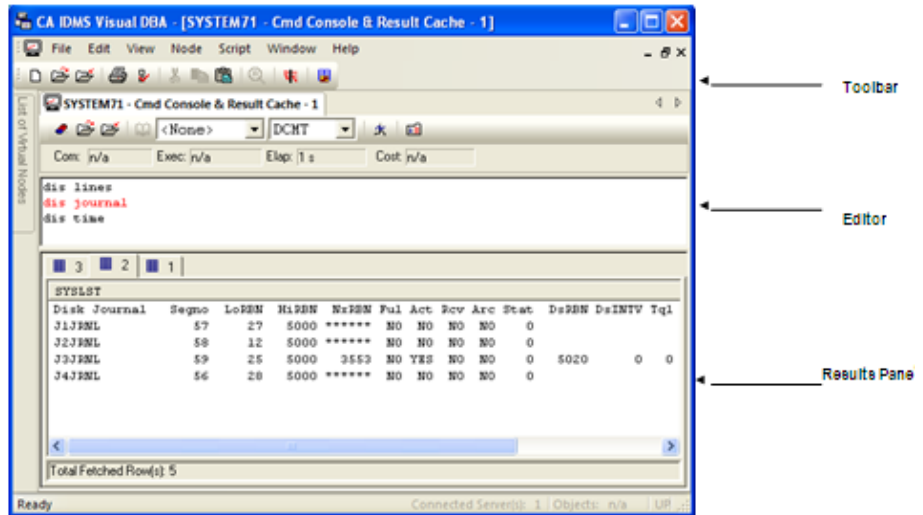
[Command Console Workspace](#) (see page 102)

[Using the Editor](#) (see page 113)

Command Console Workspace

The Command Console Workspace includes a number of features that simplify working with CA IDMS command processors.



To open the Command Console, click the Command Console button , included on the Nodes window toolbar, or select Cmd Console from the Node menu. The Command Console Workspace includes a number of features that simplify working with CA IDMS command processors as follows.



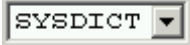
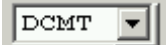




Command Console Toolbar

The Command Console toolbar enables you to perform the following functions:



Button	Description
	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 where you can choose the script that you want to open.

Button	Description
	Saves the contents of the Command Console editor. Choosing the Save Script command invokes the Save As (script) dialog 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. Note: If your profile does not allow you to run the selected command processor on the selected dictionary, this button and the associated command remains disabled.
	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, the source command for that tab is highlighted in red in the Editor window as shown in the following example.

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

Command Console Results Pane

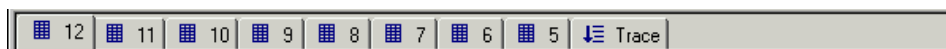
The Command Console Results pane displays the results of executing scripts.

SYSLSST	RETCODE	SEQUENCE
CURRENT TIME 19:43:28.17	0	1
CURRENT DATE 04/227	0	2
STARTUP TIME 05:17:36.40	0	3
STARTUP DATE 04/223	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 PROC RHDCEBYE	0	11

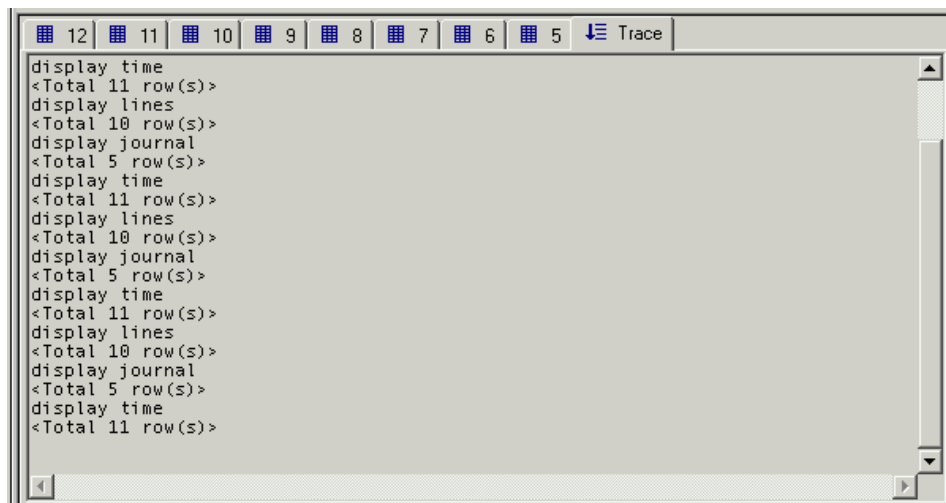
Total Fetched Row(s): 11


It includes numbered tabs that display the execution results in sequential order, with the highest number representing the most recently executed command. Click 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.

Note: For more information, see Set Command Console Preferences.



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




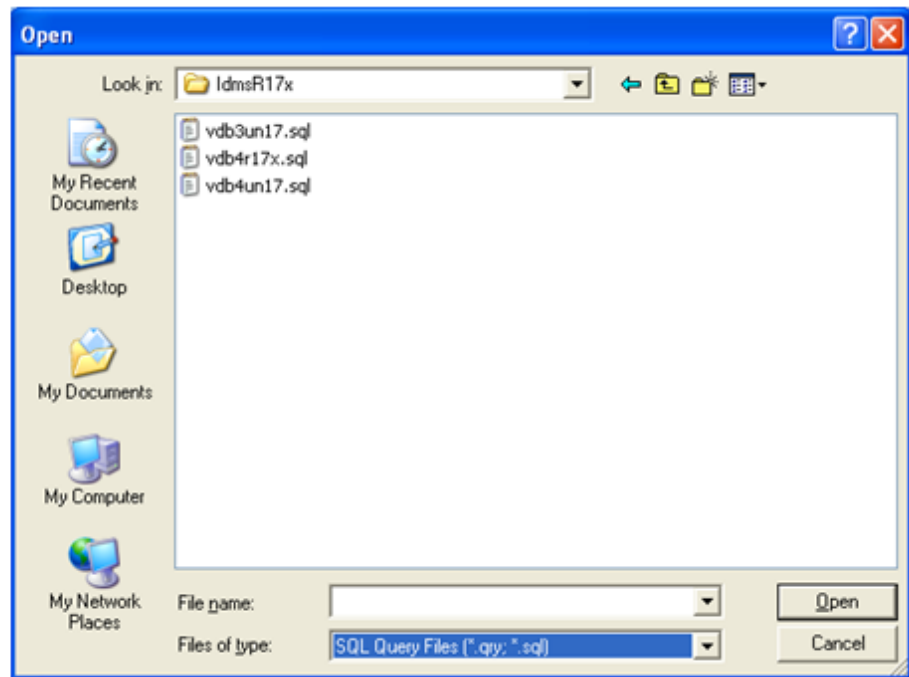
You can toggle the display of the Trace tab on and off by clicking the Trace button  on the Command Console toolbar or by selecting and deselecting the Trace window command on the Script menu.

Run a Command Script


Use the following steps to run a command script.


To run a command script

1. Select a processor from the processor drop-down list box.
2. If you are running a script for the SQL, OCF, IDD, Schema, Sysgen, or SSC processors, you must select a dictionary from the dictionary drop-down list box.
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. The Open dialog displays 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.

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, the Save dialog displays where you can save the script and execute it at a later time.

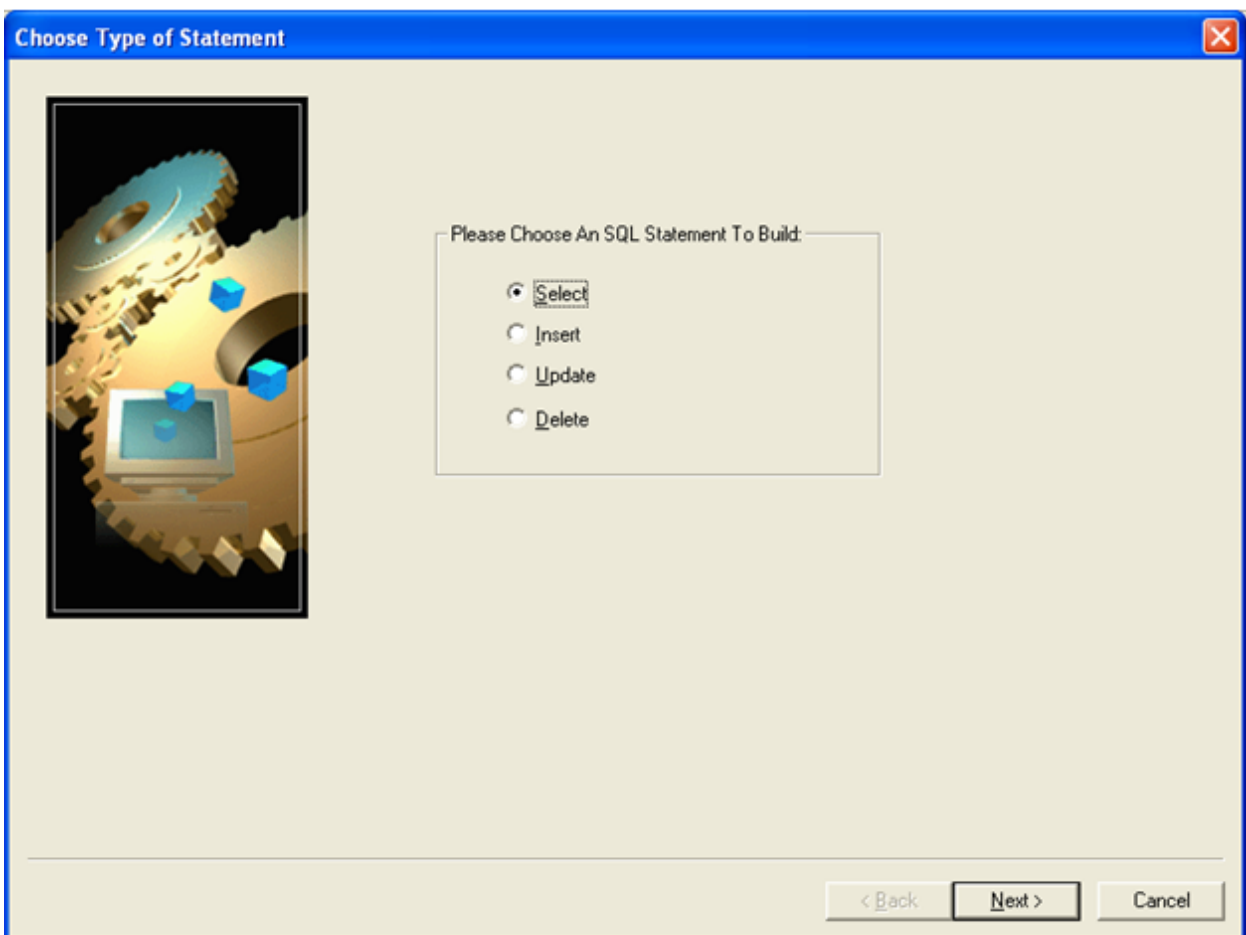
Use the SQL Assistant

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

To use the SQL Assistant

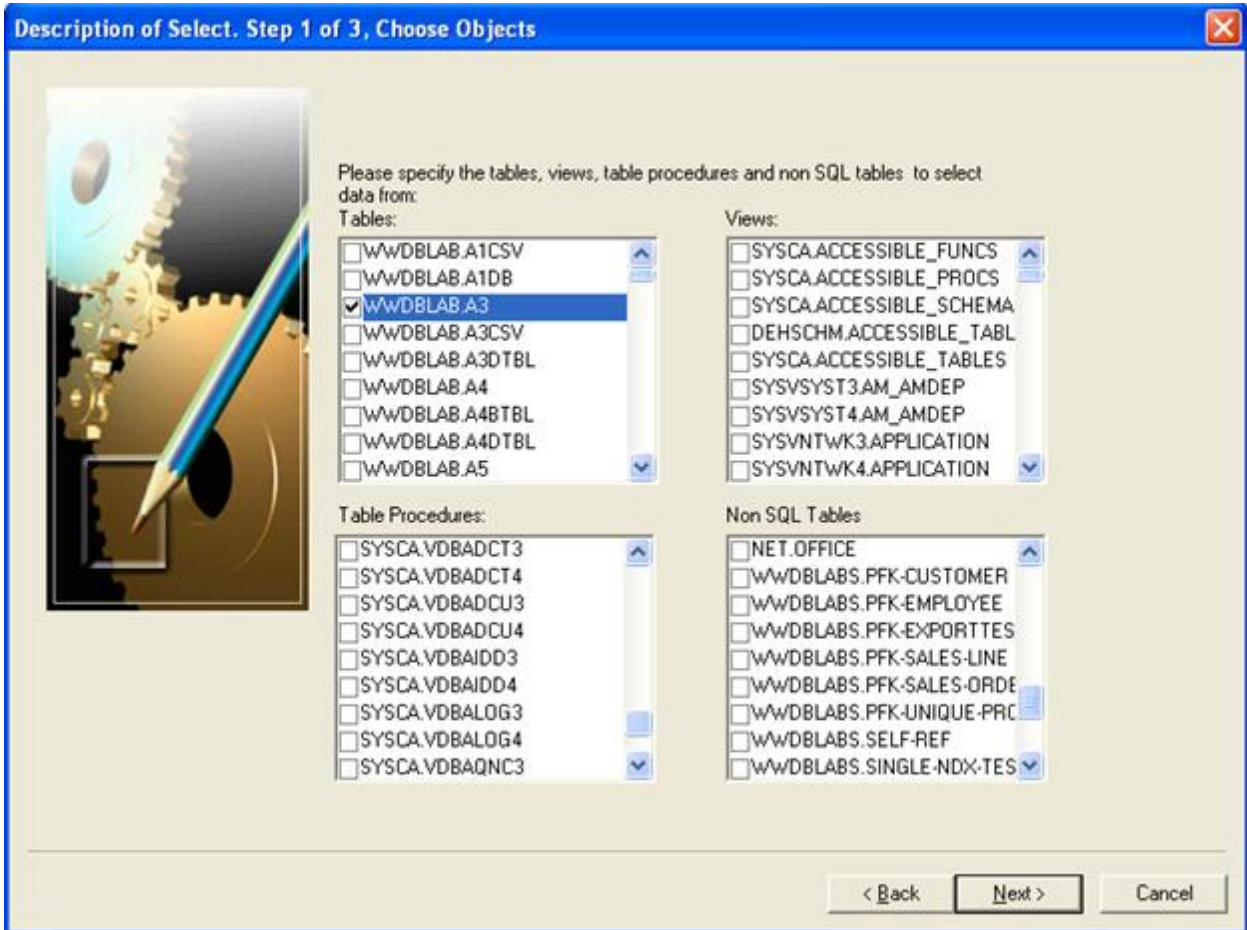
1. Select a dictionary from the Select Dictionary drop-down list box.
2. Start the SQL Assistant by clicking the SQL Assistant button  on the Command Console toolbar or by selecting the SQL Assistant command from the Script menu.

The Choose Type of Statement dialog opens, enabling you to pick the type of SQL command statement that you want to execute.



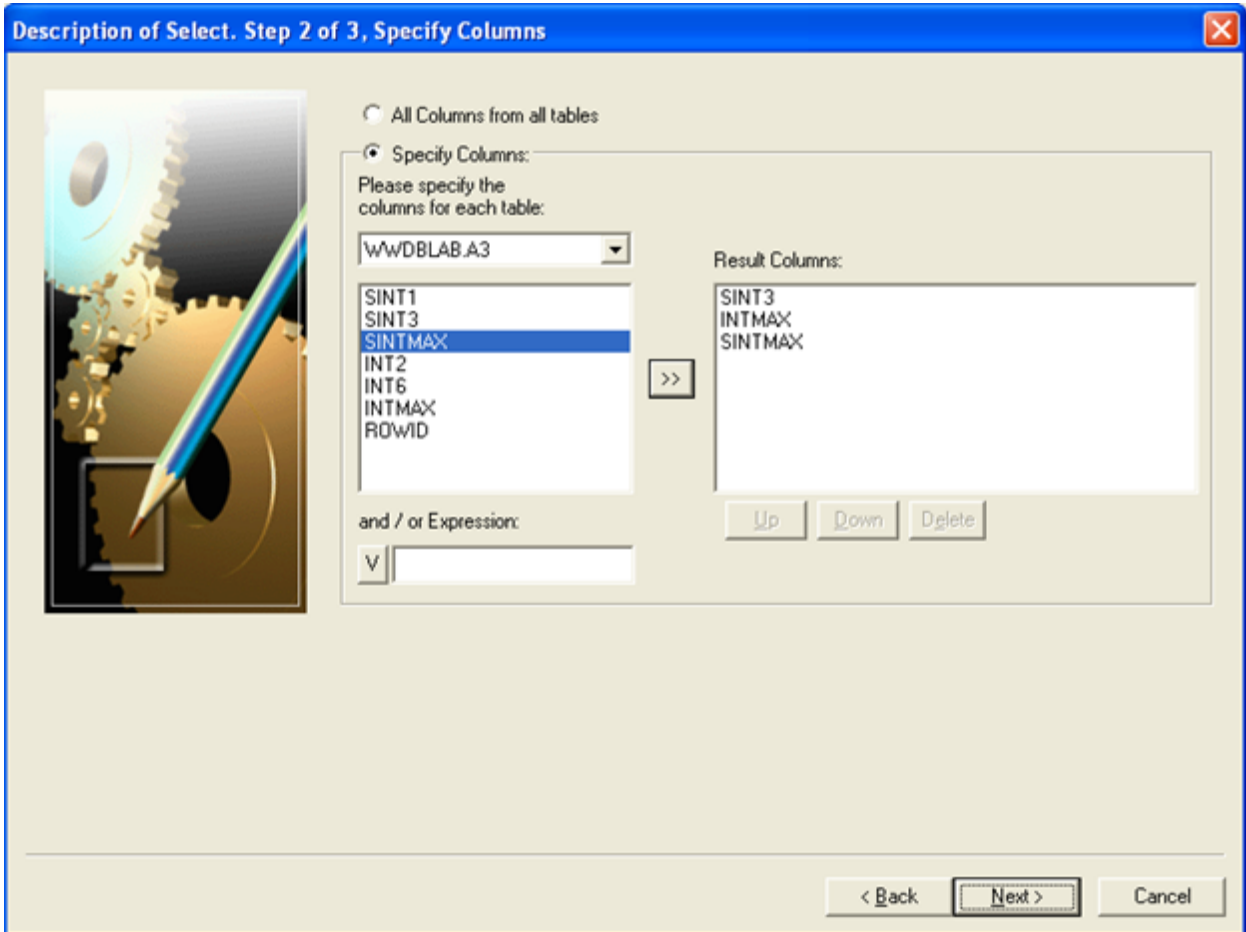
3. Make your selection and click Next.

The first of three Choose Objects dialogs displays. Here you can select the Tables, Table Procedures, Views, and Non-SQL Tables that you want to use as objects.



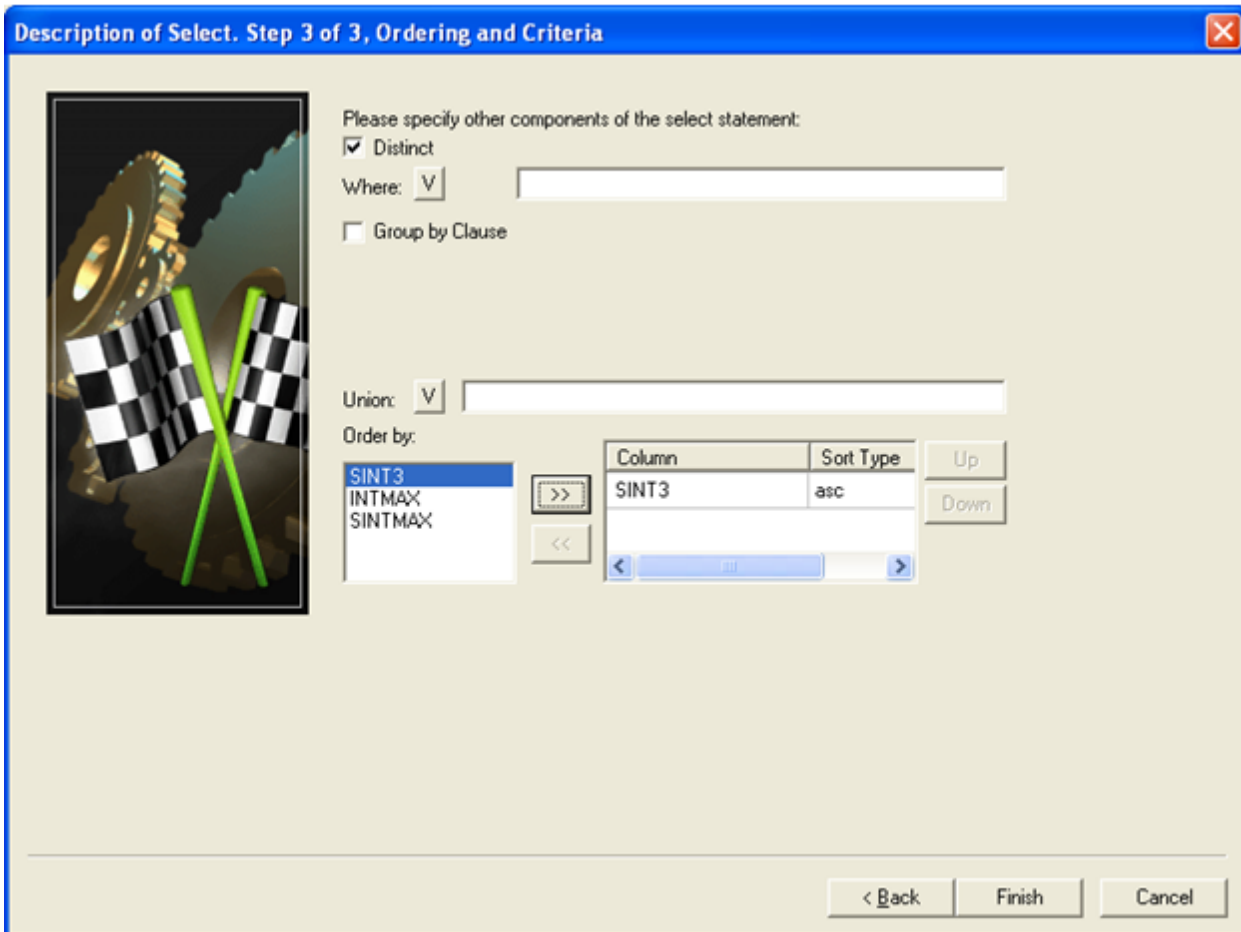
4. Click Next.

A dialog displays 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. Click finish.


The SQL statements execute and the results are displayed in the Results pane.

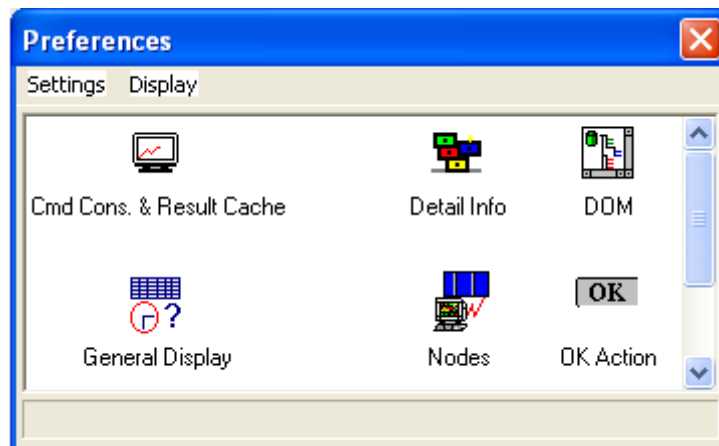


Set Command Console Preferences

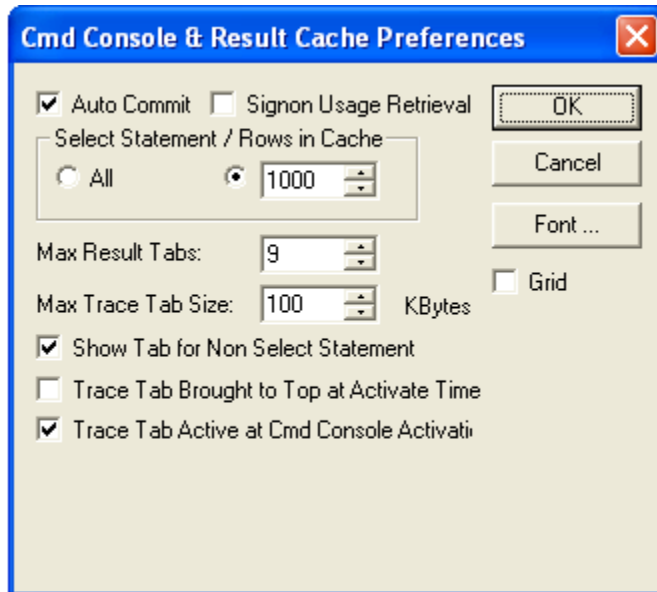
You can set the command console preferences using the following steps.

To set preferences for the Command Console

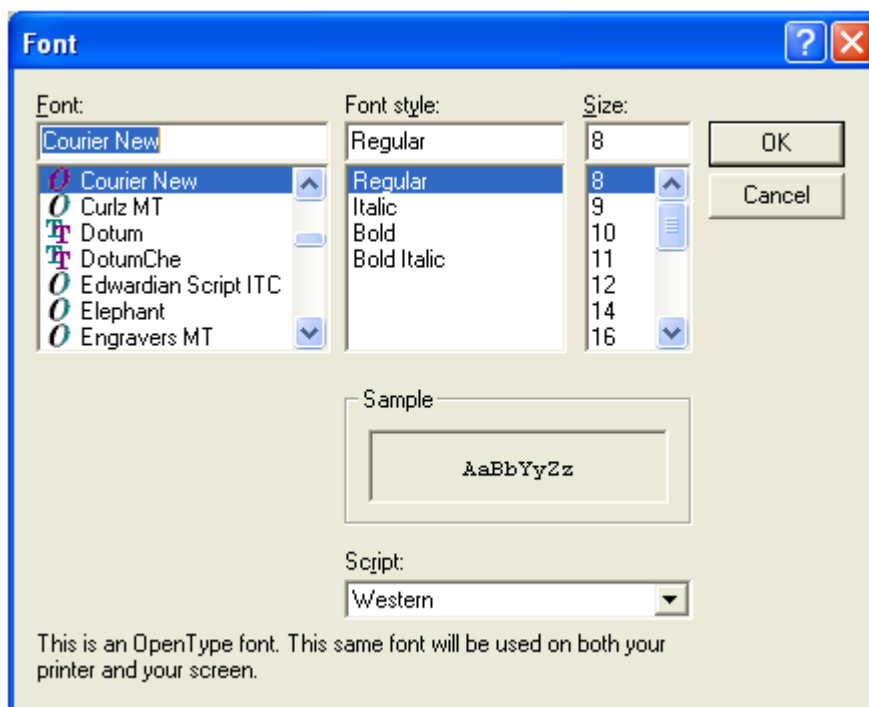
1. Click the Preferences button  on the Workspace toolbar or select the Preferences command from the File menu. Both actions open the Preferences dialog.



2. Double-click the Cmd Console icon. The Cmd Console Preferences dialog opens where you can do the following:
 - Select trace options.
 - Define how many records to 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 continues to function until the cache becomes full once again.
 - Specify how many history tabs to provide.
 - Set an indicator that determines whether changes made to the data during the execution of an SQL query are 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.



Using the Editor

The text editor is available both in the input pane of the command console and in the Detail Information Pane (DIP) of some objects.

The Detail Information Pane (DIP) of the following DOM objects contains a Tab that invokes the text editor dialog:

DOM Objects using a tab labeled *SQL Routine Body*

- Dictionary/SQL Schema/Procedure in Schema
- Dictionary/SQL Schema/Function in Schema
- Dictionary/Table Like/Procedure
- Dictionary/Table Like/Function

DOM Objects using a tab labeled *Module Text*

- Dictionary/IDD Class & Record & Module/Module
- Dictionary/IDD Class & Record & Module/.Assembler Module
- Dictionary/IDD Class & Record & Module/.Cobol Module
- Dictionary/IDD Class & Record & Module/.Culprit Module
- Dictionary/IDD Class & Record & Module/.DC Module
- Dictionary/IDD Class & Record & Module/.OCF Module
- Dictionary/IDD Class & Record & Module/.OLQ Module
- Dictionary/IDD Class & Record & Module/.PL/I Module
- Dictionary/IDD Class & Record & Module/.Process Module

Editing functions and Accelerator Keys

The following table lists the editing functions with the associated accelerator keys and descriptions.

Function	Accelerator	Description
Copy	Ctrl-C	Copies selected text to clipboard
Cut	Ctrl-x	Copies selected text to clipboard and delete
Find	Ctrl-F	Invokes Find dialog
Find Next	F3	Finds next occurrence
Paste	Ctrl-V	Pastes (inserts) text from the clipboard
Print	Ctrl-P	Prints text
Redo	Ctrl-Y	Redoes changes that have been undone
Replace	Ctrl-H	Invokes Find and Replace dialog
Select All	Ctrl-A	Selects all text
Undo	Ctrl-Z	Reverses last change

Drag-and-Drop Support

Standard Windows drag-and-drop support is available.

Appendix A: Use the Demo180.cfg File

The CA IDMS Visual DBA product folder contains an example saved configuration file named `demo180.cfg`. This file can be used to do the following:

- 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 `demo180.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 `demo180.cfg`.

The Dictionary object in the saved configuration has three 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 can 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 attempts 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 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 `demo180.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 is not executed.

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

Appendix B: Third Party Software Acknowledgements

This product includes ZLib. CA is grateful to the authors for making it available for inclusion in this software.

Index

A

- Application window • 49
- Assigning privileges
 - create, alter and drop object instances • 47
 - granting and revoking • 47
 - to objects • 46
 - to update object instances • 46

B

- Bar tab • 78

C

- CA IDMS
 - objects you can manage • 11
 - viewing and maintaining multiple systems • 25
 - what you need to know • 26
- CA IDMS Visual DBA
 - Application window • 49
 - customizing the user interface • 25
 - features • 10
 - securing objects • 45
 - starting your DBA session • 49
- Command console • 25
 - Editor • 103
 - Preferences • 111
 - Results pane • 104
 - toolbar • 102
 - Trace window • 104
- Command script, running • 105

D

- Database Object Manager window • 49, 60
 - Detail Information pane • 60
 - Object Tree pane • 60
 - opening • 56
 - toolbar • 62
- Datasource
 - adding • 53
 - connecting to • 53
- Detail Information Pane
 - Bar tab • 78
 - displaying attributes • 75
 - from DOM window • 60
 - Grantee tab • 78

- Pie tab • 78
- Privilege tab • 78
- Property tab • 78
- Raw Prop tab • 78
- Rows tab • 78
- Display options, setting • 96

E

- Environments, saving • 97

G

- Grantee tab • 78
- Granting and revoking object privileges • 47

H

- Help, using online • 26

M

- Managing objects using drag and drop • 24
- multiple CA IDMS systems, using • 25

N

- Nodes window • 49
 - toolbar • 52

O

- Object tree
 - expanding and collapsing • 63
 - refreshing • 73
 - viewing object types • 60
- Objects
 - assigning privileges to • 46
 - Displaying attributes • 75
 - finding • 71
 - restricting access • 46
 - securing • 45
 - selecting and copying • 72
 - viewing syntax • 72
 - you can manage • 11

P

- Pie tab • 78
- Preferences, setting • 96, 111

Privilege tab • 78
Property tab • 78

R

Raw Prop tab • 78
Restart from Position • 65
Rows tab • 78

S

Scratchpad • 68
Securing objects • 45
SQL Assistant • 107
Starting your CA IDMS Visual DBA session • 49

T

Tear-Out Window • 67
Toolbars
 Command Console • 102
 DOM window • 62
 Nodes window • 52
 Workspace • 50
Tree objects, manipulating • 70

U

User interface, customizing • 25

W

Workspace Environment • 49
Workspace toolbar • 50