

Oracle® Documanage

Documanage Database Administrator's Guide

version 6.6

Part number: E14904-01

March 2009

Copyright © 2009, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

THIRD PARTY SOFTWARE NOTICES

This product includes software developed by Apache Software Foundation (<http://www.apache.org/>).

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2000-2009 The Apache Software Foundation. All rights reserved.

This product includes software distributed via the Berkeley Software Distribution (BSD) and licensed for binary distribution under the Generic BSD license.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2009, Berkeley Software Distribution (BSD)

This product includes software developed by Jean-loup Gailly and Mark Adler. This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Copyright (c) 1995-2004 Jean-loup Gailly and Mark Adler

This product includes software developed by the Massachusetts Institute of Technology (MIT).

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright © 2009 MIT

This software is based in part on the work of the Independent JPEG Group (<http://www.ijg.org/>).

This product includes software developed by Sam Leffler of Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE

Copyright (c) 1988-1997 Sam Leffler

Copyright (c) 1991-1997 Silicon Graphics, Inc.

This product includes software components distributed by Steve Souza.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2002, Steve Souza (admin@jamonapi.com). All Rights Reserved.

This product includes software components distributed via the Berkeley Software Distribution (BSD).

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2000-2006 www.hamcrest.org. All Rights Reserved.

This product includes software components developed by the Independent JPEG Group and licensed for binary distribution under the Independent JPEG Group license.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 1994-1998 AIIM International. All Rights Reserved.

This product includes software components developed by Sam Stephenson.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2005-2007 Sam Stephenson

This product includes software components developed by Sun Microsystems.

Copyright (c) 1995-2008 Sun Microsystems, Inc. All rights reserved.

This product includes software components distributed by Vbnet and Randy Birch.

Copyright © 1996-2008 Vbnet and Randy Birch. All Rights Reserved

This product includes software components distributed by the Internet Software Consortium and IBM.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1996 by Internet Software Consortium.

THE SOFTWARE IS PROVIDED "AS IS", AND IBM DISCLAIMS ALL WARRANTIES, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL IBM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE, EVEN IF IBM IS APPRISED OF THE POSSIBILITY OF SUCH DAMAGES.

Portions Copyright (c) 1995 by International Business Machines, Inc.

This product includes software components distributed by RSA.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

Copyright © 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

This product includes software components distributed by Terence Parr.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2003-2007, Terence Parr. All rights reserved.

This product includes software components distributed by Computer Associates.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2002 Computer Associates. All rights reserved.

This product includes software components distributed by MetaStuff.

THE SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS "AS-IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS AND SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING IN ANY WAY OUT OF THE USE OF THE SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2001-2005 Metastuff, Ltd. All Rights Reserved.

This product includes software components distributed by JSON.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright (c) 2002 JSON.org

This product includes software components distributed by OpenSSL (<http://www.openssl.org/>).

THIS SOFTWARE IS PROVIDED BY THE OPENSSL PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENSSL PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

Copyright © 1998-2007 The OpenSSL Project. All rights reserved.

This product includes software components distributed by Yahoo! Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2008 Yahoo! Inc. All Rights Reserved.

Publication history

First issue for Version 6.4: July 2004

Revision 1 for Version 6.4 Service Release 1: December 2004

Revision 2 for Version 6.4 Service Release 2: May 2005

Republished for Version 6.4 Service Release 3: February 2006

Revision 3 for Version 6.4 Service Release 4: October 2005

Revision 4 for Version 6.4 Service Release 5: June 2006



Republished for Version 6.4 Service Release 6: July 2006

Republished for Version 6.5: September 2007



Revision for Version 6.6: September 2009

Documanage documentation roadmap

Administrating

		
Administrator's Guide	Installation Guide	Database Administrator's Guide

Programming

	
API help	Programmer's Guide

Workstation



	
Workstation Guide	Workstation help

Table of Contents

XVI PREFACE

- xvi Introduction**
- xvi Using this manual**
- xvii Contents**
- xviii Conventions**
- xviii Related documents**
- xix Suggestions**

1 DOCUMANAGE ARCHITECTURE

- 1 In this chapter**
- 1 Introduction**
- 2 Architectural Advantages**
- 2 Database Access**
- 3 Database Connections**

5 DATABASE TABLES

- 5 In this chapter**
- 5 Introduction**
- 6 Housekeeping Tables and Line of Business Tables**
- 6 Housekeeping Tables**
 - 7 OT_Docs Housekeeping Table**
 - 7 Accesscommander /390 Housekeeping Tables**

Table of Contents

8 PowerMapping

10 Database Names

13 DOCUMANAGE ADMINISTRATOR

13 Relationships

14 Cabinet Definition Dialog

14 System Dialog

14 Configure DSNs Dialog

14 Storage Dialog

15 Categories Dialog

15 Users Dialog

15 Authorities Dialog

16 Locks Dialog

17 STRUCTURE OF DOCUMANAGE DATABASE OBJECTS

17 Tables and Data Bases

18 Row-level Locking

18 Normal Table Space

19 Document Related Tables

20 Workflow Related Tables

20 Administrative Tables

22 Static Tables

25 DOCUMANAGE AUTHORITIES

27 TABLE USE, INDEXING, AND KEY RECOMMENDATIONS

27 Summary

- 27 Tables Actively Used During Normal Operation (10)
- 28 Tables Used During Start-up and Administration (17)
- 28 Static Tables of Codes and Strings Read at Startup (4)
- 28 Tables Used Only by Workflow (16)
- 29 Tables Used Only by HostLink (3)
- 29 Tables Used Only if Database Authentication/Authorization is Used (5)
- 29 Group Level Authorities Tables (1)
- 29 Synchronized Task Tables (1)
- 30 Tables Used for Document Retention Management(4)

30 Table Descriptions and Uses

38 Using Keys and Indices

- 38 Housekeeping tables
- 38 PowerMapped Line of Business tables

41 CHANGING RESERVED NAMES FROM VERSION 5.2 AND EARLIER

43 DOCUMANAGE BACKUP AND RESTORATION

45 APPENDIX A: DATABASE AND TABLE SETUP SCRIPTS

45 MVS Scripts

47 The MVS Scripts directory

Table of Contents

49 APPENDIX B: DATABASE SCHEMA

- 49 Schema Detection**
- 50 SchemaVersion and MinAppSchema**
- 50 Updating the Schema Version**
- 51 Database Schemas Defined**
 - 51 Schema Version 0
 - 52 Schema Version 1
 - 52 Schema Version 2
 - 54 Schema Version 3
 - 55 Schema Version 4
 - 56 Schema Version 5
 - 57 Schema Version 6
 - 58 Schema Version 7
 - 59 Schema Version 8
 - 60 Schema Version 9
 - 61 Schema Version 10
 - 62 Schema Version 11
 - 62 Schema Version 12
 - 63 Schema Version 13
 - 64 Schema Version 14
 - 66 Schema Version 15

69 APPENDIX C: PARAMETERIZED SQL

- 69 Database caching**
- 70 Statement Reuse**

70 SQL Logging

70 Connections and Statement Handles

71 ODBC Driver Cursor Commit Behavior

71 ODBC Driver Bugs

73 APPENDIX D: DATABASE NAMING CONVENTIONS BEFORE VER. 5.2

75 APPENDIX E: STORING DOCUMENTS IN NTFS AND SNAP VOLUMES

75 Storing Subdirectory Paths

76 Subdirectory Path Example

76 Documents Stored Using Versions Earlier Than Service Release 5

77 BLOB and VLAM Storage

Table of Contents

Preface

Introduction

Documanager is the most advanced document-management infrastructure available in today's market. One important feature is the open architecture which allows Documanager to use the most popular SQL database management systems. Documanager isolates applications from database differences and improves efficiency and control by use of a traditional three-tier client/server architecture. This architecture provides enhanced flexibility in deploying Documanager as part of an enterprise infrastructure. Documanager's open architecture allows for open document capture, document storage, document access, workflow, and document processing.

Using this manual

This *Database Administrator's Guide* describes the database structures used by Documanager. The guide discusses Powermapping and its effect on Documanager tables, naming conventions, and the structure of housekeeping tables and their use of keys and indices.

Contents

This manual is organized as follows:

- ◆ **Databases and table categories:** This chapter introduces Documanager and describes the databases that it supports.
- ◆ **Documanager database administrative functions:** This chapter discusses administration functions that perform operations on Documanager tables.
- ◆ **Structure of Documanager database objects:** This chapter discusses the structure of database objects in Documanager.
- ◆ **Documanager authorities:** This chapter discusses user accounts and tasks performed on databases.
- ◆ **Table use, indexing, and key recommendations:** This chapter lists housekeeping table, their structure and their use of keys and indices.
- ◆ **Changing authentication and table owner names after release 5.2:** This chapter describes how administrators can change previously reserved authentication ID and table owner names for releases later than 5.2.
- ◆ **Backing Up and Restoring OT_ Tables:** This chapter describes how to backup and restore the housekeeping tables and storage system elements for Documanager.
- ◆ **Appendix A: Database and Table Setup Scripts** briefly describes the OT_* Tables and Database Setup Scripts on your Documanager CD.
- ◆ **Appendix B: Database Schema** lists each schema version and describes its structure.

- ◆ **Appendix C: Parameterized SQL** explains parameterized SQL and how you can use it to improve database access.

Conventions

The *DBA Guide* provides consistent typographic conventions and keyboard formats to help you locate and interpret information easily. These conventions are provided below.

Typographic and keyboard conventions

Convention	Description
<i>Italics</i>	Command, dialog box, icon, and field names
San serif font	Directory, folder, and file names
1 Numbered lists	Step-by-step procedures for performing actions
◆ Bulleted lists	Provide grouped information, not procedural steps

Related documents

In addition to this manual, the following related publication(s) are also available from Skywire's Documanager for Oracle Software:

- ◆ *Planning Guide*
- ◆ *Workstation Guide*
- ◆ *Administrator's Guide*

Preface

Suggestions

- ◆ *Programmers Guide to Documange*
- ◆ *Dmg_API online help*

Suggestions

We welcome your comments, suggestions, and concerns about this manual or any Oracle Software publication.

Send your comments to:

Skywire's Documange for Oracle Technical Documentation
3353 Peachtree Road NE, Ste 800, Atlanta, Georgia 30326

Documange Architecture

In this chapter

This chapter will help you answer the following questions:

- What kind of an architecture does Documange use?
- How do the different Documange modules access the Database Management System?
- How do Documange Workstations (clients) and the Database Management System connect to the Documange Server?

Introduction

Documange uses a secure and efficient three-tier architecture. Documange accesses SQL DBMSs using the Open Database Connectivity (ODBC) standard.

Documange consists of the following tiers:

- ◆ Tier 1—The database management system (DBMS) hosts, which only accept connections from Tier 2 Servers and applications.
- ◆ Tier 2—The Documange Server(s) and Server applications.
- ◆ Tier 3—The Web Servers and Documange Workstation (or client). These interact with the DBMS via the Server in Tier 2.

Architectural Advantages

Documange's architecture keeps the code that directly accesses the data base management system (DBMS) on trusted Server machines. It also reduces the number of DBMS connections required by making them proportional to the number of active Documange Servers connecting to the DBMS, rather than making them proportional to the number of users.

Database Access

Documange works with any ANSI standard SQL 92 DBMS that can connect to the its Server via an ODBC connection. Some Documange modules access the DBMS using ODBC, while for performance and architectural reasons others access the DBMS directly.

- ◆ **Server**—The Documange Server accesses the DBMS using the Open Database Connectivity (ODBC) standard.
- ◆ **Administrator**—During installation and configuration, the Documange Administrator Program interacts directly with the DBMS through ODBC.
- ◆ **Workflow Designer**—The Workflow Designer accesses the DBMS through ODBC.
- ◆ **Docusave/390**—Docusave/390 connects directly to the DBMS.

Database Connections

The Documanager Server establishes two or more ODBC connections to the database in which the housekeeping tables reside—DMANAGE on most installations—and at one or more additional connections for each line-of-business table within Documanager (this number is configurable). When the Documanager Administrator refreshes the Documanager Server, it creates a new instance of the Documanager Server. Each new instance contains an additional set of connections to the databases. When all of the Workstations (clients) connected to a Server instance log off, the Server releases that instance's connections to the databases.

The Server opens these connections *as needed* basis: as the Server needs more concurrent connections to a database, it opens more connections up to the configured limit. The Server reuses the same connection as much as possible, only opening and using additional connections when needed.

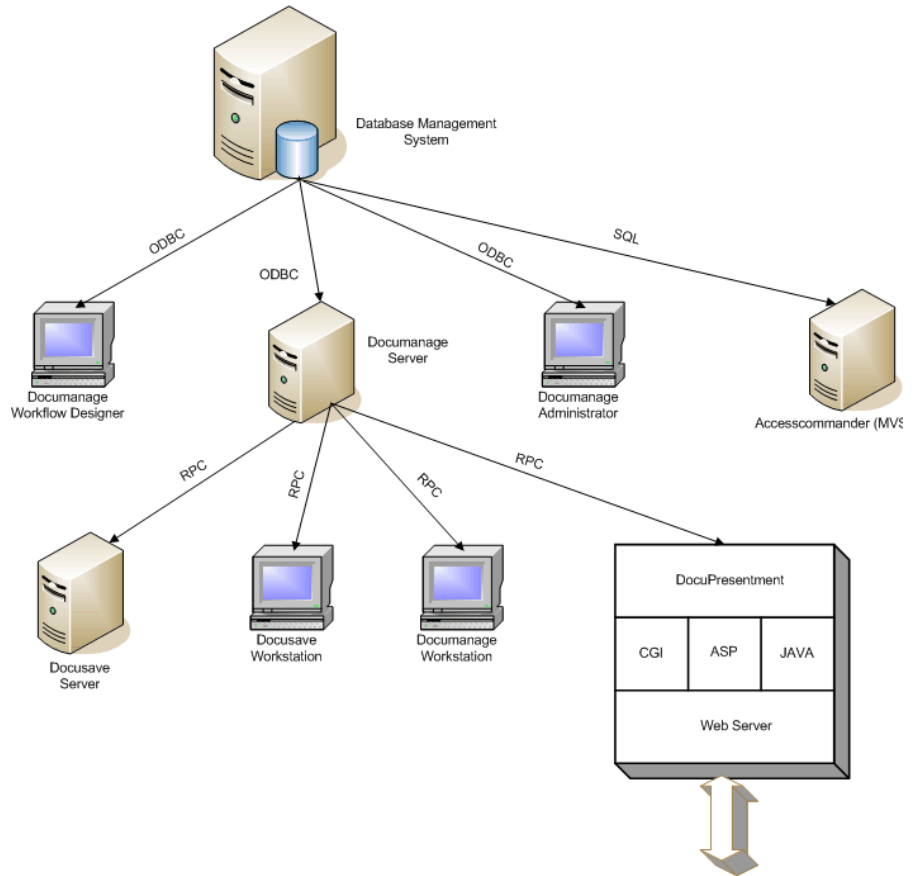
If the Server is configured to log all messages (LogDetailLevel 2 in the poffice.ini file), it logs the configured number of connections whenever the Server starts up or is refreshed, and the actual number of connections used whenever the Server shuts down or is refreshed.

The Documanager Administrator and Workflow Designer establish two ODBC connections to the database in which the housekeeping tables reside (DMANAGE on most installations), and one additional connection for each additional DSN you use (for line-of-business data sources) within Documanager.

Documange Architecture

Database Connections

The three-tier Documange architecture is shown here.



Database Tables

In this chapter

The information in this chapter will help you answer the following questions:

- What database tables does Documanager use and what does it use them for?
- How can you name these tables?

Introduction

Documanager uses database tables to keep track of documents, folders, cabinets and workflows. It uses its own housekeeping tables to catalog documents in the customer's DBMS and keep track of their status. It also adds columns to existing line of business tables. Tags in these columns allow Documanager to use rows in the tables to define virtual folders. Documanager Workstations move documents between these folders and the virtual cabinets that hold them, so that users can organize the documents and route them through workflows. Except for housekeeping tables, Documanager lets you choose your own table names.

Database Tables

Housekeeping Tables and Line of Business Tables

Housekeeping Tables and Line of Business Tables

Documanager uses housekeeping tables and PowerMapped Line of Business (LOB) tables to keep track of documents. It uses its own Housekeeping tables to catalog documents in the DBMS and keep track of their status, and it uses the client's PowerMapped Line of Business (LOB) tables to define virtual folders that contain groups of documents.

Documanager uses its Housekeeping tables to catalog documents and keep track of their status. These tables hold the data that Documanager uses to maintain the document repository, define its cabinets, manage documents and workflows, and access customer line-of-business (LOB) tables. Documanager version 6.4 uses fifty tables to track the state of the Documanager system.

PowerMapping—discussed in “PowerMapping” on page 8—allows Documanager to reference information in database tables not under its control. This "document enables" database applications that are independent of Documanager.

Housekeeping Tables

Documanager requires the creation and limited pre-loading of multiple tables so documents can be catalogued and tracked by status. Documanager uses Housekeeping tables for this purpose. Housekeeping tables contain "private data" used by Documanager to maintain the document repository, define cabinets, manage documents and workflow, and control access to the customer line-of-business (LOB) tables. Documanager version 6.4 uses 50 tables to track the state of the Documanager system.

Appendix A describes how to create the Documanager housekeeping tables.

CAUTION: Documange performance depends on database performance. While Primary Keys protect relational integrity, indexes are the normal mechanism to ensure high performance. For indexes to remain effective they may require periodic maintenance. It is expected that DBAs will monitor the indexes on Documange tables, and reorganize them as necessary. In particular adding large numbers of new documents has fragmented the index on the DocID field of OT_Docs table, at least on some database systems. Periodic REORG is recommended to maintain a high level of document ingestion performance.

OT_Docs Housekeeping Table

The OT_Docs table is the most important Documange housekeeping table. Each of its rows contains data on a document tracked by the Documange system. Other tables with the OT_ prefix track different document classes, storage areas, aliases, and Documange cabinets. A Documange cabinet provides a document-enabled view of a record in the line-of-business tables.

Accesscommander /390 Housekeeping Tables

Accesscommander /390 is a Server application that runs on an IBM System/390. Like other Client applications, it accesses the DBMS via a Server. It accesses the LOB tables to get the POWER_Tag field, and can insert records into the LOB tables. DMG is the default owner of the housekeeping tables, but can be configured in the INI files.

The Housekeeping tables read by Docusave/390 include:

Database Tables

PowerMapping

OT_CLASSVOLUMES

OT_INDEXES

OT_POWERCABINDEXES

OT_TABLES

OT_VOLUMES

The Housekeeping tables updated or inserted into by Docusave/390 are:

OT_DOCS

OT_LASTDOCID

OT_LASTRECORDID

OT_LASTTRENDTIONID

PowerMapping

PowerMapping—a unique feature of Documanage—turns the line-of-business database a model for the Documanage system, without constraining or compromising the LOB table(s). Typically, every row in an externally defined LOB table becomes a virtual folder that can contain documents in Documanage. PowerMapping thus adopts an externally defined Line of Business (LOB) table or set of tables as a basic filing structure that Documanage uses to organize document storage.

One of the most powerful aspects of PowerMapping is that it supports databases developed long before Documanage was considered or implemented as an application solution. These databases usually continue to be independently maintained outside of Documanage.

If a suitable pre-existing external table is unavailable for PowerMapping, LOB tables may be constructed as part of the Documanage installation to provide folder attributes within a Documanage cabinet.

When you PowerMap a table, Documanage modifies the LOB table, creates new housekeeping tables within that table's database (if it is the first table mapped from that database), and stores information about the mapped table in certain Documanage housekeeping tables.

Power Mapping:

- ◆ Adds a new column called POWER_Tag to the PowerMapped LOB table
- ◆ Reads the table definition and stores it in the OT_Tables and OT_Indexes tables
- ◆ Reads foreign keys for the table and stores them in the OT_Relationships tables
- ◆ Enters a record into the OT_Authority table granting default rights to Users group
- ◆ Adds a new entry in OT_DatabaseClass and creates four new tables (OT_Docs, OT_InBox, OT_Projects, OT_ActLog) in the LOB database if this is the first table in this database to be mapped.

Database Tables

Database Names

NOTE: The fully qualified names for these four tables are stored in the OT_DatabaseClass table. All queries to these tables use the qualified name read from OT_DatabaseClass.

Database Names

Documanager uses the following database naming guidelines:

- ◆ Documanager housekeeping tables are named during installation and the table names and column names are established by Documanager. These names should not be changed. Many of the tables also create one or more indexes which may be changed at will.
- ◆ Some DBMSs use the AuthorityID—or TableOwner—to define a name space for tables. That is, the TableName is unique within the scope qualified by the AuthorityID. In other systems, the TableName is unique with the scope of the Database qualifier. Documanager functions in either context.
- ◆ Documanager uses "DMANAGE" as the default housekeeping database name.
- ◆ Documanager does not require any particular database name for the housekeeping or LOB tables.
- ◆ Documanager does not impose naming requirements or use restrictions on tablespaces.

NOTE: Tablespaces can be effective tools for grouping tables together that have shared physical storage and other characteristics. Tables can reside in a shared Tablespace or in separate Tablespaces. Groupings are suggested in the setup scripts supplied with Documanager.

- ◆ Documanager does not impose naming restrictions on Line of Business (LOB) tables.
- ◆ Starting with version 5.2, Documanager only uses explicit table owner qualifiers. The table owner uses DMG as a default value.
- ◆ Documanager versions after 5.2 and allow the database name and owner to be specified as system parameters, and consistently uses qualified names (DMG.tablename, for example). DMANAGE is the default database name and the DMG owner name can be changed at installation time.

Database Tables

Database Names

Documanager Administrator

The Documanager Administrator is the user interface for the Documanager database administration functions. These functions are available through buttons on the main Administrator dialog that lead to dialogs used to perform the following functions: PowerMapping, adding new Line of Business (LOB) tables, describing relationships among tables, establishing cabinets, setting up system defaults, configuring Data Source Names (DSNs), establishing storage volumes, defining document types and associating them with storage volumes, listing users and groups, and assigning permissions to groups.

All of these functions can be performed manually, but this is not recommended. Of all of the Administrator functions, only PowerMapping and adding LOB tables affect the structures of LOB databases by creating, dropping, or altering database objects. PowerMapping and defining business tables may therefore be subject to installation security or operational policies that prevent a Documanager administrator from performing these tasks. PowerMapping and its effects on the databases are described in “PowerMapping” on page 8.

In some cases, you may find it necessary to add LOB tables and perform PowerMapping manually. If you wish to do this, request more information from Skywire’s Documanager for Oracle Customer Support.

Relationships

The Relationships dialog adds, modifies, or deletes rows in the OT_Relationships table.

Cabinet Definition Dialog

The Cabinet Definition dialog adds, deletes, or modifies rows in the OT_PowerCabGroups, OT_PowerCabVars, OT_PowerCabLevels, OT_PowerCabConsd, OT_PowerCabIndexes, and OT_PowerCabinets tables.

System Dialog

The System dialog adds, deletes or modifies rows in the OT_Profiles table.

Configure DSNs Dialog

The Configure DSNs dialog has no effect on any tables. It writes the encrypted database password and user id to the poffice.ini file.

Storage Dialog

The Storage dialog adds, modifies, or deletes rows in the OT_Volumes and OT_SysAttrs table.

Categories Dialog

The Categories dialog adds or deletes rows in the OT_ClassPickLists, OT_ClassVolumes, and OT_ObjectClasses tables. For Extended Document Properties a table is created based on the Category name OT_<category name> in the housekeeping database.

Users Dialog

If the access model for the “Users” dialog is “NTLM LAN Mgr” then the dialog does not affect any table. It gets its information from the domain controller. The functionality of this dialog can be accomplished through standard Windows NT administration tools.

If the access model for the “Users” dialog is “Database” then the dialog allows you to add, modify or delete users from the OT_Users table.

The User Groups dialog adds, deletes or modifies rows in the OT_Authorities table. The database version of the dialog also allows you to add, delete or modify groups.

Authorities Dialog

The Authorities dialog adds, deletes or modifies rows in the OT_Authorities table.

Locks Dialog

The Locks dialog adds, deletes or modifies rows in the OT_AnnotationLocks table.

Structure of Documanage Database Objects

In general, housekeeping tables are stored in a single dedicated database so that they can be managed (started, stopped, backed up, and restored, for example) as a single entity. This is convenient but not necessary.

Tables and Data Bases

A PowerMapped LOB table can be included in either the same database as the housekeeping tables, or defined elsewhere. If LOB tables predate the installation and were defined independently of Documanage, the tables are most likely to be in separate databases. However, the LOB tables can be placed in the same database as the housekeeping tables.

If the LOB tables are created for use by Documanage they can be conveniently created in the same database as the Documanage housekeeping tables, or any other data source.

The MVS DB2 setup scripts create several tablespaces and divide the tables among them according to either their physical attributes or operational requirements. OT_Authority and OT_AutoMailTask require a large tablespace (with a large buffer pool) due to wide VARCHAR column sizes. While these tables rarely contain very large records, the tablespace must be able to contain a maximum sized record. Testing at Skywire's Documanage for Oracle is generally conducted with these tables isolated in a 32 KB tablespace.

Row-level Locking

Row-level locking is generally expected in all Documanage operations which update database records. It is not specified as a requirement for all tables simply because many tables are rarely, if ever, updated by the Server and installations run quite successfully without it when the tables are partitioned as described in this section. OT_DOCS requires that row-level-locking be supported in its tablespaces. Certain operations, such as moving documents between folders, impose this requirement.

Normal Table Space

The remainder of the tables are placed together by the DB2 script in a "normal" tablespace with an average (default) buffer pool. This is not a requirement and customer database administrators may well have a better understanding of how their installation may stress certain tables and they may reconfigure locations for performance reasons, given the information in the remainder of this section.

Documanage core tables may be categorized according to how they are used and updated by the product. This may affect installation decisions such as when and how to back up tables and how to group them either in storage or logically into tablespaces. More details on this topic are provided in the section on "Table Use, Indexing, and Key Recommendations" on page 27".

Document Related Tables

"Document related" tables largely contain either "per document" or "per document revision" information. They are updated as documents are added, revised, and deleted from the Documanage system. The storage requirements for Document related tables tend to be proportional to the quantity of documents stored. Furthermore, these tables tend to be accessed to create a document search result list, retrieve a document, and/or display its attributes. Some of these tables are related to specific functions or product features that a specific installation may or may not be using. In this case, the impact that the table has on database performance and memory space varies accordingly.

Document related tables are:

OT_DIARIES;

OT_DOCS;

OT_IDRANGES;

OT_INDEXES;

OT_LASTDOCID;

OT_LASTRECORDID;

OT_LASTRENDITIONID;

OT_NOTES;

PERSONALCABINET;

Structure of Documanage Database Objects

Workflow Related Tables

Workflow Related Tables

"Workflow related" tables contain rows of information related to active and past workflow projects in the system. If the installation is not using Documanage workflow features, these tables will be minimal or empty. If in use, these tables contain records which are created by starting workflow projects and updated by completing workflow tasks and advancing the workflow through the task map.

OT_ACTLOG;

OT_INBOX;

OT_PROJECTS;

OT_PROJECTSETTINGS

Administrative Tables

Administrative tables contain information that may vary slowly over time and are updated primarily when administrative changes are made to the system, such as mapping a new database, creating a new cabinet, designing a new workflow task map, or changing the storage volume or document type list. Typically, each Documanage Server instance caches a local copy of the contents of Administrative Tables at start up.

This table data (in memory) tends to be accessed as searches are performed within the system and folder result lists are created. Much search activity will access these tables before requiring access to the document related tables. These may change rarely or frequently depending on the administrative

activity ongoing in the system. Because the data from these tables is cached in the Servers, Server instances must be "refreshed" by the Administrator when the database tables are changed.

The Administrative tables are:

OT_ANNOTATIONLOCKS

OT_AUTHORITY;

OT_CLASSPICKLISTS;

OT_CLASSVOLUMES;

OT_DATABASECLASS;

OT_GROUPS;

OT_GROUPSGROUPS;

OT_OBJECTCLASSES;

OT_LASTTASKID

OT_POWERCABCONSD;

OT_POWERCABGROUPS;

OT_POWERCABINDEXES;

OT_POWERCABINETS;

OT_POWERCABLEVELS;

Structure of Documanage Database Objects

Static Tables

OT_POWERCABVARS;

OT_PROFILE;

OT_RELATIONSHIPS;

OT_SYSATTRS

OT_TABLES;

OT_TASKLINKS

OT_TASKS

OT_TASKTEAMGROUPS;

OT_<category name> (If Extended Document Properties are used)

Static Tables

"Static" tables tend to change rarely or not at all. They are tables containing parametric or code information that are established at installation or initial set up time and do not routinely change as a result of either user or administrative activity. Static tables are:

OT_LAYERTYPES;

OT_ODBCDATATYPES;

OT_SCREENKEYS;

Structure of Documanage Database Objects

Static Tables

OT_SCREENS;

OT_SESSIONS;

OT_TASKTYPES;

Structure of Documanage Database Objects

Static Tables

Documange Authorities

The Documange components of Administrator, Server and Workflow designer connect to the databases with one or more user ID's that must have certain authorities in the database management system. Both the Administrator and Server must be able to add, modify, and delete records in the housekeeping tables. Table creation and table modification routinely occur only in the Administrator application. The Workflow Designer may create certain tables in older installations in which the tables do not yet exist, but these tables will already be present in a new installation so no tables will be created by this program. Updater creates, modifies, and drops tables and columns.

Documange then uses its own authorities scheme to limit actions that can be performed by clients accessing the system through the Server. For example, a user may or may not have permission to update rows in the LOB table or annotate a document. These permissions are assigned through the Administrator module.

The Administrator account (authID) must have full access to the tables so that it can map tables, modify Documange-specific permissions, specify storage volumes, etc.

Some of the database activities performed by Documange include:

- ◆ The Administrator module creates housekeeping tables and inserts records during PowerMapping, and modifies the structure of LOB tables to add the Power_Tag column to each table. This requires modification authority to the system tables storing table structures. Once Power Mapping is complete, this authority may be reduced. (LOB tables are generally created outside the scope of Documange - even if created primarily (or solely) for use by Documange.)

Documange Authorities

- ◆ The Workflow Designer creates tables (only in older installations in which these tables do not yet exist) and inserts and deletes records.
- ◆ During normal Documange operation, no tables are created, but records in some housekeeping tables are routinely inserted and updated by Documange Servers.
- ◆ During regular operation, the LOB table may be updated. The POWER_Tag column is modified, and users who have corresponding permissions may insert, modify and delete records in the LOB table. Separate Documange authorities control these permissions through the Documange Servers.
- ◆ Docusave Server running on OS/390 is considered a server application and performs the same operations as Documange Server when loading new documents and creating folders (LOB table rows).

Table Use, Indexing, and Key Recommendations

Summary

Very few of the Documange database tables are used frequently during operation of the Server. Most of the operations involve the Line of Business tables, which the Workstation represents using Folders. In many cases, the tables hold static configuration information about the system that is only updated by the Documange administrator using the Administrator and the Workflow Designer. Such information is not usually changed during operation. Certain tables remain empty if Workflow and users and groups based on the Documange database are not used.

Tables Actively Used During Normal Operation (10)

OT_ANNOTATIONLOCKS	OT_AUTHORITY	OT_DOCS
OT_LASTDOCID	OT_LASTRECORDID	OT_LASTTRENDITIONID
OT_TXLOG	PERSONALCABINET	OT_NOTES
OT_DIARIES	OT_IDRANGES	

Tables Used During Start-up and Administration (17)

OT_CLASSPICKLISTS	OT_CLASSVOLUMES	OT_INDEXES
OT_OBJECTCLASSES	OT_DATABASECLASS	OT_INDEXES
OT_POWERCABCONSD	OT_POWERCABGROUPS	OT_POWERCABINDEXES
OT_POWERCABINETS	OT_POWERCABLEVELS	OT_POWERCABVARS
OT_PROFILE	OT_RELATIONSHIPS	OT_TABLES
OT_VOLUMES	OT_SYSATTRS	OT_EXCLUDECATEGORIES

Static Tables of Codes and Strings Read at Startup (4)

OT_LAYERTYPES	OT_ODBCDATATYPES	OT_TASKTYPES
OT_TXLOGCODES		

Tables Used Only by Workflow (16)

OT_ACTLOG	OT_AUTOMAILTASK	OT_INBOX
OT_LASTTASKID	OT_LAUNCHERINFO	OT_POLLERINFO
OT_POLLERTASK	OT_PROJECTS	OT_PROJECTSETTINGS
OT_TASKLINKS	OT_TASKS	OT_TASKTEAMGROUPS
OT_WORKAUTHORITY	OT_WORKFLOWS	OT_CONTAINERTEAMS
OT_TASKCONTAINERS		

Tables Used Only by HostLink (3)

OT_SCREENKEYS

OT_SCREENSHOTS

OT_SESSIONS

Tables Used Only if Database Authentication/Authorization is Used (5)

OT_GROUPAUTHS

OT_GROUPS

OT_GROUPSGROUPS

OT_USERS

OT_USERSGROUPS

Group Level Authorities Tables (1)

OT_GROUPAUTHS

Synchronized Task Tables (1)

OT_SERVER_TASKS

Tables Used for Document Retention Management(4)

OT_DOCS

OT_SERVER_TASKS

OT_OBJECTCLASSES

OT_SYS_ADDR

Table Descriptions and Uses

The table that follows alphabetically lists each Documanage table, a description of the table, and how it is used.

Table	Description	Use
OT_ACTLOG	One record per step in a workflow.	Used by the Server to get workflow activity log
OT_ANNOTATIONLOCKS	One row for each Annotation Lock, deleted after the Lock is released.	Protects against simultaneous annotation by different users.
OT_AUTHORITY	One row per table per Documanage authority group.	Used by the Server to determine the access rights.
OT_AUTOMAILTASK	One row for each workflow automail task.	Inserted by Workflow Designer.
OT_CLASSPICKLISTS	Two rows per defined document type.	Rows inserted by the administrator for sub-types, flags, labels and status codes. Read by the Server at startup
OT_CLASSVOLUMES	Up to two records per Document Type.	Read at Server startup only.

Table Use, Indexing, and Key Recommendations

Table Descriptions and Uses

Table	Description	Use
OT_CONTAINERTEAMS	One row per group in each workflow task container.	Rows inserted by the Workflow designer and read by the Server to get information about the task containers and the groups to which they belong.
OT_DATABASECLASS	One row per mapped database plus one for this (housekeeping) database.	Used by the Server to determine the database connections required and to get Docs, Inbox, Actlog and Project tables.
OT_DIARIES	One row for each diary entry.	Rows are added by the Server as each diary entry is added using the client application.
OT_DOCS	One record per document version.	HIGH USE; Searched by queries, RecordID. Also searched by DocID, SourceDocID, Flags, ContentModifiedOn.
OT_EXCLUDECATEGORIES	One row for each category excluded from each cabinet.	Any document category/cabinet combination NOT in this table will be considered valid.
OT_GROUPAUTHS	This is the master table of group based authorities. These are authorities that are not tied to group/document table pairs.	Used by Documanager to determine a user's permissions based solely on group membership during authorization.
OT_GROUPS	This is the master table of groups managed by the database plugin within the Documanager system.	Used by Documanager to determine a user's permissions during authorization.
OT_GROUPSGROUPS	This table manages the groups within groups relationships. Note, if a member belongs to group 'A' and group 'A' belongs to group 'B', then the member also belongs to group 'B'.	Used by the Documanager authorization plugin to determine the complete set of groups to which a user belongs.

Table Use, Indexing, and Key Recommendations

Table Descriptions and Uses

Table	Description	Use
OT_IDRANGES	This table retains DocID ranges and RecordID ranges at Server shutdown time, so that those ranges can be used by Servers at startup time. One row for each unused range of ID values at Server shutdown.	When a document is input or a folder is created, the Server uses an available range from this table. Use is HIGH at startup and shutdown. ID ranges not completely exhausted at Server shutdown are written here, one range per row. At Server startup, these rows are read and deleted by Servers.
OT_INBOX	Workflow tasks on a per user basis.	Only for workflow.
OT_INDEXES	Mapped table indices (columns, datatype, etc.) One row per database column per mapped table.	Read by the Server at startup to determine the indices of the mapped database tables.
OT_LASTDOCID	Exactly one row.	Source for next new range of DOCID values. Updated at document input, check out, check in major/minor version, and create shortcut, when the Server has exhausted its current range of DOCID values and no such range is available in table OT_IDranges.
OT_LASTRECORDID	Exactly one row.	.HIGH during import to new Folders. Source for next sequential Power_Tag. Updated for each occasion of the first document added to an empty Folder and when a Folder is added through Documanage.
OT_LASTRENDITIONID	Exactly one row.	Source for next sequential rendition ID. Updated for each new rendition set.

Table Use, Indexing, and Key Recommendations

Table Descriptions and Uses

Table	Description	Use
OT_LASTTASKID	Exactly one row.	Source for next sequential workflow TaskID. Updated for each new initiated workflow task.
OT_LAUNCHERINFO	One row for each defined workflow Launcher task.	Rows added by workflow designer, read by the Server at startup to get information on the launcher task.
OT_LAYERTYPES	Exactly two rows.	Setup by installation; never updated.
OT_NOTES	One row for each note of a Folder or project	Rows are added by the Server for each note added to a Folder or project, using the Client application.
OT_OBJECTCLASSES	One record per Document Category.	Rows inserted by the administrator per doc-category. Read by the Server at startup time.
OT_ODBCDATATYPES	Six rows of static code and description data loaded by setup script.	Read in at start of each Server. Never updated.
OT_POLLERINFO	One row for each defined workflow Poller task.	Used by the Server to get next Polling information, in workflow.
OT_POLLERTASK	One row for each defined workflow Poller task.	Used by the Server to get Poller task information.
OT_POWERCABCONSD	One row per cabinet, four per workflow, plus one row for each consolidation.	Rows inserted by the administrator and read by the Server at startup.
OT_POWERCABGROUPS	Groups which can access a particular cabinet. One row per group per cabinet (normally empty).	Rows inserted by the administrator, determines users belonging to which group have access to a particular cabinet. Read by the Server at startup.

Table Use, Indexing, and Key Recommendations

Table Descriptions and Uses

Table	Description	Use
OT_POWERCABINDEXES	Same as OT_INDEXES but defined per cabinet. One row per column in tables mapped into PowerCabinets.	Rows inserted by the administrator. Server reads this table at startup to get all the indices(column information) of the mapped tables that form a cabinet.
OT_POWERCABINETS	One row per cabinet, 4 per workflow.	Rows are inserted by the administrator, read by the Server to get the list of all Cabinets.
OT_POWERCABLEVELS	One row per table mapped into a Power Cabinet.	Rows are inserted by the administrator. Server reads them at startup to get the cabinet level information for a cabinet.
OT_POWERCABVARS	Three rows per workflow cabinet. One row per defined variable in cabinet filter.	Rows are inserted by the administrator. Server read at startup to get variable filter on a cabinet.
OT_PROFILE	Holds system settings in a single row. Exactly one row.	Updated by Administrator. Server reads at startup to get information about transaction logging, timeout (Server and client).
OT_PROJECTS	All the workflow projects on the system	The Server adds and deletes rows as projects are added to the system and completed by the users .
OT_PROJECTSETTINGS	One row per workflow.	Updated only by Workflow Designer.
OT_RELATIONSHIPS	One row for each relationship between two mapped tables defined by Administrator.	Rows are inserted by the administrator. Server read at startup and gets relationships between tables that form a cabinet (Multilevel and consolidated).
OT_SCREENKEYS	HostLink	HostLink only.
OT_SCREEN	HostLink	HostLink only.

Table Use, Indexing, and Key Recommendations

Table Descriptions and Uses

Table	Description	Use
OT_SERVER_TASKS	One row for each synchronized task. Each row has the following entries: the name of the task, the ID of the Server either currently working on the task or the Server that last worked on the task, a timestamp for the last status update, and the last status code (Running, Failed, Completed).	The OT_ServerTasks table holds the data required to synchronize tasks among Documanage Server instances that are sharing databases. It synchronizes the retention processing task among Servers.
OT_SESSIONS	HostLink	HostLink only.
OT_SYSATTRS	Ten rows, based on settings in Storage and other system settings.	Schema version, Storage Method, Time zone origin, Content Management Options.
OT_TABLES	One row for each mapped table.	Administrator inserts rows and Server reads them at start to get list of mapped tables.
OT_TASKCONTAINERS	One row per task container in all workflows.	Rows inserted by workflow designer and read by both Server and Workflow Designer to get information on a task container.
OT_TASKLINKS	One row for each link in each workflow.	Rows added by workflow designer and read by the Server to get the next task in the workflow map.
OT_TASKS	One row per node in all workflows.	Rows inserted by workflow designer and read by both Server and workflow designer to get information on a task.
OT_TASKTEAMGROUPS	One row per group in each workflow node.	Rows inserted by the Workflow designer and read by the Server to get information about the tasks and the groups to which they belong.

Table Use, Indexing, and Key Recommendations

Table Descriptions and Uses

Table	Description	Use
OT_TASKTYPES	19 rows of static code and description data loaded by setup script.	Read in at start of each Server. Never updated.
OT_TXLOG	Server-side event log records. No need for keys or indices.	HIGH on active Servers when logging turned on. Write-only: records are appended. Any search/read is on an ad hoc report generation basis. There is no Primary Key for this table because duplicate records are allowed.
OT_TXLOGCODES	22 rows of static code and description data loaded by setup script.	Read in at start of each Server. Never updated.
OT_USERS	This is the master table of users managed by the database plugin within the Documanage system.	Used by the Documanage Server to authenticate users on login.
OT_USERSGROUPS	Manages the direct membership of user groups. Users can also be indirect members of groups via the OT_GroupsGroups table that we will discuss shortly.	Used by Documanage to determine the complete set of permissions for a user during authorization.
OT_VOLUMES	One row per storage volume.	Rows inserted by the administrator read by the Server at startup to get the list of all available volumes.
OT_WORKAUTHORITY	Exactly one row per group defined in each workflow.	Rows inserted by the workflow designer, Server reads at startup to get information about which groups have access to which workflows.
OT_WORKFLOWS	One row for each workflow.	Rows inserted by the workflow designer, Server read at startup to get the list of workflows on the system.

Table Use, Indexing, and Key Recommendations

Table Descriptions and Uses

Table	Description	Use
PERSONALCABINET	Two rows per user who has once opened their Personal Cabinet.	Rows are added by the Server, on a per user basis as each user opens their Personal Cabinet for the first time using the Client or Active Client application.

Using Keys and Indices

The paragraphs that follow discuss how keys and indices can be used in housekeeping tables and PowerMapped Line of Business (LOB) tables

Housekeeping tables

Indexes can be beneficial for large housekeeping tables with multiple rows (the OT_Docs table, for example) but are not necessary for small tables with few rows except to support keys. All tables can be indexed by site convention, if required.

When indexing any tables, search patterns must be kept in mind. For example, if the TAG column (effectively the Document Name) represents information that is suitable for indexed searching (a Policy Identifier, for example) then this may be a good candidate for a database index. Since the suitability of these columns for indexing depends on the individual setup, a database administrator should consider what indices would be beneficial based on how the Documange system will be used.

Designated keys document the structure and protect the integrity of most housekeeping tables. The current versions of the database setup scripts include these keys and create indexes. Information on individual tables is shown in the previous section, with comments on keys and on the numbers of rows that can be expected in certain tables.

PowerMapped Line of Business tables

The designers of Line of Business (LOB) tables generally choose indexes and keys for the tables. Since the design of the LOB table may predate the installation of Documange, and since the LOB tables may serve other

business functions in addition to the ones they serve for Documanage, the use of keys and indexes for the LOB tables are not constrained by Documanage. However, because the Power_Tag column is such an important field to Documanage, creating an additional index for this column is highly recommended. The fields used as part of the Key in Documanage will also benefit from being indexed.

NOTE: You should constrain line of business tables to prevent records from having duplicate Folder keys in Documanage. An easy way to do this is to ensure that all of the columns that are in the table's primary key are also used as part of the Folder key.

If a table is not constrained in this way, and records with duplicate Folder keys are entered into the table, Documanage will not be able to tell the difference between the two Folders causing it to have unexpected behaviors. Documanage keeps you from taking certain actions that could corrupt data in these circumstances. Use your database tools to prevent duplicate Folder keys so that Documanage can function fully and effectively.

Table Use, Indexing, and Key Recommendations
Using Keys and Indices

Changing Reserved Names from Version 5.2 and Earlier

Documange versions 5.2 and later allow administrators to control the authentication ID and table owner names used by Documange. Releases of Documange before version 5.2 require a table owner name of EZPOWER, or in the case of DB2 MVS, require additional configuration of site ALIAS and user SYNONYM entries. Refer to “Appendix D: Database Naming Conventions Before Ver. 5.2” on page 73.

Documange versions 5.2 and later, specify the database name and owner as installation parameters in the POFFICE.INI file and consistently use qualified names (DMG.tablename, for example). DMANAGE is the default database name and the DMG owner name can be changed at installation time. The DSN and the owner name must be set in the POFFICE.INI file on the Router machine. The table owner is configured as an installation parameter in the [EZPDSInfo] section of the POFFICE.INI file:

```
[ EZPDSInfo ]  
    DataSource=DMANAGE  
    AuthID=DMG
```

These POFFICE.INI file options are described in more detail in the Documange Administrator’s Guide.

In addition, the contents of certain database fields must be changed to reflect the new DSN and owner name. OT_DatabaseClass and OT_Tables contain references to tables with the EZPOWER DSN.

Changing Reserved Names from Version 5.2 and Earlier

OT_DatabaseClass.DataSource and OT_Tables.OT_Database must be changed from EZPOWER to the name of the actual data source.

There are also a number of references to EZPOWER in other fields and tables. These include:

- ◆ OT_DatabaseClass.DocsTbl,
- ◆ OT_DatabaseClass.InBoxTbl,
- ◆ OT_DatabaseClass.ActLogTbl,
- ◆ OT_DatabaseClass.ProjectsTbl,
- ◆ OT_Tables.LabelCalc,
- ◆ OT_Tables.Qualified,
- ◆ OT_PowerCabLevels.LabelFormula,
- ◆ and OT_PollerInfo.DataSource.

These tables may be edited using the scripts provided in the Appendices.

Documanager Backup and Restoration

Housekeeping tables must be treated as a logical unit. Ideally, backups should be taken while the housekeeping tables are not changing. See the tablespaces for logical groupings.

Similarly, the storage system elements should be backed up (and restored) in strict coordination with the backup of the housekeeping tables.

Do not restore any elements (housekeeping or LOB tables) without coordinated maintenance to the other elements.

Documanage Backup and Restoration

Appendix A: Database and Table Setup Scripts

The Documanage Database Wizard application contains scripts to create necessary Documanage housekeeping tables for common database management products. For DB2 MVS, this setup is commonly handled by a Database Administrator using management tools which are standards for that installation. Documanage comes with a directory containing DB2 compatible MVS scripts that the administrator can use to set up the housekeeping tables.

MVS Scripts

MVS-compatible DB2 scripts are provided in the distribution media as editable text files. These files may be edited with installation-specific naming conventions and parameters and then uploaded to the MVS system and run under facilities such as SPUFI to create the necessary database objects.

The setup functions are divided into four portions: (1) Database and tablespace setup, (2) housekeeping table setup with indices, (3) initialization of housekeeping table data and, (4) updating previously installed tables to the latest Documanage housekeeping table schema.

You can find the scripts on your Documanage CD under the “MVS Scripts” directory. Each script builds on the previous steps. The scripts should be edited to provide substitutions for case-sensitive parameter names embedded throughout the scripts. These names are described in commentary at the top of each script file. For example, from the first script:

//

Appendix A: Database and Table Setup Scripts

MVS Scripts

```
// dbName - database name selected
// BPn   - regular bufferpool to use
//       To list available: Command -DIS BPOOL
// BP32K - large record bufferpool to use for tables with long records
// sname - storage group name selected
//       To list available: SELECT * FROM SYSIBM.SYSSTOGROUP;
// tbTSnrm1 - normal (smaller) tablespace with normal bufferpool
// tbTS32K - tablespace for larger tables requiring a 32K bufferpool
// tbTSRLL - normal (smaller) tablespace with row-level locking
```

The script references these strings where specific names should be substituted into the scripts. For example, with a case-sensitive editor, replacing the string "dbName" with "DMANAGE", "BPn" with "BP0" and "sname" with "DMGSTGRP" would change the following raw script line:

```
CREATE DATABASE dbName BUFFERPOOL BPn STOGROUP sname;
```

into the following desired script line ready to execute:

```
CREATE DATABASE DMANAGE BUFFERPOOL BP0 STOGROUP
DMGSTGRP;
```

All of the scripts are then subjected to the same series of find/replace operations and the scripts are prepared to be executed.

The MVS Scripts directory

MVS Scripts Directory contains files that are prototypical MVS DB2 tested scripts for setting up a Documange housekeeping database. This files are written with certain names intended for substitution by editing the scripts prior to using them. The contents of this directory are listed as follows:

NOTE: All files **MUST** be edited as needed for installation conventions and to supply actual names.

DB2InsertsMVS.txt

This script inserts default and initial values into some of the tables.

DB2GrantsMVS.txt

This script can be used to enable use of another logon account if the user account used by the Documange Server to logon to the DB2 database on MVS via the ODBC connection is not the same as the specified owner authID for the tables. This script grants all privileges on all Documange tables to the user account to be used to logon to the database via ODBC. This is not required if the odbcUserID is the same as the specified authID for the tables.

DB2DatabaseMVS.txt

This script creates the database and tablespaces. You can change it as needed for installation conventions.

DB2TablesMVS.txt

This script creates the tables and some indices. You can change it as needed for installation conventions.

Appendix A: Database and Table Setup Scripts

The MVS Scripts directory

DB2UpgradeMVS.txt

This file contains statements to update previously created tables and indices to conform to the latest version of the housekeeping table schema. Statements are grouped by schema version, and separated by comments. Use only the group of statements required to update your current schema version.

Database schema version history is discussed in Appendix B of this document.

ReadmeMVS.txt

Describes contents of the MVS Scripts directory.

Appendix B: Database Schema

The schema version of the Documanage database is a snapshot of the database structure and its semantics. The schema version—or database version —identifies the layout of the database: the tables and columns in the database and how they are used. When new tables or columns are added, removed or changed, a new schema version of the database is created.

Each version of the Documanage software is designed to work with a particular schema version. Parts of the Documanage code depend on a particular database schema because the code formulates SQL queries that made assumptions about the schema, making it necessary to associate the code with a particular schema versions.

It can be said that each version of the software targets a specific schema version. However, in some cases, a Documanage version will also work with schema versions other than the schema version that it targets.

Schema Detection

The Updater examines the structure of the existing database and updates the all its components to match a given schema version. For example, if the ContentModifiedOn column is missing from any OT_Docs table—added when updating to schema version 2—the Updater recognize this as a schema version 1 database, no matter what steps for updating to a later schema have been completed. The updater does not use the SchemaVersion system

Appendix B: Database Schema

SchemaVersion and MinAppSchema

attribute (refer to “SchemaVersion and MinAppSchema” on page 50) to determine the schema version.

SchemaVersion and MinAppSchema

Once the updater updates the schema version of the database, it is registered using the "SchemaVersion" and "MinAppSchema" system attributes in the OT_SysAttrs table. Applications other than the Updater determine the database schema version by examining the SchemaVersion attribute stored in the OT_SysAttrs table.

An application will work with the database as long as the target schema version is greater than or equal to the MinAppSchema, and the database SchemaVersion is greater than or equal to the target schema version.

Updating the Schema Version

The database Updater must be run each time that a new version of Documanager is installed, to update the schema version of the Documanager database. Running the Updater is analogous to running the Database Wizard during a new installation when a database has not been installed.

The updater detects all schema versions of the database, and can update any schema version to the current version except Schema Version 0 which indicates that the Documanager database has not been installed. Schema Version 0 is a special case which is handled by the Database Wizard. To update any previous schema version to the latest schema version, the updater

only has to update each successive schema version to the next until it reaches the latest version.

Database Schemas Defined

Schema versions are represented as positive integers; larger integers represent later Schema versions. The following paragraphs describe the schema versions in the order in which they were developed.

Schema Version 0

Schema Version 0 indicates that no Documanager system was detected. When a database schema is 0, the Documanager system needs to be installed and set up, either manually or by using the Database Wizard. The Updater will not update from schema version 0.

Detection

Version 0 is assumed when no other version is detected.

NOTE: The database format has changed several times prior to Documanager 5.1.5. Documanager does not detect or update from these earlier versions. The user should manually update their system to schema version 1 (Documanager 5.1.5) before running the Updater.

Appendix B: Database Schema

Database Schemas Defined

Schema Version 1

Schema version 1 was used from PowerOffice 5.0 through Documanage 5.1.5, and is the most common schema for currently installed systems prior to Documanage 5.2.

Database Changes

The database is schema version 1 if there is an OT_Docs table with a DocID column detected in the Documanage database, and no higher schema version is detected.

Schema Version 2

Database Schema 2 was released with Documanage 5.2.

Database Changes

Schema version 2 supports new document attributes and system attributes. To update to schema version 2 from schema version 1:

- 1 **Add new column "ContentModifiedOn" of type "datetime" to the OT_Docs tables in the housekeeping database and all LOB databases.**
- 2 **Add new column "LastAccessOn" of type "datetime" to the OT_Docs tables in the housekeeping database and all LOB databases.**
- 3 **Add new column "DocumentSize" of type "Integer" to the OT_Docs tables in the housekeeping database and all LOB databases.**

- 4 **Copy the value of column LastEditOn to columns ContentModifiedOn and LastAccessOn, and set DocumentSize to 0, for all records in all OT_Docs tables.**
- 5 **Add new table, OT_SysAttrs, to the housekeeping database. This table has two columns, "PropName" of type varchar(20 -not null) and "PropValue" of type varchar(255).**

Semantic Changes

In schema version 2, the SourceDocID column is used to record the "Document ID" of the Document to which a particular Version belongs. For the current version of a document, these IDs are the same. The OT_Docs records for historical versions of a document will have unique DocIDs, but will have their SourceDocIDs set to the DocID of the current version of that document. So, the SourceDocID should be:

- ◆ The same as its DocID, for the current version of a document
- ◆ The DocID of the current version of a document, for a historical version of a document
- ◆ The DocID of the current version of a checked-out document, for a checked-out copy of a document.
- ◆ The DocID of the current version of a document the shortcut links to, for a reference ("shortcut").

For historical OT_Docs records, SourceDocIDs remain unchanged (mostly 0's) until schema version 3.

Appendix B: Database Schema

Database Schemas Defined

Schema Version 3

Database Schema 3 was released with Documanage 6.0.

Database Changes

Schema version 3 supports extended document attributes and some new system attributes including a time zone origin for the database. To update to schema version 3 from schema version 2:

- 1 **Insert time zone origin properties TZOBiases, TZOSTandardDate, and TZODaylightDate into the OT_SysAttrs table in the housekeeping database.**

NOTE: The time zone information is extracted from the Windows configuration of the machine running the Updater.

- 2 **Insert the MinAppSchema property (PropValue="1") into the OT_SysAttrs table.**
- 3 **Insert the StorageMethod property (PropValue="Round Robin") into the OT_SysAttrs table.**
- 4 **Insert the StorageReserve property (PropValue="0") into the OT_SysAttrs table.**
- 5 **Add new column "XattribTable" of type varchar(32) to OT_ObjectClasses table in the housekeeping database.**
- 6 **Add new column "EnableXDA" of type smallint to OT_PowerCabinets table in the housekeeping database. Set EnableXDA=1 for existing records.**
- 7 **For each OT_Docs table, fix the SourceDocID according to the semantic changes from schema version 2.**

- 8 **Update the SchemaVersion property in OT_SysAttrs with PropValue="3".**

Schema Version 4

Database Schema 4 was released with Documange 6.1.

Database Changes

Schema version 4 supports content management, annotation locks, and long document types (now called "Categories"). To get from version 3 to version 4:

- 1 **Add new column "Released" of type smallint to OT_Docs tables in the housekeeping database and all LOB databases.**
- 2 **Add new column "Approved" of type smallint to OT_Docs tables in the housekeeping database and all LOB databases.**
- 3 **Add new column "Obsolete" of type smallint to OT_Docs tables in the housekeeping database and all LOB databases.**
- 4 **Change the type of the "Type" columns in OT_Docs tables in the housekeeping database and all LOB databases to varchar(32).**
- 5 **Insert the RelWithoutAppr property (PropValue="No") into the OT_SysAttrs table.**
- 6 **Add a new table OT_AnnotationLocks to the housekeeping database. This table has five columns: DocID (int not null), LockUsername (varchar(64) not null), LockHostName (varchar(64) not null), LockTime (varchar(64) not null), and ServerID (varchar(64) not null).**

Appendix B: Database Schema

Database Schemas Defined

- 7 **Update the SchemaVersion property in OT_SysAttrs with PropValue="4".**

Schema Version 5

Database Schema 5 was released with Documange 6.2.

Database Changes

Schema version 5 supports workflow suspension. To get from schema version 4 to 5:

- 1 **Add new column "SuspendBy" of type varchar(32) to OT_InBox tables in the housekeeping database and all LOB databases.**
- 2 **Add new column "SuspendFor" of type varchar(100) to OT_InBox tables in the housekeeping database and all LOB databases.**
- 3 **Add new column "SuspendTill" of type datetime to OT_InBox tables in the housekeeping database and all LOB databases.**
- 4 **For each workflow in OT_Workflows in the housekeeping database, alter the select clauses for the main, pending, and overdue workflow views to add SuspendBy, SuspendFor, and SuspendTill columns from OT_InBox to the selection.**
- 5 **For each workflow in OT_Workflows in the housekeeping database, insert new records into OT_Indexes for the new SuspendBy, SuspendFor, and SuspendTill view columns in the main, pending and overdue views. (A total of nine records are inserted for each workflow.)**
- 6 **For each workflow in OT_Workflows in the housekeeping database, insert new records into OT_PowerCabIndexes for the new SuspendBy, SuspendFor, and SuspendTill view columns in the**

main, pending and overdue views. (A total of nine records are inserted for each workflow.)

- 7 For each workflow in OT_Workflows in the housekeeping database, update record (found by PowerCabID) in OT_PowerCabinets by concatenating " and (A_ResultID > 0)" to its filter column.
- 8 Update the SchemaVersion property in OT_SysAttrs with PropValue="5"

Schema Version 6

Database Schema 6 was released with Documanager 6.3.

Database Changes

Schema version 6 supports Notes for folders and projects, and Diaries. To get from schema version 5 to 6:

- 1 **Add a new table OT_Notes, to the housekeeping database. This table has six columns: AddedBy (varchar(32) not null), AddedOn (datetime not null), Notes (varchar(1024)), AdditionalInfo (varchar(64)), Deleted (smallint), and RecordID (int not null).**
- 2 **Add a new table OT_Diaries, to the housekeeping database. This table has fourteen columns:**
DiaryID (varchar(40) not null), Title (varchar(512)), Description (varchar(1024)) ,

CreatedOn (datetime not null), LastModifiedOn (datetime not null),

StartDate (datetime), NextNotification (datetime), DueDate (datetime),

CompletedOn (datetime), AssignedBy (varchar(32)),

Appendix B: Database Schema

Database Schemas Defined

Owner (varchar(32) not null), Attachment (varchar(512)), Priority (smallint), Status (smallint).

Its Primary Key is DiaryID.

Schema Version 7

Database Schema 7 was released with Documanager 6.4.

Database Changes

Schema version 7 puts version information in OT_Docs, creates additional indices and adds primary keys. Schema version 7 is not backwards compatible with earlier versions of Documanager—it will not work with Documanager version 6.3 Servers, for instance. To get from schema version 6 to version 7:

- 1 **Add VersionNotes column to OT_Docs VARCHAR (240).**
- 2 **Copy data from OT_DocVersions.VersionNotes to OT_Docs.VersionNotes.**
- 3 **Create OT_Docs index on RecordID.**
- 4 **Add any missing primary keys.**
- 5 **Create indices on OT_UsersGroups and OT_GroupsGroups.**
- 6 **Update the SchemaVersion and MinAppSchema in OT_SysAttrs with PropValue = “7”**

Schema Version 8

Database Schema 8 was released with Documanage 6.4 SR1.

Database Changes

Schema Version 8 supports global authorities for groups and allows for longer names of groups. To get from schema version 7 to version 8:

- 1 **Add a new table, OT_GroupAuths, to the housekeeping database. This has two columns, GroupID (varchar(64), not null) and CanImpersonate (smallint).**
- 2 **Change the lengths of varchar columns from 32 to 64: OT_TaskTeamGroups.MemberID, OT_Authority.GroupID, OT_Groups.GroupID, OT_GroupsGroups.GroupID, OT_UsersGroups.GroupID, OT_WorkAuthority.GroupID, and OT_GroupsGroups.MemberGroupID.**
- 3 **Insert a new record ('Users', NULL, NULL, NULL) into OT_Groups.**
- 4 **Update the SchemaVersion property in OT_SysAttrs with PropValue="8"**

Appendix B: Database Schema

Database Schemas Defined

Schema Version 9

Database Schema 9 was released with Documange 6.4 SR2.

Database Changes

Schema Version 9 supports global authorities for groups, longer names of groups, and new document attributes for identifying file contents. To get from schema version 8 to version 9:

- 1 **Add a new table, OT_LastRenditionID, to the housekeeping database, with one column, RenditionID (int), which is also its primary key. Initialize the table with one record, with a value of 1 in the column.**
- 2 **Add ContentsTag (int not null), AnoAnnotationsTag (int not null), RenditionID (int not null), and RenditionKey (varchar 256) columns to the OT_Docs table.**
- 3 **Add CanEditRenditions (smallint) column to the OT_GroupAuths table.**
- 4 **Ensure that the SourceTable columns for all OT_Docs records refer to their actual source database tables, never to their workflow views.**
- 5 **Update the SchemaVersion property in OT_SysAttrs with PropValue="9"**

Schema Version 10

Database Schema 10 was released with Documanage 6.4 SR4

Database Changes

Schema Version 10 supports document retention and protection. To get from schema version 9 to version 10:

- 1 **Add RetentionOptions (int), RetentionInterval (int), RetentionDate (datetime), Protection (smallint), and Deletion (smallint) columns to the OT_ObjectClasses table.**
- 2 **Set RetentionOptions, Protection, and Deletion to 0 for all records in OT_ObjectClasses.**
- 3 **Add ReviewDate (datetime), ProtectionDate (datetime), Protected (smallint), and Held (smallint) columns to all OT_Docs tables.**
- 4 **Set Protected and Held to 0 for all records in all OT_Docs tables.**
- 5 **Create new OT_ServerTasks table in housekeeping database with TaskName (varchar(64) not null), ServerID (varchar(128) not null), Heartbeat (datetime not null), and Status (smallint not null) columns and a primary key on TaskName.**
- 6 **Insert a new row in OT_SysAttrs where 'PropName' = 'RetentionTask' and 'PropValue' = "00:00:00" to indicate that the retention process should run at midnight GMT.**
- 7 **Update the SchemaVersion property in OT_SysAttrs with PropValue="10".**

Appendix B: Database Schema

Database Schemas Defined

Schema Version 11

Database Schema 11 was used for internal development.

Schema Version 12

Database Schema 12 was released with Documanage 6.4 SR.5.

Database Changes

Schema Version 12 supports subdirectory storage, and ID caching, to reduce the high frequency of updates to tables OT_LastDocID and OT_LastRecordID. To get from schema version 10 to 12:

- 1 **Create new OT_IDranges table in housekeeping database with CachedIDtype (int not null), RangeStart (int not null), and RangeSize (int not null) columns and a Primary Key on the combination of CachedIDtype and RangeStart.**
- 2 **Add DocPath(varchar(64)), AnnoPath(varchar(64)) and AnnoVolume(varchar(10)) to all OT_Docs tables.**
- 3 **Add Spec(varchar(5)), Indexed(smallint) and Path(varchar(256)) to OT_Volumes table.**
- 4 **Decrypt data in AreaPtr column of OT_Volumes table and fill data for new columns Spec, Indexed and Path in this table.**
- 5 **Make the Spec and Path columns of OT_Volumes table "Not NULL".**
- 6 **Update the SchemaVersion and MinAppSchema properties in OT_SysAttrs with PropValue="12".**

Semantic Changes

Under schema version 12, the DocPath and AnnoPath columns in the OT_Docs table contain the path of the storage subdirectory relative to the path of the Volume where the document and annotation files are physically stored. These columns will be empty or NULL for documents kept on storage that does not support subdirectories and for the old documents stored at the root of the Volume.

Schema version 12 supports multiple annotation volumes per Document Category. The AnnoVolume column in the OT_Docs table records the name of the volume where the annotation file is stored for the document. AnnoVolume will be NULL or empty for documents that do not have any annotations or for old documents whose annotation files are stored in the default annotation volume.

The Spec, Indexed and Path columns in OT_Volumes table record the storage specifier, whether the volume is indexed or not and the path of the storage volume in clear text. This information was stored as a single encrypted string in the AreaPtr column of OT_Volumes in previous versions of Documanage.

Schema Version 13

Database Schema 13 was used for internal development.

Appendix B: Database Schema

Database Schemas Defined

Schema Version 14

Database Schema 14 was released with Documanage 6.5.

Database Changes

Schema Version 14 supports Category by Cabinet and some additional Workflow and Task properties. This version also has the database changes necessary for Task Containers. To get from schema version 12 to 14:

- 1 **Create new OT_ExcludeCategories table in housekeeping database with PowerCabID (varchar(32) not null) and Category (varchar(15) not null) columns, a Primary Key on the combination of PowerCabID and Category, a Foreign Key on PowerCabID referencing PowerCabID in OT_PowerCabinets table and a Foreign Key on Category referencing ObjectClass in OT_ObjectClasses table.**
- 2 **Add MaxProjectDuration (int) to OT_Workflows table.**
- 3 **Add MaxTaskDuration (int), MaxWorkDuration (int) and Deactivated (smallint) to OT_Tasks table.**
- 4 **Add ProcessOrder (int) to OT_TaskLinks table.**
- 5 **Create new OT_TaskContainers table in housekeeping database with ContainerID (int not null), ParentID (int), WorkflowID (varchar(32)), Name (varchar(100) not null), Description (varchar(240)), Manager (varchar(32)) columns and a Primary Key on ContainerID.**
- 6 **Create new OT_ContainerTeams table in housekeeping database with ContainerID (int not null), GroupID (varchar(64) not null) columns and a Primary Key on the combination of ContainerID and GroupID.**
- 7 **Add ContainerID (int) to OT_Tasks table.**

- 8 **Update the SchemaVersion property in OT_SysAttrs with PropValue="14".**

Appendix B: Database Schema

Database Schemas Defined

Schema Version 15

Database Schema 15 was released with Documanage 6.6.

Database Changes

Schema Version 15 supports longer values for cabinet names, category names, and filetype, and adds new columns to OT_Docs to simplify and clarify data formerly packed into OT_Docs.CheckedOutFor. To get from schema version 14 to 15:

- 1 **Change the lengths of varchar columns to 32: OT_Docs.ObjectClass (in each OT_Docs table), OT_ObjectClasses.ObjectClass, OT_ClassPickLists.ObjectClass, OT_ClassVolumes.ObjectClass, OT_ExcludeCategories.Category.**
- 2 **In table OT_Volumes, select records where Spec is "BLOB". For each table in the Path column, change the length of varchar column EZPName to 44.**
- 3 **Change the lengths of varchar columns to 64:
OT_PowerCabinets.PowerCabID,
OT_ExcludeCategories.PowerCabID,
OT_PowerCabConsd.PowerCabID,
OT_PowerCabGroups.PowerCabID,
OT_PowerCabIndexes.PowerCabID,
OT_PowerCabLevels.PowerCabID, OT_PowerCabVars.PowerCabID,
OT_ScreenKeys.PowerCabID, OT_Workflows.WorkCabinet.**
- 4 **For each OT_Docs table, add the columns RefCabinet (varchar 64), SourceFldrFilter (varchar 124), and CheckedOutDocID (int).**
- 5 **For each OT_Docs table, parse the data in CheckedOutFor into CheckedOutFor, RefCabinet, SourceFldrFilter, and CheckedOutDocID. See "Semantic Changes" below.**

- 6 In table **OT_TxLog**, change the length of varchar column **TxTable** to **100**.
- 7 Update the **SchemaVersion** and **MinAppSchema** properties in **OT_SysAttrs** with **PropValue = "15"**.

Semantic Changes

Previously, **OT_Docs.CheckedOutFor** was packed with multiple pieces of information. Now, **OT_Docs.CheckedOutFor** holds only the user-supplied reason for checking out a document. **OT_Docs.RefCabinet** holds the source document's or checked-out copy's cabinet name for checked-out copies or checked-out original documents, respectively. **OT_Docs.SourceFldrFilter** holds the source document's folder filter, and **OT_Docs.CheckedOutDocID** holds the checked out document's DocID. None of these fields apply to historic versions of a document.

Appendix B: Database Schema
Database Schemas Defined

Appendix C: Parameterized SQL

Documanager now uses parameterized SQL, also known as dynamic SQL, for some database access. This method improves performance by eliminating redundant SQL statement construction, and by reducing interaction with the database. Documanager has implemented this enhancement in two ways; by taking advantage of database caching and by using already prepared statements.

Database caching

Some database software can be configured to automatically detect identical SQL command strings. They can take advantage of identical commands prepared earlier. This improves performance by allowing the database to respond much more quickly to a prepare request. For example, two commands that insert a database record in a table will vary only in the data values they insert. Both SQL commands would be similar, only the data values they specify will differ. Parameterized SQL separates data values from the SQL command allowing use of the exact SQL command with different data values each time the command is executed. In the command, a question mark placeholder will appear where the data value is expected, and data will be supplied when the statement is executed.

Statement Reuse

The database client (Documanager Server) can also keep and re-execute the handle of a statement that it has prepared. This removes the overhead of requesting the DBMS to prepare the statement before every execute call.

SQL Logging

When SQL Logging is enabled for Documanager, SQL commands are logged as they are prepared. You will see two main differences in the logs generated by Documanager 6.3 and later (compared to earlier Documanager versions):

- 1 **Many commands that previously included data will now show question marks in place of data.**
- 2 **Statements are logged when they are prepared, as well as when they are executed.**

Connections and Statement Handles

When a Documanager system is configured to support multiple connections on a database, each statement to be used may be prepared once for each connection. You may see the same statement prepared a handful of times because it is being prepared on different connections.

In some cases, the DBMS may be able to optimize some of these prepare calls through its cache.

ODBC Driver Cursor Commit Behavior

In the ODBC specification, databases are permitted to discard all the access patterns stored for a connection, effectively un-preparing all the prepared statements, whenever a transaction is committed or a simple statement is executed.

If an ODBC driver in use indicates that this is the case for a particular connection, Documange will always prepare statements before executing them on that connection.

ODBC Driver Bugs

Some versions of an ODBC driver may not properly support bound parameters. If you encounter strange behavior, errors, or crashes, please verify that you have installed the latest ODBC drivers for your DBMS.

MS SQL Server - Use ODBC drivers 2000.81.9031.38 (dated 2003-3-11), or later. This driver version is installed with MDAC 2.7 SP1 or SQL Server 2000 SP3a.

Appendix C: Parameterized SQL
Connections and Statement Handles

Appendix D: Database Naming Conventions Before Ver. 5.2

Different naming conventions are used for database aliases, synonyms and owner names, depending on whether you are using version 5.2 and later, or a version before 5.2. This appendix discusses naming conventions used for versions of Documanage before version 5.2, or the conventions used when the administrator has used the default database and owner name.

Early versions of Documanage required an explicit implementation convention within supported databases. The document management tables were created and owned by an "EZPOWER" user and generally stored in databases named "EZPOWER". This naming convention was accompanied by a mixture of implicit and explicit qualifiers in SQL code styles, some of which reference tables as "EZPOWER.tablename" and others, which merely reference "tablename."

The old "EZPOWER" naming conventions created problems in large, policy-driven installations, particularly those using IBM DB2 DBMSs. These IBM installations have well established and wisely chosen conventions for naming databases, table spaces, and database owners. Before Documanage was modified to facilitate these naming conventions, some additional set up work was required to allow Documanage to operate in a DB2 installation.

Since conventions vary between installations, for the sake of this discussion we assume the following naming system. This can easily be transferred to other schemes:

Appendix D: Database Naming Conventions Before Ver. 5.2

Table owner: TO

Databases owned by TO: TODBxxxx

Table spaces owned by TO: TOTSxxxx

User account used to access OS/390 via ODBC connection: ODBCUSER

Documange is not sensitive to the naming schemes or hierarchical grouping for database and tablespace names. Recommendations on these topics appear elsewhere in this document. The table owner may be established as TO, but for the product to operate in this case, these additional steps must be taken:

An alias must be set up for each table in the Documange table set under the alias owner EZPOWER. For each table the following SQL statement must be executed:

- ◆ For each Documange table:
CREATE ALIAS EZPOWER.tablename FOR TO.tablename
- ◆ Also, under the effective authID of the ODBCUSER, a synonym for each tablename must be created:
SET CURRENT SQLID = 'ODBCUSER';
- ◆ Required for each Documange table:
CREATE SYNONYM tablename FOR TO.tablename

Different Documange Servers may use multiple ODBCUSER accounts. If this configuration is desired, the additional set up of the synonyms above must be repeated for each account used by a Documange Server or Administration program to access the Documange tables.

Appendix E: Storing documents in NTFS and SNAP volumes

When Documanage stores documents in NTFS and SNAP volumes, it creates an hierarchical structure with five levels of up to 2048 subdirectories each under the volume root. It assigns document files and annotation files numerical names based on the DocIDs in the leaf subdirectories, while it stores any files with non-numeric names, including footprint.txt files and temp files, in the volume root.

Documanage creates subdirectories in the storage volume as required. An algorithm that uses a document's Address (numerical name) places it in the directory structure.

Storing Subdirectory Paths

The subdirectory paths—relative to the Volume path—for document files and annotation files are stored in the DocPath and AnnoPath columns of the OT_Docs table. The corresponding columns in the OT_Docs record store all of the information required to construct a document file or annotation file. The path for a document file can be constructed by concatenating a Volume's Path (from the Path column in OT_Volumes table corresponding to this Volume), DocPath, Address, ".", and the Type columns of OT_docs table for that document.

Appendix E: Storing documents in NTFS and SNAP volumes

The complete path for an annotation file can be constructed in the same way by concatenating AnnoVolume's Path (the Path column in OT_Volumes table corresponding to this AnnoVolume), AnnoPath, Address, ".", "ANO" or "ART".

Subdirectory Path Example

For example, a document file and its annotation file have the following data in their OT_Docs table record:

DocID	Address	Type	Volume	DocPath	AnnoPath	AnnoVolume
500	502	txt	MyVol1	0\0\0\0\0	0\0\0\0\0	MyVol2

If the paths of the volumes MyVol1 and MyVol2 from the OT_Volumes table are "C:\Vol1" and "D:\Vol2" respectively, then the complete path of the document file is "C:\Vol1\0\0\0\0\0\502.txt" and the complete path of the annotation file is "D:\Vol2\0\0\0\0\0\502.ANO".

Documents Stored Using Versions Earlier Than Service Release 5

The files Documents stored using a version of Documanage before version 6.4 SR5, will remain in the volume root on NTFS and SNAP storage. Only new files and updated files will be created in the appropriate subdirectory in the volume.

BLOB and VLAM Storage

Subdirectory storage does not apply to and is not supported by the BLOB and VLAM storage types. The files on BLOB and VLAM volumes will be stored at volume root. Because of this, in OT_Docs, the subdirectory path strings DocPath and AnnoPath will be empty or NULL for such documents.

Appendix E: Storing documents in NTFS and SNAP volumes
