

enaio[®]

Softwaredokumentation
enaio[®] Systemhandbuch DMS

Version 8.50

Sämtliche Softwareprodukte sowie alle Zusatzprogramme und Funktionen sind eingetragene und/oder in Gebrauch befindliche Marken der OPTIMAL SYSTEMS GmbH, Berlin oder einer ihrer Gesellschaften. Sie dürfen nur mit gültigem Lizenzvertrag benutzt werden. Die Software sowie die jeweils zugehörige Dokumentation sind nach deutschem und internationalem Recht urheberrechtlich geschützt. Das illegale Kopieren und Vertreiben der Software stellt Diebstahl geistigen Eigentums dar und wird strafrechtlich verfolgt. Alle Rechte vorbehalten, einschließlich der Wiedergabe, Übermittlung, Übersetzung sowie Speicherung mit/auf Medien aller Art. Für vorkonfigurierte Testszenarien oder Demo-Präsentationen gilt: Alle Firmennamen und Personen, die in Beispielen (Screenshots) erscheinen, sind frei erfunden. Eventuelle Ähnlichkeiten mit tatsächlich existierenden Firmen und Personen sind zufällig und unbeabsichtigt.

Copyright 1992 – 2018 by OPTIMAL SYSTEMS GmbH
Cicerostraße 26
D-10709 Berlin

11.06.2018
Version 8.50

Inhalt

| | |
|--------------------------------------------------------------|----|
| Vorwort | 7 |
| Allgemeine Beschreibung des DMS | 8 |
| Konfigurierbarkeit und Abbild der Informationsstrukturen | 8 |
| Dokumentenimport, -konvertierung und -verarbeitung | 8 |
| Dokumentenbearbeitung und Recherche | 9 |
| Dokumentenausgabe und Export | 9 |
| Archivierung | 9 |
| 3-Tier-Architektur, Verfügbarkeit, Benutzeranzahl | 9 |
| Datensicherheit | 10 |
| Zukunftssicherheit | 11 |
| Systembeschränkungen | 11 |
| enaio®-Versionen und -Konfigurationen | 11 |
| Systemarchitektur | 12 |
| Datenbank und Dokumentenablage | 12 |
| Objektdefinition, Verschlagwortungsmasken und Objekttabellen | 12 |
| Medienstruktur | 17 |
| Anwendungsserver | 21 |
| Loadbalancing und Skalierbarkeit | 21 |
| Modularität | 23 |
| Multithreading | 24 |
| Clusterfähigkeit | 26 |
| Session-Management | 27 |
| Clientprogramme | 28 |
| Integrierter Workflow | 28 |
| Schnittstellen zu anderen Systemen | 28 |
| COM-Schnittstelle | 28 |
| Office Makros | 29 |
| Schnittstelle zu SAP | 30 |
| enaio® file-system-archiver | 35 |
| enaio® communicator | 35 |
| Optionale Komponenten | 37 |
| Anzeigedienste | 37 |
| enaio® appconnector | 38 |
| enaio® pdfa-dispatcher | 39 |
| enaio® mail-archiver | 39 |
| Datensicherheit | 39 |
| Unterstützung durch die Systemarchitektur | 39 |
| Benutzer, Benutzergruppen und Systemrechte | 40 |
| Weitere Funktionen zur Datensicherheit | 42 |
| Installation von enaio® | 44 |
| Überblick | 44 |
| Ausbaustufen des enaio®-Systems | 45 |
| Einzelplatzsystem | 46 |
| Einfaches System mit einem Anwendungsserver | 46 |
| Komplexes System mit mehreren Servergruppen | 47 |

| | |
|-----------------------------------------------------------------------------------|----|
| Anlegen von Dokumenten | 49 |
| Allgemeine Voraussetzungen..... | 49 |
| Die Objektdefinition in der Datenbank | 50 |
| Objektrelationen in der Datenbank..... | 50 |
| Konfigurationseinträge für Kataloge, AddOns und das Dialogelement 'Tabelle' | 50 |
| Verschlagwortung der Ordner-, Register-, Dokumenttypen..... | 51 |
| Der Standort von Registern und Dokumenten | 53 |
| Die Protokollierung in der Datenbank..... | 54 |
| Dokumentensperrung in der Datenbank..... | 55 |
| Verschlagwortung und Standort von DMS-Objekten | 56 |
| Anlegen von Ordnern und Registern | 57 |
| Anlegen von Bilddokumenten..... | 57 |
| Anlegen von Windows-Dokumenten | 60 |
| Übernahme von E-Mails..... | 63 |
| enaio® document-storage | 65 |
| Übernahme von Filmdokumenten..... | 66 |
| Übernahme von XML-Dokumenten | 66 |
| Übernahme über das Dateisystem..... | 67 |
| Übernahme über enaio® printer | 67 |
| Modulübergreifende Dokumente | 68 |
| Dokumente ohne Seiten | 68 |
| Die Ablage..... | 69 |
| Arbeiten mit Dokumenten | 71 |
| Allgemeine Voraussetzungen..... | 71 |
| Recherchieren..... | 72 |
| Trefferlisten..... | 73 |
| Ordnerfenster | 74 |
| Gespeicherte Anfragen | 75 |
| SQL-Anfragen..... | 76 |
| Anzeige von XML-Dokumenten | 77 |
| Ordner in der Objektsuche | 77 |
| Verweise auf DMS-Objekte | 77 |
| Verknüpfungen zu externen Programmen..... | 78 |
| Mappen | 79 |
| Wiedervorlagen | 80 |
| Abonnements | 81 |
| Notizen und Verknüpfungen | 82 |
| Exportieren von Daten..... | 82 |
| Verschlagwortung exportieren | 83 |
| Dateien exportieren..... | 84 |
| Versenden von DMS-Objekten | 84 |
| Ausdruck von Daten | 85 |
| XML-Daten..... | 86 |
| Druckkennzeichnung für DMS-Dokumente..... | 88 |
| Formulardruck | 89 |
| Bearbeiten von Dokumenten | 90 |
| Allgemeine Voraussetzungen..... | 90 |
| Änderung der Verschlagwortung | 90 |
| Indexdatenhistorie..... | 91 |
| Ordner-, Register-, Dokumentverschlagwortung..... | 92 |
| Sammeländerungen..... | 92 |
| Ändern des Standorts..... | 93 |
| Mehrere Standorte durch Verweise..... | 93 |
| Änderung von Dokumentdateien | 94 |

| | |
|---------------------------------------------------------------------------------|---------|
| Veränderungen an Bild-Dokumenten | 95 |
| Veränderungen an Film-Dokumenten..... | 95 |
| Veränderungen an Windows-Dokumenten | 95 |
| Anmerkungen auf Folien..... | 97 |
| Elektronische Signatur | 98 |
| Löschen von Objekten | 99 |
| Bearbeitungshistorie..... | 101 |
| Wiederherstellen der Verschlagwortung..... | 101 |
| Wiederherstellen von Versionen | 101 |
| Volltextsuche | 103 |
| Überblick | 103 |
| Sicherheit | 104 |
| Überblick | 104 |
| Schutz von Daten und Dokumentdateien | 105 |
| Absicherung auf Netzwerk-, Domänen- und Betriebssystem-Ebene | 105 |
| Absicherung durch das enaio®-System..... | 105 |
| Objektidentität..... | 108 |
| Sicherung der App OS mobileDMS unter iOS | 111 |
| Archivierung | 114 |
| Backup..... | 114 |
| Grundlagen..... | 115 |
| Einleitung..... | 115 |
| Ablaufschema der Arbeit mit archivierten und nichtarchivierten Dokumenten | 116 |
| Medien, Medienset und Spiegelmedien | 118 |
| Archivierung und Zugriff auf archivierte Dokumente..... | 119 |
| Arbeit mit archivierten Dokumenten | 126 |
| Externe Archivierungssysteme..... | 129 |
| Administration | 130 |
| Überblick | 130 |
| Abgrenzung: Administrations-Tätigkeiten außerhalb des enaio®-Systems..... | 130 |
| Abgrenzung: Produktive und administrative Tätigkeiten innerhalb von enaio®..... | 131 |
| enaio®-Administration | 131 |
| Vorbereitung des Produktivbetriebs | 133 |
| Installation | 133 |
| Erstellung der Objektdefinition | 134 |
| Einrichtung des Sicherheitssystems..... | 137 |
| Dezentrale Benutzeradministration | 144 |
| Active Directory Service | 145 |
| Authentifizierungs-Reihenfolge..... | 145 |
| Active Directory Synchronisation | 146 |
| Einrichtung des Lizenzsystems | 146 |
| Einrichtung von Dokumenttypen | 148 |
| Events und Scripting (Eventeditor, AddOn, Index-Skripte) | 149 |
| Systemerhaltung und -Pflege..... | 154 |
| Überwachung des Anwendungsservers..... | 154 |
| Kommunikation zwischen Server und Client..... | 154 |
| Automatische Aktionen | 155 |
| Mehrsprachigkeit..... | 158 |
| Startverhalten des enaio® client..... | 159 |
| Kommandozeilenparameter des enaio® client | 160 |
| Zusammenfassung der enaio®-Administrationstools..... | 161 |
| enaio® editor | 161 |
| enaio® administrator..... | 161 |
| enaio® enterprise-manager | 162 |

| | |
|-------------------------------------------------------------|---------|
| Protokolleinrichtung | 162 |
| enaio® editor-for-events | 162 |
| enaio® capture im Administrationsmodus | 163 |
| Serverkonsole | 163 |
| Transport von enaio® Systemeinstellungen | 163 |
| Aufbau eines Testsystems | 163 |
| Objektdefinition | 164 |
| Benutzerdaten und Zugriffsrechte | 166 |
| W-Vorlagen | 166 |
| Registrierungseinträge | 167 |
| Konfigurationseinträge in der <code>as.cfg</code> | 168 |
| Automatische Aktionen zum Datenimport und Datenexport | 169 |
| Events | 170 |
| Workflow | 171 |
| enaio® webclient / Webanwendungen | 172 |
| Protokollierung | 172 |
| SQL-Anfragen | 172 |
| enaio® document-storage | 172 |
| enaio® capture | 173 |
| Protokollierung | 174 |
| Protokollierung - Einführung | 174 |
| enaio® manager-for-logfiles | 175 |
| enaio® protocol-viewer | 175 |
| COM-Schnittstelle | 177 |
| Überblick | 177 |
| Interfaces der COM-Schnittstelle | 177 |
| COM-Schnittstelle und enaio®-Sicherheitssystem | 178 |
| enaio® capture | 179 |
| Überblick | 179 |
| Administration von enaio® capture | 180 |
| Voraussetzungen für den Einsatz von enaio® capture | 180 |
| Verteilung auf mehrere Arbeitsstationen im Netz | 184 |
| Anhang | 188 |
| Systemrollen | 188 |
| enaio® enterprise-manager | 192 |
| Registrierungseinträge für enaio® server | 193 |
| Start von enaio® server und die Kommandozeile | 193 |
| Switches und Variablen | 193 |
| Startvorgang | 194 |
| Kommandozeile | 195 |
| Main Thread | 196 |
| Report | 197 |
| Die Server-Konsole | 199 |
| Checkliste bei Problemen beim Serverstart | 202 |
| Sprache der Oberfläche in enaio® umschalten | 207 |

Vorwort

Dieses Systemhandbuch wendet sich an alle, die über eine reine Bedienung hinaus Informationen über Abläufe, Datenflüsse, Wechselwirkungen, Strukturen und Technologien des Dokumenten-Management-Systems (DMS) enaio® benötigen. Es ist für alle Anwender und Administratoren gedacht, die mit Einrichtung, Konfiguration und Betreuung der Produktfamilie enaio® beschäftigt sind.

Es wurde versucht, alle Informationen zusammenzutragen und zu systematisieren, die benötigt werden, um eine Vorstellung über die Systemarchitektur sowie die Wechselwirkungen zwischen den Systemkomponenten zu entwickeln. Damit sollte es möglich sein, Abläufe bei bestimmten Vorgängen besser zu verstehen und den Sinn sowie die Vor- und Nachteile bestimmter Einstellungen, Konfigurationen und Installationen zu erkennen.

Es wurde vermieden, soweit es für die Verständlichkeit möglich war, Informationen der Benutzer-Handbücher zu wiederholen. An einigen Stellen wird auf weitergehende Beschreibungen verwiesen, die konkrete Informationen über Teilbereiche des DMS enthalten, wie z.B. die Beschreibung der Datenbank, des Anwendungsservers, der Installation etc.

Das Systemhandbuch unterstützt den Anwender in Fragen der Konfiguration, des Betriebs und der Wartung. In diesem Sinne soll das Systemhandbuch auch Hilfe beim Beheben von Problemen und der Erkennung von Fehlern sein.

Das Handbuch wurde in zwei Teile untergliedert. In diesem ersten Teil wird das Dokumenten-Management-System und im zweiten Teil das Workflow-Management-System innerhalb der Produktfamilie behandelt.

Allgemeine Beschreibung des DMS

Ein Dokumenten-Management-System (DMS) ermöglicht die Erfassung (Import), das Speichern (Ablage, Archivierung), die Verwaltung (Recherche), die Bearbeitung und Bereitstellung (Export) digitaler Dokumente. enaio® ist ein DMS auf der Basis der Betriebssystemfamilie MS Windows. Im Folgenden sollen die wichtigsten Punkte und ihre Umsetzung in enaio® dargestellt werden.

Konfigurierbarkeit und Abbild der Informationsstrukturen

Ein wesentlicher Vorteil von enaio® ist, dass es die Informations- und Organisationsstruktur der Unternehmensbereiche, in denen es eingesetzt werden soll, adäquat abbilden kann, so dass es an die Strukturen und Abläufe im Unternehmen angepasst wird und nicht umgekehrt.

Für das Einrichten von Schränken, Ordnern und Registern stellt enaio® eine Reihe von Programmmodulen wie den enaio® editor und den enaio® administrator, aber auch Einstellmöglichkeiten im enaio® client zur Verfügung, womit beliebig viele Möglichkeiten von Informationsstrukturen erzeugt werden können.

enaio® unterstützt alle wichtigen, gebräuchlichen Dokumenttypen, wie

- § Pixelgrafiken (z. B. TIFF, JPEG, BMP),
- § E-Mails,
- § XML und
- § alle Dokumente von Windows-Anwendungen.

Neben der Konfiguration der Informationsstruktur unterstützt enaio® auch die Abbildung der Organisationsstruktur eines Unternehmens / Unternehmensbereichs. Im enaio® administrator können beliebige Benutzer und Benutzergruppen eingerichtet werden, die den entsprechenden Organisationseinheiten des Unternehmens entsprechen können.

Dokumentenimport, -konvertierung und -verarbeitung

Mit einem DMS werden digitale Dokumente verwaltet und nicht hauptsächlich erzeugt. enaio® erlaubt das manuelle Einfügen von Dokumenten im enaio® client und besitzt leistungsfähige Funktionen, mit denen ein automatischer Import großer Mengen von Dokumenten durchgeführt werden kann. Außerdem verfügt enaio® über eine große Anzahl von Filter- und Konvertierungsprogrammen, über die Dokumente bestimmter Formate (z.B. COLD, DICOM etc.), aufbereitet und importiert werden können. Dabei kann eine automatische Erfassung von Indexdaten erfolgen. Die Stapelbelegerfassung enaio® capture ermöglicht die automatisierte Erfassung von großen Belegmengen. Außerdem stellt enaio® verschiedene Schnittstellen zur Datenübernahme aus anderen führenden Systemen (SAP, b2, u.v.m.) zur Verfügung.

Dokumentenbearbeitung und Recherche

Der enaio® client verfügt über eine eigene Bedienungsfläche, mit der grafische Dokumente bearbeitet werden können, wie z. B. Hinzufügen oder Löschen von Seiten oder die Veränderung der Reihenfolge von Seiten im Dokument. Außerdem können dem Dokument Notizen, Bemerkungen, Anmerkungen etc. hinzugefügt werden.

Für Windows-Dokumente wird die jeweils registrierte Anwendung zur Bearbeitung aufgerufen. Speziell für MS Office-Anwendung (Word, Excel, Powerpoint) werden geeignete Utilities und Makros zur Verfügung gestellt, mit deren Hilfe Daten aus dem DMS in ein konkretes Dokument übernommen werden können. Über solche Erweiterungen wie den Formulareindruck kann eine automatische Bearbeitung einer großen Anzahl gleichförmiger Dokumente erfolgen, wie sie z. B. in öffentlichen Verwaltungen oder auch im Gesundheitswesen vorkommen.

Neben der Recherche in der Datenbank bietet enaio® auch ebenfalls die Möglichkeit eine Volltextsuche über alle im DMS befindlichen Informationen und Dokumente durchzuführen. Außerdem wird in diesem Zusammenhang die OCR von FineReader unterstützt, so dass auch alle z.B. gescannten Dokumente durchsucht werden können.

Dokumentenausgabe und Export

Nachdem Dokumente importiert und bearbeitet wurden, sollen sie auch wieder ausgegeben werden können. enaio® bietet über Drucken und Speichern im enaio® client eine manuelle Ausgabe von Dokumenten. Zusätzlich wird ein Export-Modul zur Verfügung gestellt, mit dem ein automatischer Export der Dokumente realisiert werden kann.

Archivierung

enaio® ist in der Lage, Dokumente strukturiert zu archivieren, so dass die Anforderungen an eine 'Dokumentenechte Archivierung' erfüllt werden. Eine regelmäßige Archivierung schützt vor Datenverlust und erhöht die Effektivität der Arbeit mit aktuellen Dokumenten. enaio® bietet eine automatische Archivierung, die so konfiguriert werden kann, dass regelmäßig zu bestimmten Zeiten (z. B. nachts) Dokumente, die bestimmten Kriterien genügen, archiviert werden, ohne dass dabei die Arbeit am System eingestellt werden muss.

Dabei kann enaio® mit externen Systemen wie iTernity, Centera oder iXOS zusammenarbeiten.

3-Tier-Architektur, Verfügbarkeit, Benutzeranzahl

Jeder Anwender möchte natürlich auf sein DMS beliebig zugreifen können und vor allem in größeren, gut vernetzten Einheiten werden hohe Anforderungen an die Verfügbarkeit gestellt. Um die Anforderungen eines extensiven Netzbetriebes erfüllen zu können, wurde enaio® in einer klassischen 3-Tier-Architektur implementiert:

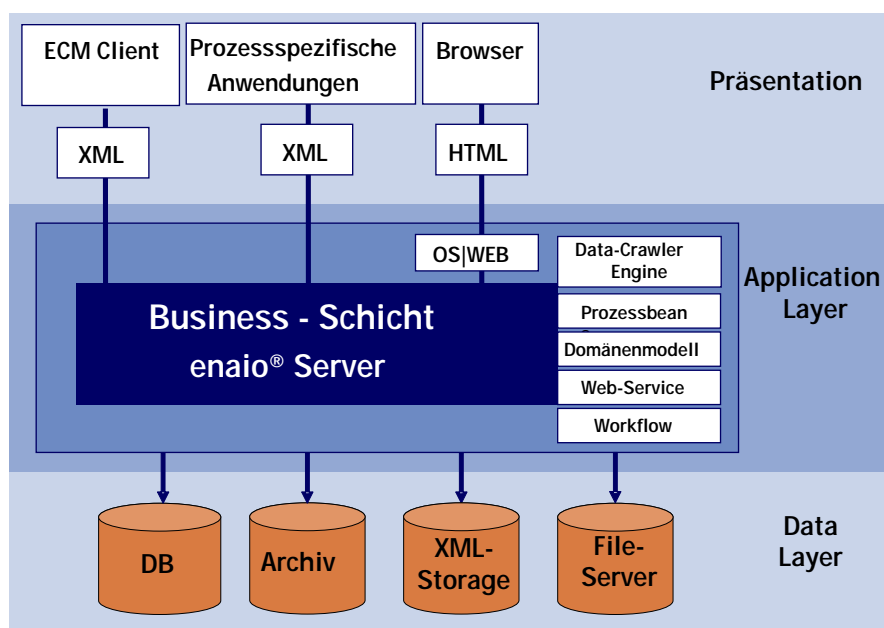


Abbildung 1: 3-Tier-Architektur von enaio

Diese Architektur bietet gerade für den Netzbetrieb große Vorteile. Die Datenbank kann im Netz auf einem speziellen Rechner mit einem beliebigen Betriebssystem in Betrieb sein. Auch eine Jukebox für die Archivierung kann über einen beliebigen (Windows-)Rechner integriert werden. Die Dokumentenablage muss nicht unbedingt auf einem Rechner erfolgen, auf dem sich auch ein enaio®-Server (Anwendungsserver) befindet.

Es können beliebig viele Anwendungsserver im Netz arbeiten. Diese werden in Servergruppen zusammengeschlossen, denen entsprechende Benutzer zugeordnet werden können. Die Dokumente und Informationen des DMS sind in allen Servergruppen zugänglich. Somit ist das Gesamtsystem enaio® beliebig skalierbar. Sollten die schon installierten Anwendungsserver an ihre Leistungsgrenze (meist auch die Leistungsgrenze der Hardware, auf der die Anwendungsserver laufen) stoßen, so braucht nur ein weiterer hinzugefügt werden. Somit kann auf ein beliebiges Anwachsen von Benutzern reagiert und eine hohe Verfügbarkeit erreicht werden. Die Verwaltung von einzelnen Servern und Servergruppen erfolgt über den enaio® enterprise-manager.

Datensicherheit

Datensicherheit bedeutet sowohl der Schutz vor Datenverlust als auch der Schutz vor nicht autorisiertem Zugriff auf Daten. Um den Verlust eines Dokuments schnell bemerken zu können, existieren verschiedene Tools, mit denen die Konsistenz der Datenbankeinträge und der Dokumentenablage überprüft werden kann und die auch automatisch, regelmäßig ausgeführt werden können. Für den Schutz vor unberechtigtem Zugriff auf Daten wurde in enaio® ein komplexes System zur Vergabe von Zugriffsrechten eingerichtet:

- § Nur Benutzer, die im System mit Benutzernamen und Passwort eingetragen wurden (Login-Accounts), können Zugang zu Daten erhalten. Außerdem kann eingestellt werden, ob ein bestimmtes Benutzerkonto gesperrt werden soll, wenn das Login dreimal aufeinander fehlgeschlagen ist.

- § Benutzern werden in Gruppen zusammengefasst, denen bestimmte Zugriffsrechte auf Dokumente und Register erteilt wird. Zugriffsrechte können von Klauseln abhängig gemacht werden.
- § Über Systemrollen wird definiert, welche Benutzer Zugriff auf welche Programme und Funktionsbereiche innerhalb von Programmen erhalten.

Zukunftssicherheit

Für die Zukunftssicherheit von DMS ist der Einsatz von Standardtechnologien von großer Bedeutung. Auch wenn über proprietäre Formate und Protokolle kurzfristig eine bessere Leistung erreicht werden kann, so stellt die Nutzung verbreiteter Standards sicher, dass das System auch in Zukunft lauffähig ist. So verwendet enaio® für die Verbindung der Clientprogramme mit den Servern das Standardnetzprotokoll TCP/IP. Dabei wird XML RPC's verwendet, wodurch beliebige Daten und Parameter übergeben und empfangen werden können. Über das allgemeine XML-Format ist es möglich von beliebigen Plattformen und über das Internet auf den Server zuzugreifen.

Systembeschränkungen

Die maximale Größe von Dokumentdateien, die mit enaio® verwaltet werden können, beträgt in der aktuellen Version 2 GB. Die maximale Seitenanzahl bei Bilddokumenten im PDF-Format beträgt etwa 50.000 Seiten. Bei SinglePage-Tiffs gilt eine maximale garantierte Seitenzahl von 4096. Die Möglichkeit der Verarbeitung derart großer Dateien stellt jedoch weitreichende Anforderungen an die Speicherausstattung der Maschinen auf denen diese Dokumente angezeigt und erzeugt werden. Umfangreichere Dokumentdateien können mit diesem System nicht verwaltet werden.

Des Weiteren soll an dieser Stelle auf die Beschreibung der Systemvoraussetzungen in der begleitenden Dokumentation verwiesen werden. Dort sind weitere Randbedingungen bzgl. der unterstützten Infrastruktur beschrieben.

enaio®-Versionen und -Konfigurationen

Im Regelfall wird die Aufwärtskompatibilität von Konfigurationen in den verschiedenen enaio®-Versionen angestrebt, sodass Konfigurationen, die in einer früheren Systemversion angelegt wurden, auch in späteren Systemversionen verwendet werden können.

Im Gegensatz dazu ist die Abwärtskompatibilität nicht sichergestellt, da die Weiterentwicklung der enaio®-Plattform neue Funktionen und damit Konfigurationserweiterungen umfasst. Konfigurationen aus späteren Systemversionen können also in früheren Systemversionen nicht verwendet werden.

Mit großer Wahrscheinlichkeit gehen beim Versuch, Konfigurationen in frühere Systemversionen zu importieren, Einstellungen für neue Funktionen verloren und es kann zu Formatinkompatibilitäten und/oder Programmfehlern kommen. Deshalb muss beim Betrieb von Test-, Entwicklungs- und Produktivsystemen auf eine Harmonisierung der Versionen geachtet werden und es darf auf keinen Fall der Versuch unternommen werden, Konfigurationen in spätere Systemversionen zu importieren oder zu aktivieren. Das betrifft alle Konfigurationsdateien, insbesondere aber Objektdefinitionen und Workflowmodelle.

Systemarchitektur

Datenbank und Dokumentenablage

Datenbank und Dokumentenablage bilden die Datenschicht der 3-Tier-Architektur.

enaio® verwendet eine Datenbank unter anderem

- § für die Verwaltung der Dokumente,
- § für die Verwaltung weiterer Informationen zu den Dokumenten,
- § für die Zugriffssteuerung auf Dokumente und andere Ressourcen des Systems,
- § für die Speicherung und Verwaltung von Benutzern, Benutzergruppen und Zugriffsrechten.

Die Kommunikation mit den Datenbanken erfolgt über ODBC-Treiber, die das Betriebssystem bereitstellt, oder welche vom Datenbankhersteller mitgeliefert werden.

Dokumente werden in enaio® nicht in der Datenbank, sondern in einem eigenen, speziellen Dateisystem abgelegt. Dadurch werden die Ressourcen der Datenbank geschont und die Performance der Datenbank bleibt erhalten.

Objektdefinition, Verschlagwortungsmasken und Objekttabellen

Unter der Objektdefinition wird in enaio® eine Liste von Verschlagwortungsmasken verstanden, über die die verschiedenen DMS-Objekte angelegt werden. Diese Verschlagwortungsmasken (Abbildung 2) werden mit Hilfe von enaio® editor erzeugt. Sie können eine Vielzahl von Dialogelementen enthalten, die im Handbuch von enaio® editor detailliert beschrieben sind.

Bei enaio® wird die Objektdefinition komplett in der Datenbank gehalten. Dazu dienen die Datenbanktabellen

- § `osobjdef`, Beschreibung der Objekte,
- § `osobjfields`, Beschreibung der Dialogelemente auf den Verschlagwortungsmasken,
- § `osobjparrel`, Festlegung der Objekthierarchie, d.h. welches Objekt in welchem (anderem) DMS-Objekt enthalten sein kann,
- § `osconf`, Speicherung der Daten aus der `aslisten.dat`,
- § `oslistctrl`, Beschreibung des Tabellenelements,
- § `osproperties`, weitere Eigenschaften von Objekten und Dialogelementen.
- § `ospagectrl`, Beschreibung von Pagecontrols.

Abbildung 2: Beispiel einer Verschlagwortungsmaske (hier für E-Mails)

enaio® verwendet folgende DMS-Objekte für die Abbildung von Informationsstrukturen:

- § Schrank - Der Begriff Schrank wurde als Analogie zu einem Aktenschrank gewählt, der Ordner, Register und Dokumente enthält. Er stellt die oberste Verwaltungsebene im DMS dar. Alle weiteren DMS-Objekte können nur als Subobjekt eines Schrankes existieren. Ein Schrank erhält als einziges Attribut einen Namen. Eine Datenbanktabelle wird für einen Schrank nicht erstellt.
- § Ordner - Für jeden Schrank kann genau eine Ordnerart mit einer Verschlagwortungsmaske angelegt werden. Von diesen Ordnern können beliebig viele im Schrank enthalten sein. (Beispiel: Schrank 'Kunden' mit Ordnern 'Kunde X')
- § Register - Register dienen der weiteren Feingliederung der Informationsstruktur. Es können beliebig viele Registertypen in der Objektdefinition angelegt werden.
- § Dokumente - Über die Verschlagwortungsmasken werden die Dokumenttypen festgelegt, die in einem Schrank abgelegt werden können. Ein konkretes Dokument (z.B. ein Brief, ein Fax oder ein gescannter Artikel aus einer Zeitschrift) kann nur über das Ausfüllen (Anlegen) einer entsprechenden Verschlagwortungsmaske in das enaio®-System importiert werden.

Für jedes DMS-Objekt wird in der Datenbank eine eigene Tabelle eingerichtet (siehe Ordner, Register und Dokumente weiter unten). In der Objektdefinition kann auch festgelegt werden, ob und welche Beziehungen zwischen den DMS-Objekten eines Schrankes bestehen. Die Abbildung 3 bis Abbildung 5 zeigen, wie die Relationen zwischen Objekttypen eines Schrankes, in dem ein Ordner 'Geschäftsvorfälle', die Registertypen 'Akte' und 'Partner-Typ', sowie unterschiedliche Dokumenttypen enthalten sind, festgelegt sein könnten. Abbildung 3 zeigt, dass im Ordner nur Register vom Typ 'Akte' angelegt werden können, Abbildung 4 zeigt, dass in einem Register vom Typ 'Akte' nur Register vom Typ 'Partner-Typ' enthalten sein dürfen und Abbildung 5 zeigt, dass nur in Registern vom Typ 'Partner-Typ' die

gewünschten Dokumente abgelegt werden können. Mit diesen Einstellungen wird also folgende Objekthierarchie erzeugt:

Ordner 'Geschäftsvorfälle' / Register 'Akte' / Register 'Partner-Typ' /
Dokument D,P,X,W,XML

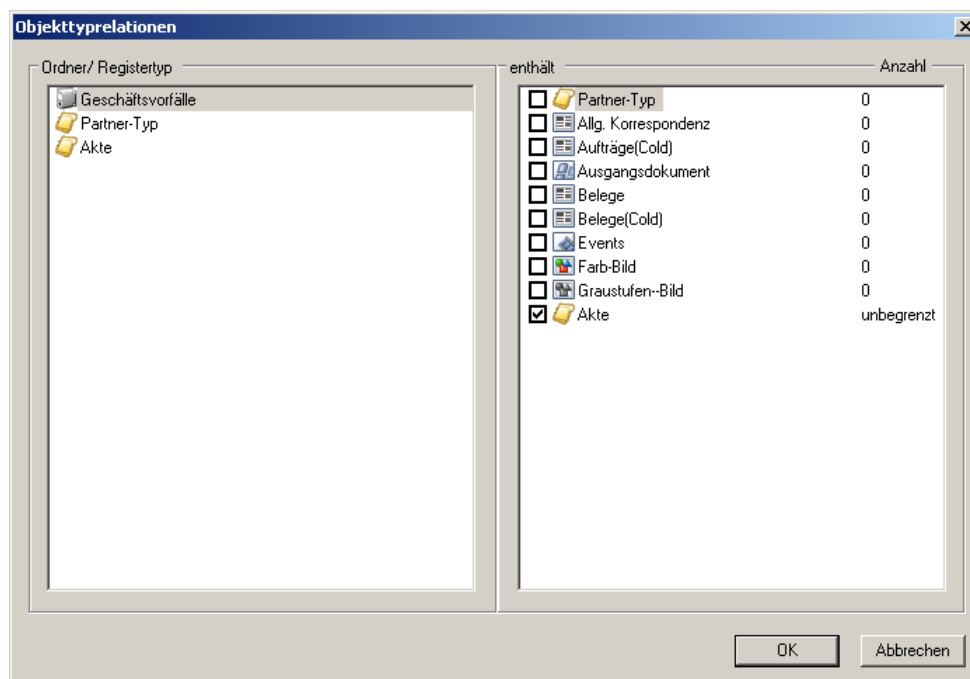


Abbildung 3: Objektrelation Ordner – andere Objekte

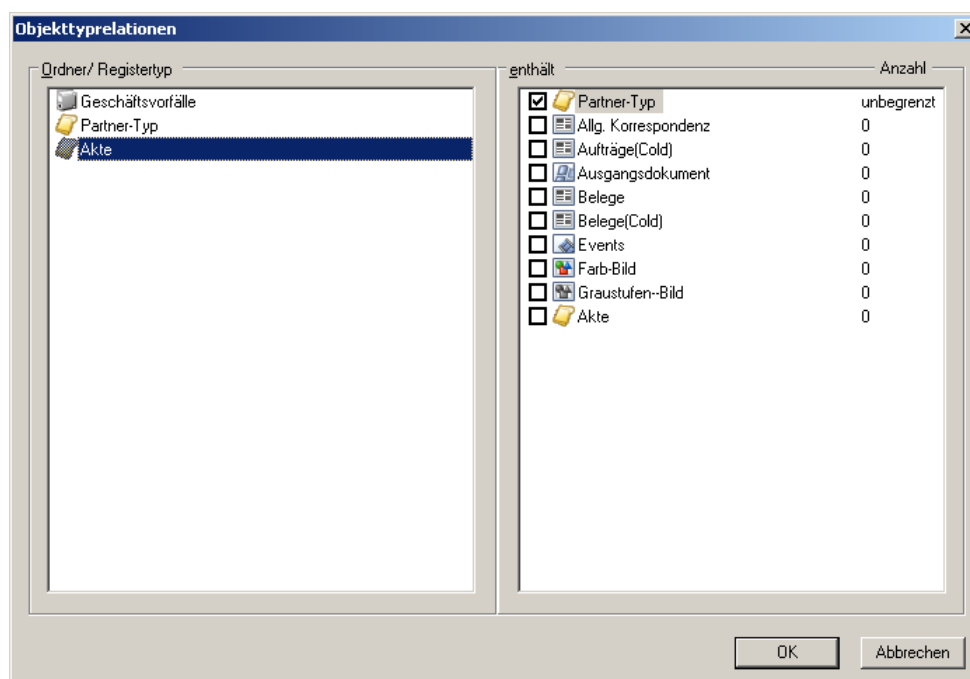


Abbildung 4: Objektrelation Register – Register

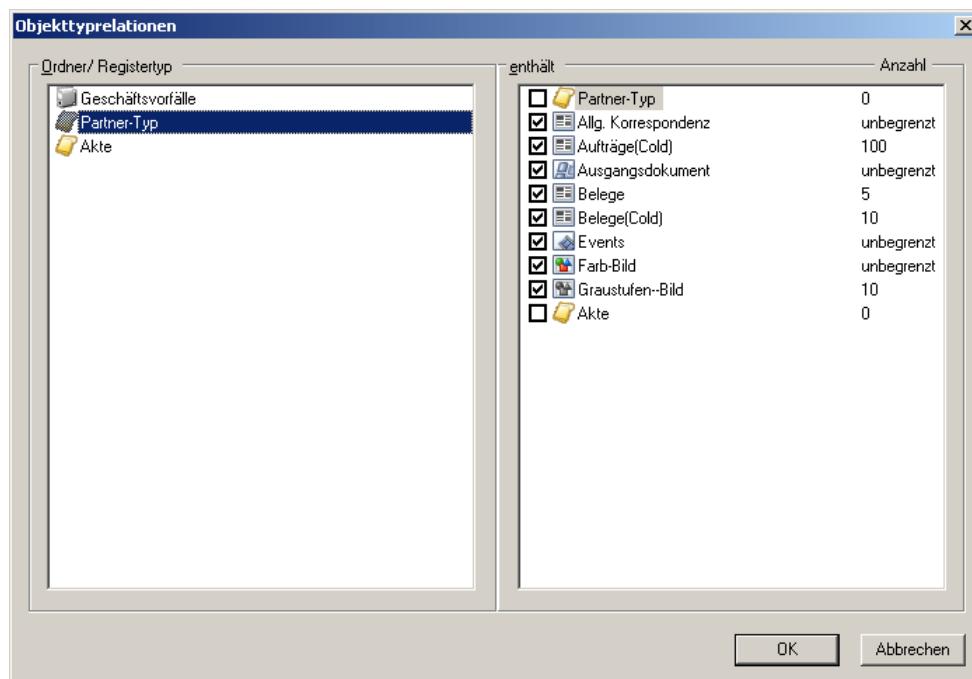


Abbildung 5: Objektrelation Register – Dokumente

Ohne Objektdefinition ist das enaio®-System wie eine leere Hülle, die noch keine Fächer für die zu verwaltenden Dokumente besitzt. Erst nachdem eine Objektdefinition angelegt wurde, kann sinnvoll mit ihm gearbeitet werden.

Objekttypen

enaio® unterscheidet folgende Dokumenten-Haupttypen:

| Nummer | Bezeichnung | Ablage |
|--------|------------------------|------------------------------------------------------------------------------------------------------|
| 00 | Ordner | In der Datenbank |
| 99 | Register | In der Datenbank |
| 01 | Grauwert-Modul (X) | Dateiendungen der einzelnen Seiten werden pro Dokument von 000 bis FFF hoch gezählt. |
| 02 | SW-Modul (D) | Dateiendungen der einzelnen Seiten werden pro Dokument von 000 bis FFF hoch gezählt. |
| 03 | Farbbild-Modul (P) | Dateiendungen der einzelnen Seiten werden pro Dokument von 000 bis FFF hoch gezählt. |
| 04 | Windows Quelldaten (W) | Je Dokument wird nur eine Datei angelegt und mit der korrekten Dateiendung gespeichert. |
| 05 | Film-Modul (M) | Dateiendungen der einzelnen Seiten werden pro Dokument von 000 bis FFF hoch gezählt. |
| 06 | E-Mails (Q) | Die Attachments (mehrere Seiten) werden pro Dokument gepackt und mit der Dateiendung 'MIM' abgelegt. |
| 07 | XML-Modul | Je Dokument wird nur eine Datei angelegt und mit der korrekten Dateiendung gespeichert. |

Tabelle 1: Dokumenten-Haupttypen

Obwohl Ordner und Register keine Dokumenttypen sind, werden sie in der Tabelle mit aufgeführt, weil der angegebene Haupttyp eine wichtige Rolle bei der Verwaltung der DMS-Objekte spielt.

Für jeden gewünschten Dokumenten-Haupttyp muss in einem Schrank jeweils mindestens eine Verschlagwortungsmaske angelegt werden. Wenn z.B. in einem Schrank Briefe im W-Modulformat .doc abgelegt werden sollen, dann muss eine Maske des Typs W-Modul (Haupttyp 04) angelegt werden. Untertypen werden durch zusätzliche Masken eingerichtet. Da der Untertyp in einem WORD, also 2 Byte großem Datenformat, gespeichert wird, können genau 64 K (2^{16}) DMS-Objekte je Haupttyp angelegt werden. Das gilt sowohl für Dokumente als auch für Ordner und Register.

Die Reihenfolge, in der diese Masken in der Objektdefinition erzeugt werden, bestimmt die Nummerierung der Dokument-Untertypen. Das erste Objekt eines Haupttyps erhält die Unternummer 0, das zweite die 1 usw.

Der Objekttyp wird durch ein DWORD beschrieben, bei dem im HIWORD der Haupttyp und im LOWORD der Untertyp des DMS-Objektes eingetragen wird. In verschiedenen Datenbanktabellen wird der Objekttyp, da aus ihm eindeutig Haupt- und Untertyp des DMS-Objektes bestimmt werden können (z. B. zweites angelegtes D-Dokument eines Haupttyps in Hexadezimal-Darstellung 0002 0001: 2 ist der Dokumenttyp, 1 bezieht sich auf das erste zusätzlich angelegte D-Dokument desselben Typs, in Dezimal-Darstellung: 131073).

Als weiteres Beispiel wird in Abbildung 6 die Tabelle `osobjparrel` gezeigt, in der die in den Abbildungen 3-5 dargestellten Objektrelationen gespeichert sind.

| obitype | parobitype | maxcount |
|---------|------------|----------|
| 196613 | 7 | 0 |
| 327682 | 7 | 0 |
| 131079 | 7 | 0 |
| 131078 | 7 | 0 |
| 262148 | 7 | 0 |
| 65541 | 7 | 0 |
| 131076 | 7 | 0 |
| 131077 | 7 | 0 |
| 6488079 | 6488077 | 0 |
| 131079 | 6488077 | 10 |
| 131078 | 6488077 | 5 |
| 65541 | 6488077 | 10 |
| 131077 | 6488077 | 100 |
| 6488077 | 6488077 | 0 |
| 196613 | 6488079 | 0 |
| 327682 | 6488079 | 0 |
| 6488079 | 6488079 | 0 |
| 131079 | 6488079 | 0 |
| 131078 | 6488079 | 0 |
| 262148 | 6488079 | 0 |
| 65541 | 6488079 | 0 |
| 131076 | 6488079 | 0 |
| 131077 | 6488079 | 0 |
| 6488077 | 7 | 0 |
| * | | |

Abbildung 6: Tabelle `osobjparrel` mit Objektrelationen

Die Objektrelationen werden in dieser Tabelle mit der Anzahl der maximal erlaubten Instanzen abgelegt (in den Klammern der Objekttyp in Hexadezimaldarstellung):

§ im Objekttyp 7 (Ordner 'Geschäftsvorfälle') können sich folgende Objekte nicht befinden:

| | |
|--------|----------------------------------------------|
| 196613 | (0x03 0005) P-Dokument |
| 327682 | (0x05 0002) M-Dokument |
| 131079 | (0x02 0007) D-Dokument (Belege(Cold)) |
| 131078 | (0x02 0006) D-Dokument (Belege) |
| 262148 | (0x04 0004) W-Dokument |
| 65541 | (0x01 0005) X-Dokument. |
| 131076 | (0x02 0004) D-Dokument (allg. Korrespondenz) |
| 131077 | (0x02 0005) D-Dokument (Aufträge(Cold)) |

- 6488077 (0x63 000D) Register (Belege) [99=0x63]
- § Im Objekttyp 6488077 (erstes Register 'Partner-Typ' in der Objektdefinition) können folgende Objekte nicht oder begrenzt angelegt werden:
- 131079 (0x02 0007) D-Dokument (Belege(Cold)) [10 Obj. erlaubt]
- 131078 (0x02 0006) D-Dokument (Belege) [5 Objekte erlaubt]
- 65541 (0x01 0005) X-Dokument [10 Objekte erlaubt]
- 131077 (0x02 0005) D-Dokument (Aufträge(Cold)) [100 Obj. erl.]
- 6488077 (0x63 000D) Register selbst [99=0x63]
- 6488079 (0x63 000F) Register (Akte) (99=0x63)
- § im Objekttyp 6488079 (zweites Register 'Akte' in der Objektdefinition) können folgende Objekte nicht angelegt werden:
- 196613 (0x03 0005) P-Dokument
- 327682 (0x05 0002) M-Dokument
- 131079 (0x02 0007) D-Dokument (Belege(Cold))
- 131078 (0x02 0006) D-Dokument (Belege)
- 262148 (0x04 0004) W-Dokument
- 65541 (0x01 0005) X-Dokument.
- 131076 (0x02 0004) D-Dokument (allg. Korrespondenz)
- 131077 (0x02 0005) D-Dokument (Aufträge(Cold))
- 6488079 (0x63 000F) dieses Register selbst [99=0x63]

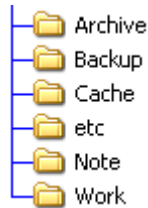
Für jeden DMS-Objekt-Untertyp wird eine eigene Datenbanktabelle eingerichtet. In dieser Tabelle wird die Verschlagwortung der konkreten Objekte (Ordner, Register oder Dokument), die mit dieser Verschlagwortungsmaske erzeugt wurden, gespeichert. Hier wird zwischen Ordnern, Registern und Dokumenten unterschieden (# steht für eine Nummer):

- § stamm# für Ordner,
- § register# für Register,
- § object# für Dokumente.

Zwischen Untertyp und dem Namen der Tabelle wird eine Verknüpfung über das Feld `tablename` der Tabelle `osobjdef` hergestellt. Jede dieser Tabellen besitzt ein Feld `id`, in dem ein datenbankweit eindeutiger Index (entnommen der Tabelle `osnextindex`) für jedes konkrete Objekt eingetragen wird. Dieser Index sowie der Objekttyp (Dokumenten- und -untertyp) erlauben die eindeutige Identifizierung und effektive Recherche nach den im enaio®-System abgelegten Objekten (Dokumenten, Ordnern, Registern usw.).

Medienstruktur

Die Dokumente werden nicht in der Datenbank, sondern in einem eigenen, speziell strukturierten Dateisystem abgelegt. Jede Servergruppe verfügt über ein eigenes Dateisystem. Dieses Dateisystem beginnt ab einem beliebigen Pfad mit den Verzeichnissen:



WORK

Im Verzeichnis WORK werden alle Dokumente gespeichert, die aktuell im System verwaltet werden und nicht archiviert sind. Auf dieses Verzeichnis greifen alle Anwendungsserver zu, wenn sie einem Clientprogramm ein Dokument zur Verfügung stellen sollen, das nicht archiviert ist, oder wenn ein neues Dokument im System abgelegt (importiert) wird. Der Pfad zum WORK-Bereich steht in der Datenbanktabelle `path` (bestehend aus Pfad\\WORK) mit Flag=0. Der WORK-Bereich selber hat folgende Struktur, in der Haupttyp, Untertyp und Dokumenten-ID (DocID) verarbeitet werden. Dabei werden im untersten Register von der DocID nur die beiden letzten Stellen angegeben.

Pfad\\WORK\\Haupttyp\\Untertyp\\DocID\\DocID.Seitenzahl

Pfad\\WORK\\Haupttyp\\Untertyp\\DocID\\DocID.ext

Die erste Schreibweise gilt für alle Pixelgrafik-Dokumenttypen (die Seitenzahl beginnt bei 000), die zweite Schreibweise für die W-Dokumente (z.B. Word), M-, Mail- und XML-Module (ext ist durch die jeweilige Datei-Erweiterung zu ersetzen, bei Word-Dokumenten also durch doc). Die DocID im Dateinamen ist hexadezimal achtestellig (vorne mit Nullen aufgefüllt).

In einer Multiserverarchitektur, wie sie mit Balanced Servern erzeugt werden kann, müssen alle Server auf dieses Verzeichnis mit seinen Unterverzeichnissen vollen Zugriff haben, d.h. dieses Verzeichnis muss für den Zugriff über das Netz freigegeben werden und sollte in der Tabelle `path` als UNC-Pfad eingetragen sein. Über einen UNC-Pfad kann direkt (ohne Laufwerksverknüpfung) auf eine beliebige Ressource im Netzwerk zugegriffen werden.

CACHE

Werden Dokumente archiviert, wird das Verzeichnis CACHE zum Zwischenspeichern für einen schnellen Dateizugriff verwendet. Das Verzeichnis CACHE wird vom ersten gestarteten Anwendungsserver parallel zu dem WORK-Verzeichnis angelegt (der Pfad wird aus der Tabelle `path` entnommen). Ein Anwendungsserver sieht zuerst in diesem Verzeichnis nach, ob er hier ein archiviertes Dokument findet. Wenn ja, wird davon ausgegangen, dass es aktuell ist, die Suche wird beendet und das Dokument der Anfrage zur Verfügung gestellt. Wenn nicht, dann wird das Dokument auf dem in der Datenbank hinterlegten Medium gesucht. Auch für das CACHE-Verzeichnis gilt, dass alle Server einer Gruppe darauf vollen Zugriff benötigen. Die Struktur der Dokumentenablage entspricht vollständig der Struktur des WORK-Verzeichnisses.

Besteht ein enaio®-System aus mehreren Servergruppen, dann spielt das CACHE-Verzeichnis auch ohne Archivierung eine Rolle. Versucht ein Clientprogramm auf ein Dokument zuzugreifen, dann ermittelt der Anwendungsserver aus dem Feld `medium_doc` der entsprechenden objekt#-Tabelle die ID des Mediums, auf dem das Dokument abgelegt wurde. Anschließend wird in der Tabelle `medien` die `server_id` (eigentlich `id` der Servergruppe) ermittelt, zu der dieses Medium gehört. Damit kann festgestellt werden, ob es sich bei dem Dokument um ein 'eigenes' Dokument oder um ein Dokument einer anderen Servergruppe handelt.

- § Handelt es sich um ein eigenes Dokument, so wird es zur Verfügung gestellt und der Job ist beendet.
- § Ist es aber ein Fremdobjekt, so wird als erstes unterschieden, ob es sich bei der Anfrage um die Bereitstellung eines Dokuments zum Betrachten oder zum Verändern handelt.
- § Soll das Dokument verändert werden, dann wird es über eine Anfrage an einen Server der entsprechenden Gruppe in das eigene WORK-Verzeichnis transportiert. Die Einträge in `medium_doc`, `medium_name`, `medium_dia` und `name_dia` der `objekt#`-Tabelle werden so geändert, dass sie auf den eigenen WORK-Bereich verweisen. Der Server der anderen Gruppe verschiebt Dokument und Dia in seinen CACHE-Bereich. Zum Abschluss stellt der Server das Dokument dem anfragenden Client im entsprechenden Client-Cache bereit.
- § Soll das Dokument nur angezeigt werden, dann wird im eigenen CACHE-Bereich nachgesehen, ob das Dokument vorhanden ist. Über einen Hash-Wert wird geprüft, ob das Dokument zwischenzeitlich verändert wurde. Ist das Dokument aktuell, wird es bereitgestellt. Kann das Dokument im eigenen CACHE-Bereich nicht gefunden werden oder ist es nicht aktuell, so wird es vom Server der anderen Gruppe angefordert und samt Dia im eigenen CACHE-Bereich abgelegt. Anschließend wird es zum Client übertragen.

Dieser Ablauf verringert die Datenübertragung zwischen den Servergruppen, was vor allem dann von Vorteil ist, wenn zwischen den Servergruppen nur eine langsame Verbindung zur Verfügung steht. Zu beachten ist, dass das zum Anschauen angeforderte Dokument nicht unbedingt aktuell sein muss. Da es zuerst im eigenen CACHE gesucht wird, kann es vorkommen, dass sich dort eine ältere Version als bei der Eigentümergruppe befindet. Dieser Nachteil kann kompensiert werden. Im Kontextmenü des enaio® client zu einem Dokument kann die Funktion 'Objektinformationen' (F8) aufgerufen werden. Angezeigt wird daraufhin das folgende Dialogfenster mit Informationen zum Dokument und der Möglichkeit, das Dokument im CACHE-Bereich zu löschen:

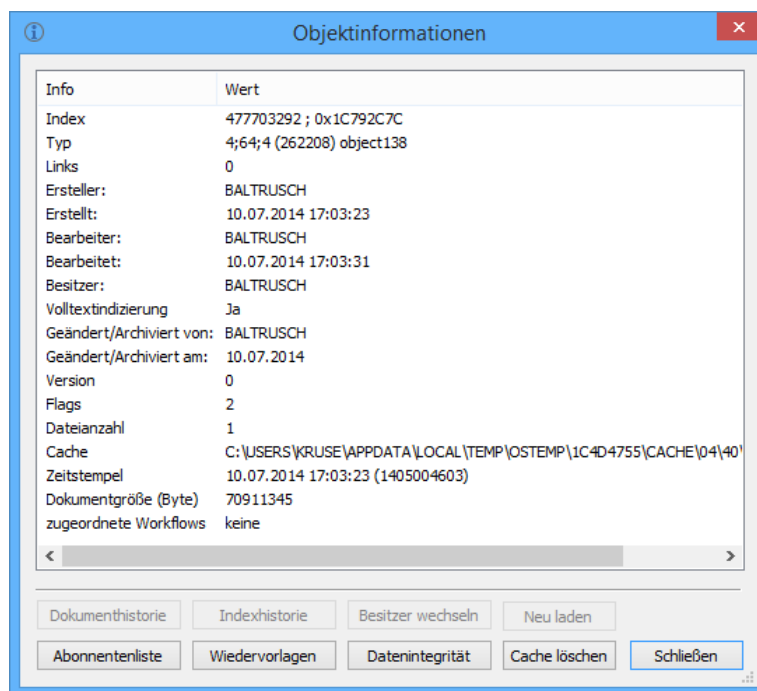


Abbildung 7: Löschen des CACHE über Dialog 'Objektinformationen'

Durch das Anklicken von 'Cache löschen' wird das Dokument im Servergruppen-CACHE gelöscht. Eine neue Anfrage führt dazu, dass die aktuellste Version vom Fremdserver in den CACHE-Bereich geholt und an den Client weitergeleitet wird.

ARCHIVE

Für die Archivierung wird parallel zum WORK-Verzeichnis auch das Verzeichnis ARCHIVE angelegt. Dieses Verzeichnis wird verwendet, um die Objektdefinition und alle Tabellen der Verschlagwortungen der archivierten Dokumente zu speichern. Eine detaillierte Beschreibung der Archivierung erfolgt im Kapitel 'Archivierung'.

Für das Archivieren, d.h. die dokumentenechte Speicherung auf speziellen Medien, wird auf den Archivmedien eine etwas andere Ablagestruktur erzeugt. Die Pfade zu den Archivmedien sind der Datenbanktabelle `path` (bestehend aus Pfad\\Archivname) mit Flag=1 zu entnehmen. Sie haben folgende Struktur:

```
Pfad\\Archivname\\Mediumname\\Haupttyp\\Untertyp\\Dokumententyp\\DocID\\DocID.Seitenzahl
```

```
Pfad\\Archivname\\Mediumname\\Haupttyp\\Untertyp\\Dokumententyp\\DocID\\DocID.ext
```

Das Unterverzeichnis 'Dokumententyp' kann nur zwei Werte annehmen:

- § '02' für das Dia des Dokuments und
- § '03' für das eigentliche Dokument.

Backup

Um die Datensicherheit bei der Archivierung zu erhöhen, kann bei der Konfiguration im enaio® enterprise-manager festgelegt werden, dass alle Dokumente zusätzlich zum Archivmedium auch in einem Verzeichnis BACKUP gespeichert werden.

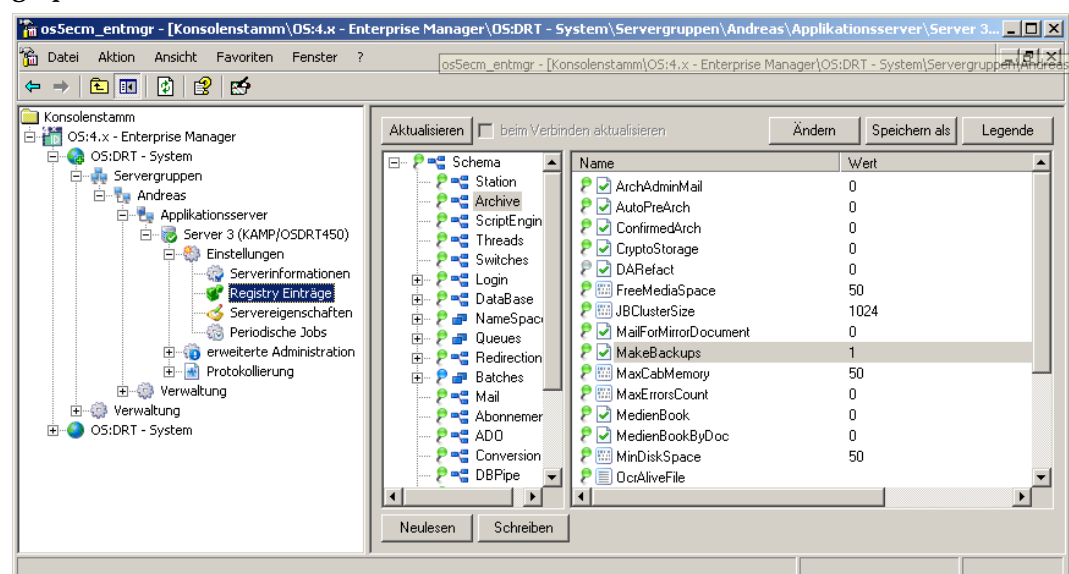


Abbildung 8: Einstellungen zur Archivierung

Dieses Verzeichnis wird (wenn gewünscht) von einem Anwendungsserver parallel zum Verzeichnis WORK angelegt. Die Struktur der Dokumentenablage ist identisch mit der Struktur, die auf den Archivmedien erzeugt werden.

NOTE

Parallel zum WORK-Verzeichnis wird ein Verzeichnis NOTE für Notizen zu Dokumenten angelegt. Notizen sind entweder Bemerkungen in Form von Textdateien, in denen unformatierter ASCII-Text enthalten ist, oder Verknüpfungen mit anderen DMS-Objekten. Die Verwaltung der Textdateien erfolgt über die Tabelle `remarks`, während die Verknüpfungen durch die Tabelle `hypInk` verwaltet werden. Jede Notiz erhält eine systemweit eindeutige ID, über die sie im Dateisystem der Notizen gefunden werden kann. Der Pfad zu den Notizen (Textdateien) steht in der Datenbanktabelle `path` mit `Flag=2`. Für die Verzeichnisstruktur der Notizen (Textdateien) werden Haupt- und Nebentyp des Dokumentes verwendet, zu dem die Notiz angefertigt wurde. Gemeinsam mit der `NoteID` ergibt dies einen Pfad, dessen Struktur analog der des WORK - Pfades ist:

```
Pfad\\NOTE\\Haupttyp\\Untertyp\\NoteID(mod 0x100)\\NoteID.txt
```

Bei der Installation jeder Servergruppe werden in der Tabelle `path` die Pfade zum WORK- (Flag=0) und zum NOTE-Verzeichnis (Flag=2) sowie die später konfigurierten Pfade zu den Archivmedien eingetragen. Hier erfolgt eine Zuordnung zu den Servergruppen (Feld `server_id`).

etc

Anders ist die Situation bei Konfigurationsdateien. Diese wurden unter `optimal_AS 3.60` im Verzeichnis `etc` abgelegt. In ihm wurden neben den Dateien der Objektdefinition (`asobjdef.dat` und `aslisten.dat`) auch alle anderen Konfigurationsdateien wie `as.cfg`, `asimpexp.cfg`, `ascold.cfg` usw. abgelegt. Seit der Vorgängerversion OS:DRT werden diese Dateien als Ressourcen betrachtet. Für die Ablage dieser Dateien wird das Verzeichnis `etc` parallel zum WORK-Verzeichnis angelegt und die Dateien mit vollständigem Dateinamen in die Tabelle `osresources` eingetragen.

Anwendungsserver

Der enaio®-Server ist als NT-Dienst implementiert. Er stellt in der 3-Tier-Architektur des enaio® die Business-Logik bereit. Die Clientprogramme (aus der Präsentationsschicht) greifen nicht direkt auf die Datenbasis zu, sondern stellen entsprechende Anfragen an den Anwendungsserver. Dieser ermittelt aus Datenbank, Dokumentenablage und Dokumentenarchiv (Datenschicht) die gewünschten Informationen und stellt sie dem anfragenden Client zur Verfügung. Dieser bereitet die gelieferten Informationen (Dokumente, Rechercheresultate etc.) auf und stellt sie in für den Anwender verständlicher Form dar.

Loadbalancing und Skalierbarkeit

Ein einfach aufgebautes enaio®-System besteht aus einer Datenbank, einem Anwendungsserver und einer Anzahl Clientprogramme. Je mehr Clientprogramme an dem Anwendungsserver arbeiten bzw. je komplexer die Anfragen an den Anwendungsserver werden, umso stärker wird der Server ausgelastet. Ab einem bestimmten Punkt reicht die Leistungsfähigkeit der eingesetzten Hardware oder die Performance des Anwendungsservers nicht mehr aus, um alle Anfragen hinreichend schnell bedienen zu können. Die Clientprogramme müssen warten, es kann zu Instabilitäten kommen. Eine Lösung dieses Problems aller vernetzten Systeme ist, die nun knapp gewordenen Ressourcen ausbauen zu können. Diese Fähigkeit wird als Skalierbarkeit bezeichnet. Je nach Anforderungen können mehr oder weniger Ressourcen bereitgestellt werden. Der Anwendungsserver von enaio® realisiert die Skalierbarkeit dadurch, dass 'einfach' ein weiterer Anwendungsserver

auf einem zusätzlichen Rechner zum DMS hinzugefügt und den Clientprogrammen mitgeteilt wird, dass sie sich nun auch mit diesem zusätzlichen Server verbinden können. Die Verteilung der Last wird 'Load Balancing' genannt, die Server selbst heißen 'Balanced Servers'. Wie schon gesagt, wird den Clientprogrammen mitgeteilt, welche Server für eine Verbindung zur Verfügung stehen. Diese Gruppe von Servern, auf die nun die Gesamtlast verteilt wird, wird 'Servergruppe' genannt.

Alle Server, die in einer Gruppe zusammengefasst sind, müssen die gleiche Funktionalität aufweisen, d.h. die gleichen Exekutoren geladen haben. Das hängt damit zusammen, dass beim Aufbau der Verbindung zu einer Serverinstanz nicht bekannt ist, welchen Funktionsumfang dieser bietet.

Es ist im laufenden Betrieb möglich, der Servergruppe neue Anwendungsserver hinzuzufügen und damit das Gesamtsystem zu skalieren. Es ist auch möglich, mehrere Anwendungsserver auf einem Rechner zu installieren, ausreichende Leistungsfähigkeit des Rechners vorausgesetzt.

Folgende Schritte sind beim Hinzufügen eines neuen Anwendungsservers erforderlich:

- § Installation einer neuen Anwendungsserver-Instanz in eine neue Servergruppe.
- § Bekanntgabe der neuen Anwendungsserver-Instanz an die Clients. Diese erfolgt durch Einträge in der Konfigurationsdatei `asinit.cfg` bei den installierten Clients.

Beispiel einer `asinit.cfg` zur Verbindung der Clientprogramme mit mehreren Servern in der Servergruppe 'Hauptgruppe':

```
[Archiv]
Comstring=Hauptgruppe
[Hauptgruppe]
SERVER0=dmsdata1#4010,30
SERVER1=dmsdata2#4020,40
SERVER2=dmsdata3#4030,30
```

Es wird eine Sektion mit der Bezeichnung angelegt, die im Eintrag `Comstring` vergeben wird. In dieser Gruppe werden alle Server, auf denen Anwendungsserver laufen, mit den entsprechenden Portnummern für die Kommunikation über TCP/IP aufgeführt. Danach wird eine Wahrscheinlichkeit angegeben, mit der ein Client sich mit diesem Server verbinden wird. Das ist für den Fall sinnvoll, dass die eingesetzten Server unterschiedlich ausgestattet sind und daher unterschiedliche Leistungsfähigkeit aufweisen.

Der Eintrag `dmsdata1` ist der Name eines Rechners oder dessen IP-Adresse im Netzwerk. Der Client verbindet sich mit diesem Rechner und versucht über den angegebenen Port eine TCP/IP-Verbindung zum Anwendungsserver auf diesem Rechner herzustellen.

In der Datenbank werden die Servergruppen und Server in den Tabellen `servergroup` und `server` eingetragen (siehe Abbildung 9 und Abbildung 10). Außerdem erfolgt die Zuordnung der WORK und NOTE-Verzeichnisse in den Tabellen `path` und `medien` zu den Servergruppen (siehe Abbildung 11 und Abbildung 12).

| id | name | description |
|----|-------------|---------------------|
| 3 | Hauptgruppe | erste Servergruppe |
| 7 | Nebenstelle | zweite Servergruppe |

Abbildung 9: Tabelle servergroup der Datenbank as400 einer enaio-Installation

| id | name | comstring | port | instance | flag | group_id | account_id |
|----|-----------------|-----------------|------|----------|------|----------|------------|
| 3 | RSCHROEDER-TEST | 195.127.121.122 | 4000 | astest2 | 1 | 3 | <NULL> |
| 7 | RSCHROEDER-TEST | 195.127.121.122 | 4020 | astest1 | 1 | 7 | <NULL> |

Abbildung 10: Tabelle server der Datenbank as400 einer enaio-Installation

| server_id | name | flag |
|-----------|--------------------------------|------|
| 3 | \\Rschroeder-test\\Data1\\WORK | 0 |
| 3 | \\Rschroeder-test\\Data1\\NOTE | 2 |
| 7 | \\Rschroeder-test\\Data2\\WORK | 0 |
| 7 | \\Rschroeder-test\\Data2\\NOTE | 2 |

Abbildung 11: Tabelle path der Datenbank as400 einer OS/4x-Installation

| id | name | batch | server_id | set_id | state | medorder | medsize | written | mirror_id |
|----|------|--------|-----------|--------|-------|----------|---------|---------|-----------|
| 4 | WORK | <NULL> | 3 | <NULL> | 3 | <NULL> | <NULL> | <NULL> | <NULL> |
| 5 | NOTE | <NULL> | 3 | <NULL> | 4 | <NULL> | <NULL> | <NULL> | <NULL> |
| 8 | WORK | <NULL> | 7 | <NULL> | 3 | <NULL> | <NULL> | <NULL> | <NULL> |
| 9 | NOTE | <NULL> | 7 | <NULL> | 4 | <NULL> | <NULL> | <NULL> | <NULL> |

Abbildung 12: Tabelle medien der Datenbank as400 einer enaio-Installation

Der erste Server einer Gruppe erhält die gleiche ID wie die Servergruppe. Alle weiteren Server erhalten ID entsprechend dem Zustand der Tabelle osnnextindex. Im Feld `name` wird der Rechnername eingetragen, unter `comstring` die IP-Adresse des Rechners, im Feld `port` die Nummer des für den Server zu verwendenden IP-Ports, unter `instance` wird der Dienstname eingetragen. Die Tabelle Abb. 13 zeigt zwei Servergruppen auf einem Rechner, was in der Praxis eher unwahrscheinlich, für Testzwecke aber recht nützlich ist. Der Name des Anwendungsservers aus einer Servergruppe mit dem sich die Clientanwendung verbindet wird z.B. im enaio® client in der Statuszeile angezeigt.

In der Tabelle `path` erfolgt die Zuordnung von WORK, NOTE und der Archivmedien (falls die Archivierung konfiguriert ist) zu den Servergruppen, nicht zu den Servern, auch wenn das Feld `server_id` etwas anderes suggeriert. Jeder Servergruppe werden ein WORK- und ein NOTE-Verzeichnis zugeordnet. Der Pfad dafür kann in der Eingabemaske 'Datenbank' im Eingabefeld 'Datenverzeichnis' festgelegt werden. Dabei ist zu beachten, dass alle Server einer Gruppe Zugriff auf das Datenverzeichnis haben müssen. Das Datenverzeichnis muss also für den Netzzugriff freigegeben sein. Am sichersten ist hier die Verwendung von UNC-Pfaden.

In der Tabelle `medien` erhalten die einzelnen Verzeichnisse oder Datenbereiche eigene IDs. Durch die Einträge der Verzeichnisse in die entsprechenden Datenbanktabellen können verteilte Architekturen mit mehreren Servergruppen an unterschiedlichen Standorten realisiert werden.

Damit können Dokumente, die in einer Servergruppe angelegt wurden, auch Clientprogrammen zur Verfügung gestellt werden, die an einem anderen Standort und damit mit anderen IP-Konfigurationen installiert sind.

Modularität

enaio®-Clients senden Job-Anforderungen an den Anwendungsserver-Kernel, der für die Verteilung der Jobs an die ausführenden Komponenten, die Verwaltung von Systemfunktionen und die Bereitstellung der Laufzeitumgebung zuständig ist. Anhand der Jobbezeichnung werden die Jobs in Queues gestellt. Jeder Engine ist eine Queue zugeordnet, d.h. in dieser Queue findet der Exekutor die ihn betreffenden Jobs.

Folgende Engines und Namespaces werden standardmäßig installiert:

| Engine | Namespace |
|--------------|----------------------------------|
| axsvckrn.exe | adm (Administrative Jobs). |
| | lic (Lizenz-Jobs). |
| | krn (Kernel-Jobs). |
| oxjobwfm.dll | wfm (Workflow-Jobs). |
| oxjobstd.dll | std (DMS-Jobs und Archivierung). |
| oxjobocr.dll | ocr (Zeichenerkennungs-Jobs). |
| oxjobabn.dll | abn (Abonnement-Funktion) |
| oxjobdms.dll | dms (weitere DMS-Jobs) |
| oxjobcnv.dll | cnv (Formatkonvertierungen) |
| oxjobmng.dll | mng (Jobs zur Systemverwaltung) |
| oxjobdbp.dll | dbp (Datenbankpiping) |
| oxjobado.dll | ado (DB Zugriff über ADO) |

Tabelle 2: Exekutors und Namespaces

Multithreading

Oft besitzen Anwendungen oder auch Dienste nur einen Thread – den, der mit dem Prozess verbunden ist. Das Betriebssystem Windows ist aber auch in der Lage, mehrere Threads je Anwendung oder Dienst zu unterstützen. Das Betriebssystem stellt jedem Prozess (Anwendung oder Dienst) eine bestimmte Rechenzeit zur Verfügung. Ist diese abgelaufen, so wird der aktuelle Prozess angehalten, der Zustand eingefroren und der nächste Prozess erhält für eine bestimmte Zeit die Möglichkeit, seine Aufgaben abzuarbeiten. Die jedem Prozess zugewiesene Zeit ist unabhängig von der Anzahl der im Prozess laufenden Threads. Für asynchrone Aufgaben ist die Einrichtung von Threads sehr sinnvoll. Während ein solcher Thread z.B. auf den asynchronen Empfang von Daten wartet, kann ein anderer Thread mit schon vorhandenen Informationen andere Aufgaben lösen. Das Warten auf den Eingang von Daten blockiert nicht den gesamten Prozess. Vor allem für Server, die von verschiedenen Clients angesprochen werden und die außerdem z.B. auf die Antwort von einer Datenbank warten müssen, würde wertvolle Zeit verloren gehen, wenn alle Aufgaben nur synchron abgearbeitet würden.

Die Multithreadarchitektur erlaubt, die Effektivität des Servers zu verbessern, unproduktive Wartezeiten bei langfristigen Operationen zu vermeiden und Reaktionszeit auf Clientanfragen zu verkürzen. Operationen, die mit möglichen Verzögerungen (wie Netzwerkverbindungen, Wechselwirkung mit der Konsole, geplante Aufgaben, Datenbank- und Dateisystemzugriffe usw.) verbunden sind, werden in separaten Threads ausgeführt.

Andererseits ist die Menge der parallel laufenden Threads festgelegt (wird durch Konfigurationsparameter bestimmt), wodurch der Aufwand für die Umschaltung

der Threadkontexte minimiert sowie der Verbrauch des physischen Speichers vermieden werden kann.

Der Server nutzt die folgenden 7 Threadtypen:

§ `HauptThread`

Übernahme der Konfigurationsparameter, Serverinitialisierung, Initialisierung externer Engines, Initialisierung weiterer Threads, Interthread-Kommunikation und -Synchronisation sowie Ressourcenfreigabe und Beenden des Servers.

§ `ConnectionThread`

Herstellung der Clientverbindung. Der Thread erzeugt den Socket, der im 'Listen-Modus' läuft; nach erfolgreichem Connect eines neuen Clientprogramms erzeugt der Thread die notwendigen Strukturen für die Clientverbindung und trägt sie in das Verzeichnis der aktiven Verbindungen ein.

§ `ReadThread`

Empfang der Clientanfragen, Syntaxprüfung, Einordnung der Anfrage in die Bearbeitungsqueues. Jeder Job wird in die Queue eingereiht, die dem jeweiligen Namensraum entspricht. Für jeweils 60 Clientverbindungen, die ein Server gleichzeitig bedienen soll wird ein weiterer Thread gestartet. Die maximale Anzahl der zugelassenen Clientverbindungen (`MaxConnections`) wird über den enaio® enterprise-manager festgelegt.

§ `JobThread`

Jobübernahme aus der Queue, Ermittlung der für die Abarbeitung zuständigen Engine, Übergabe des Clientaufrufes zur Bearbeitung an den Exekutor, Rückgabe der Resultate an das Clientprogramm und Auslösung von Fehlermeldungen. Mit jeder Jobqueue ist mindestens ein `JobThread` verbunden. Es können aber auch mehr sein, was über die Konfigurationsparameter festgelegt werden kann.

§ `SendThread`

Versendung von Notifications (Mitteilungen) an Clientprogramme über Serverereignisse, die nicht unmittelbar Resultat von Clientanfragen sind. Notifications werden nur an die Clientprogramme gesendet, die den Server über die Notwendigkeit solcher Mitteilungen informiert und einen speziellen Kanal dafür erzeugt haben.

§ `ConsoleThread`

Darstellung des aktuellen Zustandes und der vom Server ausgeführten Aktionen im Konsolenfenster, Annahme von Benutzerbefehlen und deren Übergabe an andere Threads.

§ `ExLoadThread`

Dynamische Änderung des im Server laufenden Satzes externer Exekutoren. Jeder externe Exekutor kann zur Laufzeit beliebig in den Server geladen oder aus dem Server entfernt werden. Diese Aktionen sind auch in der Konsole möglich (also auch bei der Fernwartung).

Die Multithreadarchitektur stellt hohe Forderungen an die Synchronisierung und Koordination der Operationen aller Threads.

Server-Jobs (das sind Anfragen von Clientprogrammen oder von anderen Servern) werden über sog. 'Queues', d.h. Warteschlangen, abgearbeitet. Queues erhalten ebenfalls Namen. Bei der Installation werden einige Standard-Queues automatisch angelegt (z.B. `common`, `dms`, `workflow`, `dbpipe`, `ocr`). Jede Queue hat einen oder mehrere Windows-Threads zur Verfügung, in denen die Jobs aus der Warteschlange bearbeitet werden.

Die Anzahl der Threads pro Queue kann vom Administrator konfiguriert werden. In einem Thread kann ein Job verarbeitet werden. Threads bearbeiten Jobs unabhängig voneinander. Die Abarbeitung von Jobs in eine Queue erfolgt nach dem 'Fifo'-Prinzip (Fifo = First in, first out), d.h. der Job, der als erster in die Queue gestellt wird, wird als erster einem freien Thread übergeben.

Eine Queue kann Jobs aus mehreren Namespaces verarbeiten. Die Namespaces der Engines können frei verteilt werden, ihre Einrichtung nimmt der Administrator nach Installation des Anwendungsservers vor. Bei der Installation von enaio® werden Standardvorgaben in der Registrierung eingetragen.

Folgende Grafik soll den internen Ablauf der Jobverarbeitung in einem Server darstellen:

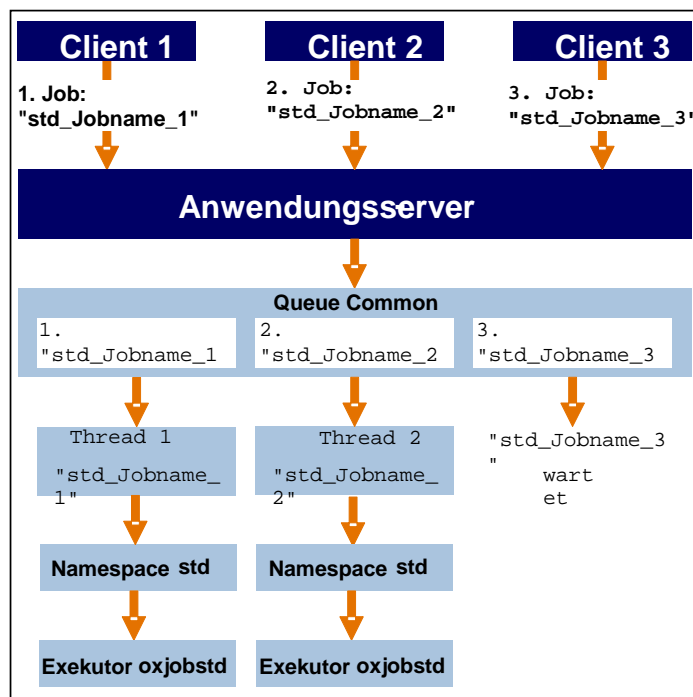


Abbildung 13: Beispiel für den Job-Flow in einem Anwendungsserver

Das Beispiel zeigt die Abarbeitung von drei Serverjobs, die von drei Clients an einem Anwendungsserver nacheinander übergeben wurden sind. Die Jobs werden im Namespace 'std' bearbeitet. Diesem Namespace sind der Exekutor 'oxjobstd.dll' und der Queue Common zugeordnet. Es stehen zwei Threads zur Verfügung. Da die drei Jobs sehr zeitnah übergeben werden, werden der erste und der zweite Job an Threads übergeben, der dritte Job dagegen muss auf einen freien Thread warten.

Clusterfähigkeit

Der Anwendungsserver kann auch auf Failover-Clustern betrieben werden. Bei einem Failover-Cluster übernimmt bei einem Ausfall des Dienstes ein anderer Rechner die Anwendung und bedient dann weiterhin die Clients. Dem Cluster wird per Konfiguration des Betriebssystems (z.B. Windows Server 2008) eine virtuelle IP-Adresse zugewiesen und dann auf die physischen IP-Adressen der im Cluster zusammengefassten Rechner umgesetzt. Welcher Rechner gerade angesprochen wird, hängt davon ab, welcher Dienst gerade durch das Betriebssystem aktiviert wurde. Dadurch ist es möglich, im Gegensatz zum Loadbalancing nur eine IP-Adresse im Connectstring anzugeben, nämlich die des Clusters.

[Archiv]
ComString=Cluster#4000

Dabei müssen auf den einzelnen Clusterrechnern die Anwendungsserverdienste unter dem gleichen Dienstenamen und dem gleichen Port eingerichtet werden. Die Clusterfähigkeit wird erreicht, indem die Serverdienste sich im System als genau ein Server darstellen. Wichtig ist, dass immer nur genau ein Dienst aktiv ist.

Damit sich beide Dienste auf den Clusterrechnern als ein Server für das System darstellen, muss in der Registrierung ein Rechnername eingetragen werden, der für beide Dienste identisch sein muss. Sinnvoll ist, den Rechnernamen des zuerst installierten Dienstes zu verwenden. Der Eintrag erfolgt unter:

HKLM\SOFTWARE\Optimal Systems\ASArchive2\Schema4\4.0\ComputerName

Anhand dieses Eintrages (zzgl. Dienstname und Portnummer) wird in der Datenbank ermittelt, ob der Server eingetragen ist. Fehlt der Eintrag, so wird automatisch der Rechnername verwendet, auf dem der Dienst gestartet wurde. Der Unterschied zwischen Cluster und Loadbalancing sieht so aus, dass beim Failover-Cluster immer nur ein Dienst aktiv ist, und damit für das System praktisch nur ein Dienst vorhanden ist, wobei beim Loadbalancing mehrere Dienste oder Server auf den Datenbereich zugreifen. Beim Loadbalancing erfolgt darüber hinaus eine Lastverteilung.

Session-Management

Clients ermitteln anhand der Datei `asinit.cfg` eine Serverinstanz, verbinden sich mit dieser und bleiben während der gesamten Session mit dieser Instanz verbunden. Beim Start eines Clients wird eine Session an einem der zur Verfügung stehenden Anwendungsserver erzeugt. Dabei erhält der Client eine Session-ID. Diese Session-ID dient dazu, Systemressourcen, z.B. Lizenzen und Systemdateien, einer Clientverbindung zuzuordnen und zu verwalten. Beim Beenden des Clients wird entweder explizit, d.h. über einen Job oder implizit, etwa bei Abbruch der Verbindung, die Session gelöscht und die verwendeten Systemressourcen wieder freigegeben.

Sessions werden in der Datenbank verwaltet und in der Tabelle `osession` gespeichert. Dazu wird zu jeder Session auch der Server vermerkt, der die entsprechende Session erzeugt hat. Ressourcen, die einer Session zugeordnet sind, werden in der Tabelle `oslockedres` während der gesamten Session gespeichert.

Alle Anwendungsserver tragen sich periodisch in einem Abstand von einer Minute in der Tabelle `ospingtable` ein und vermerken dort einen Zeitstempel ihres letzten Eintrages. Alle Server der Gruppe prüfen alle fünf Minuten, ob sich alle anderen Server eingetragen haben, d.h. ob die Dienste noch laufen. Wird dabei festgestellt, dass ein Dienst nicht mehr läuft, dann werden alle Sessions, die mit diesem Server verbunden waren, in der Tabelle `osession` gekennzeichnet, indem das Feld `serverid` auf 0 gesetzt wird.

Gleichzeitig wird in der Spalte `preservedterm` ein Zeitstempel für das Ende der Gültigkeit der bestehenden Sessions des beendeten Servers eingetragen. Der Zeitstempel wird als Summe aus der aktuellen Zeit und einem Timeoutintervall von drei Stunden berechnet. Alle Server prüfen periodisch, ob abgelaufene Sessions in der Tabelle vorhanden sind. Werden solche gefunden, dann werden die verbundenen Systemressourcen freigegeben und die Einträge aus der Session-Tabelle gelöscht.

Alle angegebenen Zeiten sind Standardeinstellungen und können über den enaio® enterprise-manager in der Registrierung des Servers geändert werden.

Weiter prüft jeder Server, ob beim Start des Dienstes noch Sessions aktiv sind, die für seine ID gültig waren. Auch hier setzt er im Falle, dass er solche Sessions findet, die `serverid` auf 0, da die physische Verbindung verloren ist und eine neue Verbindung geöffnet werden muss.

Versucht nun ein Client, sich mit einer abgelaufenen Session-ID zu verbinden, so erhält er eine entsprechende Fehlermeldung vom Server und muss eine neue Session generieren. Da bei der automatischen Freigabe der Systemressourcen der ursprüngliche Stand wiederhergestellt wird, gehen die zwischenzeitlichen Änderungen ggf. verloren.

Bei DBPIPING laufen alle Clientanfragen über den jeweiligen Anwendungsserver. Entsprechend hoch sind die Anforderungen an die Hardware, auf dem ein Anwendungsserver betrieben werden kann.

Clientprogramme

Entsprechend der 3-Tier-Architektur bilden die Clientprogramme die Präsentationsschicht, über die der Benutzer Zugriff auf das System hat, das System konfigurieren, nach Daten recherchieren und mit Dokumenten arbeiten kann. enaio® besitzt vier Clientprogramme für die Arbeit mit dem System:

- § enaio® editor für die Erstellung der Objektdefinition,
- § enaio® administrator, enaio® enterprise-manager, für die Konfiguration und Verwaltung des Systems,
- § enaio® client, für die Arbeit mit den Dokumenten und Daten des Systems,
- § enaio® capture, für die Erstellung und Abarbeitung von Folgen von Arbeitsschritten für den Import verschiedenartiger Dokumente.

Eine detaillierte Beschreibung dieser Module erfolgt in den jeweiligen Handbüchern.

Für die Systemarchitektur ist nur von Bedeutung, dass die Clientprogramme ihre Anfragen in Form von XML RPC-Jobs über TCP/IP an den jeweils für die Session kontaktierten Anwendungsserver richten und von dort ebenfalls über TCP/IP eine Antwort erhalten (Abbildung 13). Ein direkter Zugriff auf die Daten in der Datenbank und in das Dateisystem erfolgt nur durch den Server. Die Verwendung von TCP/IP hat zur Folge, dass für die Client-Server-Kommunikation keine Freigaben im Netz mehr erforderlich sind. Dies ist ein nicht unwesentlicher Beitrag für die Datensicherheit im enaio®-System.

In großen Netzen mit vielen Benutzern ist die Pflege der installierten Software eine große Aufgabe. enaio® bietet in Fortsetzung von optimal_AS, OS:DRT und enaio® die Installation von sogenannten Fileservern. Fileserver bedeutet hier, dass die Clientprogramme nur einmal im System auf einem Rechner installiert werden. Auf allen anderen Rechnern werden nur Netzininstallationen eingerichtet, bei denen der Zugriff auf die Clientprogramme über Verknüpfungen zum Fileserver erfolgt. Nur einige wenige Dateien bzw. Komponenten müssen noch lokal installiert werden. Die Pflege eines solchen Systems ist dann recht einfach. Bei Aktualisierungen der Clientprogramme reicht es meistens aus, die entsprechenden Dateien nur auf einem Rechner, dem Fileserver, auszutauschen.

Integrierter Workflow

Mit dem Workflowsystem wird das Produktspektrum von enaio® um eine prozessorientierte Komponente ergänzt. Mit ihr wird es dem Betreiber ermöglicht, Geschäftsprozesse elektronisch abzubilden und zu steuern. Insbesondere die enge

Kopplung an das DMS bietet erhebliche Optimierungspotentiale in der Bearbeitung von Geschäftsprozessen. Dokumente stehen an allen Bearbeitungsschritten für die berechtigten Anwender zur Verfügung und werden entsprechend dem jeweils festgelegten Prozessmodell automatisch nach der Bearbeitung an den nächsten Bearbeiter weitergeleitet.

Ausführlich wird das Workflowsystem im Systemhandbuch 'Workflow Management System' beschrieben.

Schnittstellen zu anderen Systemen

COM-Schnittstelle

COM steht für Component Object Model, das ist ein von Microsoft eingeführtes Modell zur Kommunikation zwischen Windowsanwendungen. Häufig wird dafür auch der Begriff OLE verwendet, das für Object Linking and Embedding steht. Hier wird ausschließlich der Begriff COM verwendet. enaio® nutzt als echte MS Windows-Anwendung die vom Betriebssystem bereitgestellten Funktionalitäten für die Interprozesskommunikation. Die dabei implementierten COM-Schnittstellen bilden die Basis für alle weiteren Schnittstellen zu Fremdsystemen wie Office Makros oder SAP-Link. Diese Schnittstellen werden wie üblich registriert und stehen dann anderen Anwendungen für die Kommunikation mit enaio® zur Verfügung. Eine genaue Beschreibung der COM-Schnittstelle finden Sie im entsprechenden Handbuch.

Office Makros

Über die enaio® office-utilities

Die enaio® office-utilities stellen über Makros Funktionen zur Verfügung, die die Arbeit mit W-Dokumenten erleichtern. W-Dokumente werden in externen Anwendungen bearbeitet, und müssen dazu ausgecheckt und nach der Bearbeitung wieder eingecheckt werden. Ausgecheckte W-Dokumente können von anderen Mitarbeitern nur schreibgeschützt geöffnet werden. So ist sichergestellt, dass nicht mehrere Mitarbeiter ein W-Dokument zur gleichen Zeit bearbeiten.

enaio® office-utilities erleichtern insbesondere das Ein- und Auschecken von W-Dokumenten und ermöglichen, aus externen Anwendungen direkt auf enaio®-Funktionen zuzugreifen.

Die Datenübernahme aus enaio® und enaio® editor-for-office sind ebenfalls in die enaio® office-utilities eingebunden.

enaio® office-utilities stehen für die folgenden Anwendungen zur Verfügung:

- § Microsoft Word
- § Microsoft Excel
- § Microsoft PowerPoint
- § Microsoft Project

Nicht alle Funktionen der enaio® office-utilities stehen für alle Anwendungen zur Verfügung.

Über die enaio®-Makros

Datenübernahme aus enaio®

Mit der Datenübernahme aus enaio® fügen Sie Verschlagwortungen oder den Inhalt von Dokumenten in ein MS-Word Dokument ein, das Sie in enaio® verwalten und bearbeiten. Sie können Daten direkt übernehmen, Bedingungen für die Datenübernahme formulieren und übernommene Daten umformatieren.

Das Makro für die Datenübernahme ersetzt Übernahmefelder, die Sie in das Word-Dokument eintragen, durch die entsprechenden Daten.

Die Übernahmefelder und Anweisungen für die Datenübernahme können Sie in MS-Word in das Dokument eintragen oder mit Hilfe des Feldeditors erstellen. Übernahmefelder und Anweisungen können auch in der Schablone einer W-Anwendung enthalten sein.

enaio® editor-for-office

Mit dem Feldeditor (enaio® editor-for-office) erstellen Sie Übernahmefelder und Anweisungen für die Datenübernahme aus enaio®. Die Übernahmefelder und Anweisungen fügen Sie in ein beliebiges MS Word-Dokument ein.

enaio® editor-for-office greift auf Daten aus der Objektdefinition - Bezeichnungen von Archivobjekten und Verschlagwortungsfeldern - zu und kann deshalb nur an einem Arbeitsplatz gestartet werden, an dem auch der enaio® client läuft.

enaio® editor-for-office starten Sie aus MS Word über das Menü enaio oder die enaio Symbolleiste. Über Registerkarten wählen Sie Feldtypen und geben die jeweils benötigten Parameter an. Der Feldeditor erzeugt daraus die Syntax für die Übernahmefelder.

Office- und Outlook-Add-Ins

Für Microsoft Office und Outlook stehen Add-Ins mit wenigen, aber zentralen Funktionen zur Verfügung, die an Arbeitsplätzen installiert werden können, die die komplexen und umfangreichen enaio® office-makros nicht benötigen. Diese Add-Ins werden über MSI-Pakete installiert.

Schnittstelle zu SAP

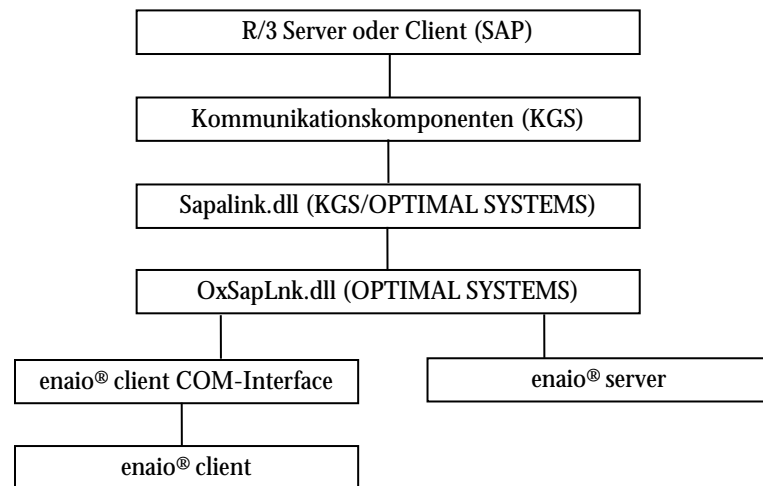
Einführung

Durch die Schnittstelle enaio®[R/3] werden die Systemwelten SAP und DMS auf Daten- und Anwendungsebene miteinander verbunden. Die Schnittstelle implementiert den ArchiveLink-Standard des SAP-R/3 Systems in der Version 3.1 und 4.5. Dadurch wird ermöglicht, dass ein bereits installiertes SAP R/3 System mit DMS-Funktionalitäten ausgerüstet wird, es also möglich ist, Dokumente abzulegen, zu recherchieren und Auswertungen über die Indexdaten zu ermöglichen. Dies macht insbesondere an solchen Arbeitsplätzen Sinn, wo neben den reinen im SAP-System gespeicherten Indexdaten auch zusätzlich noch auf die Belege in elektronischer Form zugegriffen werden soll. Um den ArchiveLink möglichst versionsunabhängig zu gestalten, wurden für die reine Schnittstelle zwischen der SAP-Welt und den DMS-Funktionalitäten mehrere Module der KGS Software-Gesellschaft für Systemanalyse und Engineering mbH eingesetzt.

Systembeschreibung

Architektur

Die Kommunikation zwischen den verschiedenen Komponenten des Gesamtsystems erfolgt über mehrere Schichten, wobei versucht wurde, für jede logische Schicht auch ein entsprechendes Modul einzusetzen, um die Schnittstellen klar voneinander trennen zu können. Folgende Abbildungen zeigen schematisch den modularen Zusammenhang zwischen den einzelnen Komponenten.



R/3 Server oder Client – je nach Anwendungsszenario kann der ArchiveLink von Server- oder Clientkomponenten eingesetzt werden. Davon abhängig werden unterschiedliche Funktionalitäten aufgerufen und verwendet. Insbesondere betrifft dies das Scannen und Anzeigen von Dokumenten.

Kommunikationskomponenten – stellen die abstrakte Schnittstelle zwischen den SAP-Anwendungsschichten und den ArchiveLink-Modulen dar.

Sapalink.dll – Schnittstellenbibliothek, die von den KGS Kommunikationskomponenten eingebunden wird. Aus dieser Bibliothek werden sämtliche Aufrufe an die oxsaplnk.dll weitergeleitet, in der dann die DMS-Funktionalitäten implementiert werden.

Oxsaplnk.dll – Implementierungsbibliothek der DMS-Funktionalitäten. Real werden hier Funktionsaufrufe entweder in Form von Jobs zum Archivserver geschickt oder Methoden des DMS-Clients zum Betrachten und Scannen von Dokumenten (Components) aufgerufen.

enaio® client COM-Interface – stellt eine Schnittstelle zum OLE-Automationsserver zur Verfügung, um von einer anderen Anwendung aus auf Funktionalitäten innerhalb des Clients zuzugreifen.

enaio® server – Bearbeitung der Jobs, Verwalten der Daten in der Datenbank und Verwalten der Dokumente im Dateisystem.

enaio® client – Komponenten zum Betrachten und Scannen von Dokumenten

Szenarien und Abläufe

Serverseitiger Zugriff:

Hier wird der ArchiveLink dazu verwendet, um größere Datenmengen, die innerhalb des SAP-Systems erzeugt werden, entgegenzunehmen, zu archivieren und bei Bedarf wieder abzuliefern. Das kann insbesondere Drucklisten und Reorg-Daten

betreffen. Wichtig ist hier, dass bei serverseitigem Zugriff niemals die Methoden zur Anzeige und zum Scannen von Dokumenten aufgerufen werden und somit auch keine Schnittstelle zum DMS-Client aufgebaut werden muss, die teilweise asynchron über Prozessgrenzen hin erfolgt und von daher für einen Serverzugriff nicht geeignet ist.

Clientseitiger Zugriff

Vom SAP-Client aus wird vorwiegend auf Dokumente zugegriffen, um diese zu scannen bzw. zu betrachten. Dabei erfolgt die Verwaltung der Indexdaten regelmäßig im SAP-System, im DMS werden die Adressen der Dokumente bzw. Komponenten gespeichert. Bei einem Aufruf zum Scannen wird der Schnittstelle die ID einer Komponente übergeben, zu der eine Seite hinzugefügt werden soll. Der Aufruf zum Scannen erfolgt synchron, d.h. die Funktion kehrt erst dann zurück, wenn das Dokument gescannt und abgelegt wurde oder wenn ein Fehler aufgetreten ist. Bei der Anzeige von Komponenten gilt dies nicht, es wird entweder der interne Viewer geöffnet oder eine entsprechende Anwendung aufgerufen, die in der Lage ist, das gewünschte Dokument anzuzeigen.

Initialisierung und Bearbeitung der Methoden

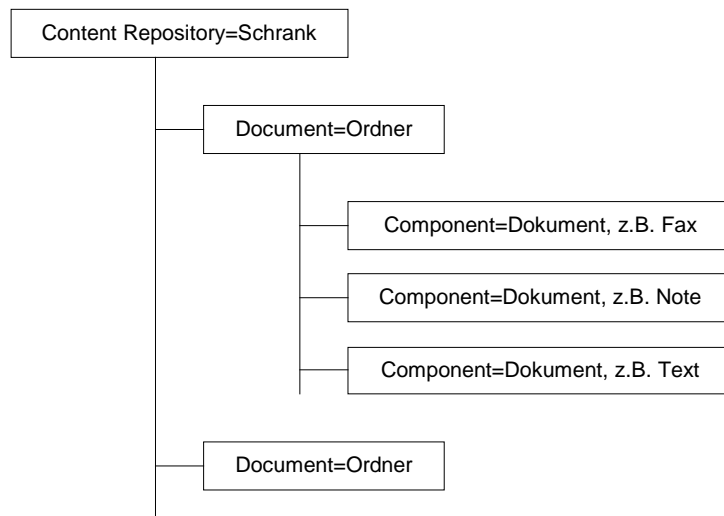
Aufgrund der unterschiedlichen Szenarien wird es im Modul oxsaplnk.dll einen Schalter geben, über den eingestellt wird, ob das Modul im Server- oder Clientmodus läuft. Wird ermittelt, dass die DLL im Kontext eines Servers läuft, dann wird das Client-COM-Interface nicht initialisiert und die Methoden zum Scannen und Anzeigen von Komponenten liefern einen Fehler. Unabhängig davon wird das COM-Interface des Clients erst dann initialisiert, wenn zum ersten Male ein Dokument angezeigt bzw. gescannt werden soll.

Alle Methodenaufrufe, die nicht mit der Initialisierung, dem Scannen und Anzeigen von Dokumenten zusammenhängen, werden als Job verpackt und zur Bearbeitung an den Archivserver weitergeleitet. Damit verbunden sind zwei Vorteile: erstens wird auf der ArchiveLink-Schnittstelle keine Datenbankverbindung zum DMS-Datenbank benötigt und zweitens ist es nicht notwendig, dass Archivserver und ArchiveLink auf einem Rechner laufen, sondern es ist ausreichend, wenn die beiden Komponenten mittels eines TCP-IP Netzwerkes verbunden sind. Am Archivserver werden nun die Jobs bearbeitet und die Ergebnisse wiederum auf TCP-Ebene zurückgesandt. Besteht das Ergebnis oder die Eingabe einer Methode in einer Datei, so kann diese entweder per TCP übertragen werden oder per Verweis auf einen zugreifbaren Plattenbereich übermittelt werden.

Datenmodell

Für den Einsatz des ArchiveLink ist es notwendig, eine entsprechend vorbereitete Objektdefinition zu verwenden. Dabei gibt es pro Content Repository genau einen Schrank, in dem alle zu dieser logischen Archiveinheit anfallende Dokumente und Komponenten abgelegt werden.

| R/3 - SAPALINK-Bezeichnung | enaio® - Bezeichnung |
|------------------------------|--------------------------------------------------------|
| Content Repository (z.B. FI) | Schrank (z.B. SAP-FI) |
| Document | Ordner |
| Component | Dokument (z.B. Scanned Components, Default Components) |



Die SAP-Dokumente werden also mit Ordnern abgebildet und enthalten dann die einzelnen Komponenten der Dokumente. Die Indexdaten sind hier den Attributen der SAP-Dokumente entnommen:

| SAPALINK-Konstante | SAPALINK-Feldbedeutung | enaio®-Feld | Datentyp | Bemerkung |
|-------------------------|------------------------------------------|------------------------|-----------|-----------|
| docID | SAP Dokument-ID | DocID | char(256) | Indexfeld |
| DOC_PROPERTY_REPOSITORY | Repository to which the document belongs | Content Repository | char(32) | |
| DOC_PROPERTY_ADATE | Document creation date | Creation Date | char(32) | |
| DOC_PROPERTY_ETIME | Document creation time | Creation Time | char(32) | |
| DOC_PROPERTY_MDATE | Document modification date | Date last modification | char(32) | |
| DOC_PROPERTY_MTIME | Document modification time | Time last modification | char(32) | |
| DOC_PROPERTY_PROTECTION | Document protection flag | Document protection | char(32) | |
| DOC_PROPERTY_ALVERSION | Archive Link Version | ArchiveLink Version | char(32) | |
| | n.v. | Barcode | char(256) | |
| | n.v. | Barcode sent to R/3 | Boolean | |

Jede Komponente besitzt einen Contenttype. Anhand dieses Contenttypes wird in der 'oxsaplnk.dll' mittels der Zuordnung in der Konfigurationsdatei 'oxsaplnk.cfg' entschieden, welcher Dokumenttyp im ECM verwendet wird, um den Inhalt der Komponente aufzunehmen. Ist der Contenttyp unbekannt, d.h. es existiert keine

Entsprechung in der Konfigurationsdatei, so wird die Komponente einem Default-W-Dokumenttyp zugeordnet, dem jedoch keine Anwendung zugeordnet ist, sodass bei Aufruf des Viewers die unter Windows verknüpfte Anwendung gestartet wird.

Die Indexdaten aller Komponententypen sind gleich und beinhalten alle durch die SAPALINK.DLL vorgegebenen Attribute. Weitere Indexfelder sind für diese Dokumenttypen nicht vorgesehen, sondern sollen dann über andere Schränke modelliert werden, so dass dann auch vom DMS-Client aus in den SAP-Belegen recherchiert werden kann.

| SAPALINK-Konstante | SAPALINK-Feldbedeutung | enaio®-Feld | Datentyp | Bemerkung |
|---------------------------|-----------------------------|------------------------|-----------|-----------|
| compID | SAP Komponenten-ID | Component_ID | char(256) | Indexfeld |
| COMP_PROPERTY_FILENAME | original filename | FileName | char(256) | |
| COMP_PROPERTY_CONTENTTYPE | Contenttype | Content Type | char(256) | |
| COMP_PROPERTY_ADATE | Component creation date | Creation Date | char(32) | |
| COMP_PROPERTY_ETIME | Component creation time | Creation Time | char(32) | |
| COMP_PROPERTY_MDATE | Component modification date | Date last modification | char(32) | |
| COMP_PROPERTY_ETIME | Component modification time | Time last modification | char(32) | |
| COMP_PROPERTY_CHARSET | Charset | Charset | char(32) | |
| COMP_PROPERTY_ALVERSION | ArchiveLink-Version | ArchiveLink Version | char(32) | |
| COMP_PROPERTY_VERSION | Application Version | Application Version | char(32) | |

enaio® winapp

enaio® winapp dient der Anbindung von Fremdanwendung zur Durchführung von enaio®-relevanten Aktionen. Im Vordergrund steht dabei das SAP-GUI mit der Möglichkeit einer Recherche über den Datenbestand des enaio®.

Als Screenscraping-Anwendung kann enaio® winapp aber auch mit anderen Anwendungen zusammenarbeiten, Daten aus diesen Anwendungen übernehmen und konfigurierte Aktionen mit diesen Daten ausführen.

Der angebotenen Fremdanwendung können keine Daten rückübermittelt werden.

enaio® winapp wird bei der Installation in das Verzeichnis `\clients\client32` kopiert. Die Konfiguration erfolgt über die Konfigurationsdatei `axvbWinApp.ini`.

Eingebunden werden können Konverter-Bibliotheken, die übernommenen Daten so aufbereiten und abgleichen, dass diese in einer Recherche über den Datenbestand zu eindeutigen Ergebnissen führen.

Lizenzen für die Module 'OLS' und 'ASC' werden am Arbeitsplatz beansprucht.

enaio® file-system-archiver

Der enaio® file-system-archiver ermöglicht es, aus dem Dateisystem komplette Verzeichnisse mit dem gesamten Inhalt in enaio® zu übertragen.

Verzeichnisse und Dokumente werden enaio®-Objekten, Register- und Dokumententypen, zugeordnet. Alle Kenndaten der Verzeichnisse und Dokumente können als Verschlagwortungsdaten in enaio® übertragen werden.

Der enaio® file-system-archiver, `axfsarch.exe`, wird bei der Installation in das Verzeichnis `clients\client32` kopiert. Benutzer, die den enaio® file-system-archiver benutzen, benötigen die Lizenz 'FSA'. Besondere Systemrollen sind nicht notwendig.

Für die Konfiguration wird enaio® file-system-archiver mit dem Parameter `-config` gestartet. Die Konfigurationsdaten werden in der Konfigurationsdatei `axfsarchconfig.xml` im Anwendungsverzeichnis gespeichert. Sollen diese Konfigurationsdaten stattdessen im zentralen Verzeichnis `\etc` des Datenverzeichnisses verwaltet werden, wird beim Aufruf der Parameter `-serverConfig` ergänzt. Werden mehrere Server verwendet, müssen die Konfigurationsdaten in alle entsprechenden etc-Verzeichnisse kopiert werden.

Ein Verweis auf den enaio® file-system-archiver kann in das Kontextmenü des 'Senden an'-Menüs eingebunden werden. So können die gewünschten Daten an enaio® übergeben werden.

Details befinden sich im Handbuch 'enaio® file-system-archiver'.

enaio® communicator

Der enaio® communicator ist eine Plattform für das Anbinden externer Systeme an enaio®. Über den enaio® communicator können regelbasiert Importe, Exporte und Transformationen von Daten ausgeführt werden. Dazu können Komponenten des enaio® communicator auf Datenbanken, Dateien und Dienste zugreifen.

Der enaio® communicator besteht aus dem Kernel, Komponenten für das Messaging, Logging und Konfiguration und aus den Kommunikationskomponenten. Es gibt zwei Klassen von Kommunikationskomponenten, die Konnektoren und Transformer.

Konnektoren

Konnektoren verbinden sich mit externen Systemen:

- § Periodisch (receive -> publish), d.h. die Komponente holt Daten und publiziert diese an das interne Messaging-Modul.
- § Ereignisgesteuert (subscribe -> send), d.h. die Komponente bekommt Daten (hat diesen Topic also abonniert) und übergibt diese Daten einem externen System.

Mit jeweils unterschiedlichen Konnektoren kann auf Daten z.B. per TCP/IP, aus dem Dateisystem, aus Datenbanken, per SOAP zugegriffen werden.

Konnektoren stehen jeweils am Anfang und Ende einer Kommunikationsstrecke. Sie bieten Funktionen zur Steuerung des Verbindungszustandes (connection state management).

Transformer

Transformer implementieren spezielle Transformationsregeln.

Beispielsweise kann ein Transformer HL7-Daten oder CSV-Daten in ein XML-Format transformieren. Ein Transformer kann mehrere Topics abonnieren bzw. publizieren (vgl. Publish-Subscribe).

Transformer können auf unterschiedliche Weise implementiert werden.

§ C-DLL

§ XSLT

§ COM (beispielsweise in Visual Basic)

Transformer sollen möglichst nicht auf Systemfunktionen zugreifen. Für regelbasierte Validierungen ist es jedoch teilweise notwendig, Datenbankabfragen an die Archivdatenbank zu stellen.

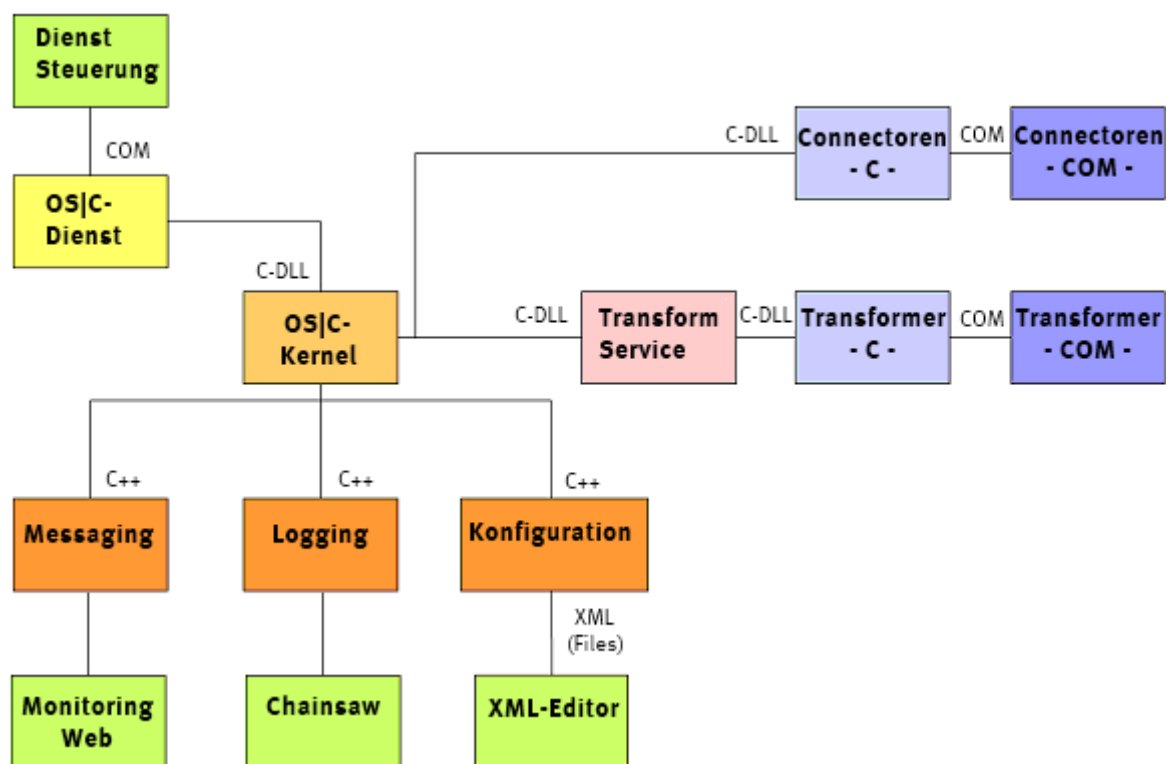
Publish-Subscribe

Publish-Subscribe Systeme dienen zur effektiven Verteilung von Information und Nachrichten in verschiedensten Anwendungen.

Typischerweise wird eine große Anzahl von Nachrichten (messages) über einen oder mehrere Server (enaio® communicator) an interessierte Nutzer (Konnektoren bzw. Transformatoren) verteilt. Aus der großen Zahl publizierter Nachrichten können die Clients durch ein Abonnement (Subscription) festlegen, an welcher Art von Information sie Interesse haben.

Dieses Prinzip hat den Vorteil, dass sich Absender und Empfänger gegenseitig nicht kennen und somit technisch voneinander unabhängig sind (loose coupling).

Architektur:



Der enaio® communicator arbeitet als Windows-Dienst. Für das Starten und Stoppen wird die Dienststeuerung von Windows oder der enaio® communicator-Monitor verwendet werden.

Einige Komponenten, insbesondere Komponenten, die Funktionalitäten des enaio® server verwenden, müssen lizenziert werden.

Der enaio® communicator kann über das Setup von enaio® installiert werden.

Optionale Komponenten

Neben den Kernkomponenten und Schnittstellen verfügt enaio® über eine stetig wachsende Anzahl optionaler Komponenten, die das System erweitern, indem sie das Einbinden externer Anwendungen und Funktionen erlauben.

Anzeigedienste

Anzeigedienste sind enaio®-Kerndienste zur flexiblen Anzeige von Dokumenten und Dokument- sowie Indexdaten. Weitere Kerndienste sind enaio® fulltext für die Volltextindizierung und enaio® webservice für die Anbindung externer Anwendungen an enaio®.

Die Anzeigedienste können in enaio® client, enaio® webclient oder externen Anwendungen, wie zum Beispiel Microsoft Outlook, eingebunden werden.

Zu den Anzeigediensten gehören:

- § enaio® gateway
- § enaio® documentviewer
- § enaio® detailsviewer
- § enaio® contentviewer

enaio® gateway

enaio® gateway ist ein Proxy, der als Kommunikationsschnittstelle zwischen den Kerndiensten eingesetzt wird.

enaio® gateway ermöglicht die Anzeige der Inhalts- und Detailvorschau in enaio® client und die Kommunikation und Authentifizierung der Kerndienste sowie den Betrieb von enaio® webclient.

enaio® documentviewer

enaio® documentviewer bindet die Vorschau eines markierten enaio®-Dokuments in enaio® client ein. Der Inhalt von markierten Ordnern oder Registern bzw. die Vorschau eines markierten Objekts kann auch in enaio® webclient angezeigt werden. Dabei stellt enaio® documentviewer einfache Funktionen zum Betrachten, Navigieren und Suchen zur Verfügung.

enaio® documentviewer bietet dank der integrierten Komponenten enaio® renditionplus und Rendition-Cache auch die Möglichkeit zur Konvertierung von Dateien in andere Dateiformate und zur Erkennung von Text in Bilddateien.

enaio® documentviewer besteht aus folgenden Komponenten:

§ Webanwendung

Die Webanwendung organisiert die Anzeige der Dokumentvorschauen in enaio® client.

§ Konvertierungskomponente enaio® renditionplus

enaio® renditionplus erzeugt aus Dokumenten Renditions (Bilder, PDFs, TIFF, Text, Miniaturansichten usw.) und bietet standardmäßig die Möglichkeit der Texterkennung mit Abby FineReader.

§ Speicherkomponente Rendition-Cache

Rendition-Cache ist ein Zwischenspeicher, der erzeugte Renditions zentral verwaltet. Pro Dokument wird nur eine Vorschau erzeugt. Wenn ein und dasselbe Dokument mehrfach versendet oder ein Verweisdokument erstellt wird, verwendet enaio® documentviewer die Vorschau aus dem Rendition-Cache wieder.

enaio® documentviewer ermöglicht es, dass beim Senden eines Dokuments an einen internen Empfänger ein Verweis auf die Vorschaudatei in die E-Mail eingebunden sowie die Vorschau bei enaio® webclient aktiviert wird. Zusätzlich kann eine Miniaturvorschau der ersten Seite eines Dokuments in den E-Mail-Body eingefügt werden.

enaio® detailsviewer

enaio® detailsviewer ermöglicht den flexiblen HTTP-Zugriff auf Index- und Dokumentendaten. Mit enaio® detailsviewer können beispielsweise Indexdaten von markierten enaio®-Objekten in enaio® client und anderen externen Anwendungen angezeigt werden.

enaio® detailsviewer wird über die enaio®-Komponente enaio® appconnector realisiert. enaio® appconnector ist eine REST-Schnittstelle, die mit einer bestimmten Konfiguration als enaio® detailsviewer betrieben werden kann. Alternativ kann enaio® appconnector als Schnittstelle zu mobilen Anwendungen dienen und als Kommunikationskomponente den strukturierten Zugriff auf enaio® server ermöglichen. In letztgenannten Einsatzszenarien muss enaio® appconnector kostenpflichtig erworben werden.

enaio® detailsviewer kann im Projekt visuell angepasst werden. Wenden Sie sich in diesem Fall an das Consulting-Team von OPTIMAL SYSTEMS.

enaio® contentviewer

enaio® contentviewer ist eine Rahmenanwendung, welche enaio® documentviewer mit enaio® detailsviewer kombiniert und dadurch eine zusammengefasste Ansicht beider Dienste in enaio® webclient ermöglicht.

Für Schnittstellenprojekte kann enaio® contentviewer auch anderweitig eingebunden werden. Wenden Sie sich dafür an das Consulting-Team von OPTIMAL SYSTEMS.

enaio® appconnector

enaio® appconnector ist ein Kerndienst von enaio®.

enaio® appconnector ist eine REST-Schnittstelle (Representational State Transfer), die den ressourcenorientierten, flexiblen HTTP-Zugriff auf Index- und Dokumentendaten in enaio® ermöglicht.

enaio® appconnector dient somit beispielsweise:

- § als Schnittstelle zu mobilen Anwendungen, wie den enaio® apps für Android-, iOS- und Windows-basierte Smartphones und Tablets
- § als enaio® detailsviewer zur Anzeige von Indexdaten im Info-Fenster von enaio® client und anderen externen Anwendungen

Mit enaio® appconnector können Droptargets in enaio® eingebunden werden.

Droptargets sind eine Möglichkeit, um Daten und Dokumente über enaio® appconnector in das enaio®-System zu importieren. Gleichzeitig ist es möglich, mit einem Droptarget verschiedene Aktionen im System auszulösen, wie etwa das Starten von Workflows und das Erstellen oder Aktualisieren von DMS-Objekten.

Des Weiteren ist der Pushnotificationsservice (PnS) Bestandteil von enaio® appconnector und wird für die Zustellung von Nachrichten eingesetzt.

Der PnS prüft regelmäßig, ob neue Nachrichten über Abonnements, Wiedervorlagen oder Workflowarbeitsschritte vorliegen. Die App zeigt dann eine

Meldung über jede neu eingegangene Nachricht an, auch wenn die App nicht aktiv ist. Die Anzeige der Meldung können Benutzer in der App individuell einstellen.
Der enaio® appconnector erfordert eine Lizenzierung des Moduls 'APP'.

enaio® pdfa-dispatcher

Der enaio® pdfa-dispatcher erzeugt aus enaio®-Dokumenten mit Bild- oder PDF-Dateien, Dokumente mit PDF/A-Dateien. Die in PDF/A konvertierten Dateien können entweder die Ausgangsdateien ersetzen, als Varianten den Ausgangsdokumenten zugeordnet werden oder als neues Dokument in enaio® gespeichert werden.

Der enaio® pdfa-dispatcher kann ebenfalls eingesetzt werden, um Bild- oder PDF-Dateien von enaio® Dokumenten in PDF/A-Dateien zu konvertiert und zu exportieren.

Dabei kann eine OCR eingeschaltet werden, deren Ergebnis als 'Hidden Text' mit in die Datei übernommen wird.

Die Konvertierung erfolgt entweder über den LuraDokument PDF Compressor oder über die an den enaio® server angebundene Konvertierung.

Die Komponenten für den enaio® pdfa-dispatcher werden bei der Installation des enaio® server in das Serververzeichnis kopiert. Eingebunden wird der enaio® pdfa-dispatcher als Dienst, der regelmäßig Konvertierungen steuert.

Eine Komponente kann als externes Programm in den enaio® client eingebunden werden. Somit können Benutzer Dokumente an den enaio® pdfa-dispatcher senden um eine PDF/A-Datei zu erstellen.

Der enaio® pdfa-dispatcher erfordert eine Lizenzierung des Moduls 'CLE'.

enaio® mail-archiver

Der enaio® mail-archiver ist ein SMTP Relay-Server, über den E-Mails strukturiert in enaio® übernommen werden können und über den mit den E-Mail-Daten Workflowprozesse gestartet werden können.

Die übernommenen E-Mails können, wie alle anderen enaio®-Dokumente, dokumentenecht archiviert werden.

Der enaio® mail-archiver kann so eingesetzt werden, dass er E-Mails übernimmt, Anhänge herauslöst, in enaio® speichert, in den E-Mails durch Verweise ersetzt und dann die E-Mails an ein Groupware System weiterleitet.

Der enaio® mail-archiver kann flexibel in unterschiedlichen Umgebungen eingesetzt werden. Er kann als SMTP Server, als POP3-Client oder als BCC Host eingebunden werden.

Die Komponenten für die Installation des enaio® mail-archiver befinden sich unter den Installationsdaten im Verzeichnis ...\\disk1\\components\\OSMAILARCHIVER.

Der enaio® mail-archiver wird als Dienst installiert und benötigt die Lizenzierung des Moduls 'MAR'.

Die Konfiguration erfolgt über die Konfigurationsdateien config.xml, rules.xml, std-archive.xml.

Datensicherheit

Unterstützung durch die Systemarchitektur

Für die Client-Server-Kommunikation wird TCP/IP eingesetzt. Der Server öffnet einen IP-Port für diese Kommunikation. Die Portnummer wird in der `asinit.cfg` den Clientprogrammen bekannt gegeben. Die Client-Programme greifen nie direkt auf die Dokumente zu. Die Dokumente werden vom Server per TCP/IP auf Clientanfrage bereitgestellt.

enaio® kann somit mit einem Minimum an Freigaben im Netz auskommen. Ganz lassen sie sich aber nicht vermeiden, vor allem bei Multiserverinstallationen bzw. der Implementierung mehrerer Servergruppen. Da alle Server einer Gruppe Zugriff auf das gruppenweite Dateisystem mit Dokumenten benötigen, muss spätestens bei zwei Servern in einer Gruppe eine Freigabe des Datenverzeichnisses erfolgen. Da die Konfigurationen (außer der Archivierung) gruppenübergreifend sind, muss eine Freigabe des Verzeichnisses, in dem die sich Konfigurationsdateien befinden, für alle Server aller Gruppen erfolgen.

Aber aufgrund der Systemarchitektur existiert genau eine Installationsarchitektur, bei der ein Maximum an Systemsicherheit erreicht wird, bei einem System:

- § dass nur aus einer Servergruppe mit einem enaio®-Server besteht,
- § dass enaio®-Dateisystem lokal beim Anwendungsserver installiert,
- § der Fileserver für die Netzclients auf einem zweiten Rechner in einem freigegebenen Verzeichnis ist. Die Freigabe wird nur für den Zugriff der Clients benötigt. Auch diese Freigabe könnte noch vermieden werden, wenn lokale Installationen der Clients statt Netzclients verwendet werden
- § das in der Betriebsart DBPIPING (der empfohlenen Betriebsart) betrieben wird

Dann werden überhaupt keine Freigaben benötigt. Die gesamte Kommunikation läuft über TCP/IP, der Zugriff auf Dokumente erfolgt auf dem Server auch ohne laufende Loginsession.

Benutzer, Benutzergruppen und Systemrechte

Unter enaio® können nur dem System bekannte Benutzer Client-Programme starten. Jeder Benutzer erhält bei seiner Konfiguration ein Passwort. Beim Start eines jeden Clientprogramms wird ein Login-Dialog durchgeführt, bei dem das Passwort des Benutzers einzugeben ist.

Um einen Missbrauch des Systems zu erschweren, wurden drei, im enaio® enterprise-manager einstellbare, Sicherheitsebenen eingeführt:

- § SecurityLevel 0: nur Passwortüberprüfung (keine weiteren Einschränkungen)
- § SecurityLevel 1: Beenden der Anwendung nach 3 Fehlversuchen bei der Anmeldung
- § SecurityLevel 2: Sperren des Benutzerkontos nach 3 Fehlversuchen

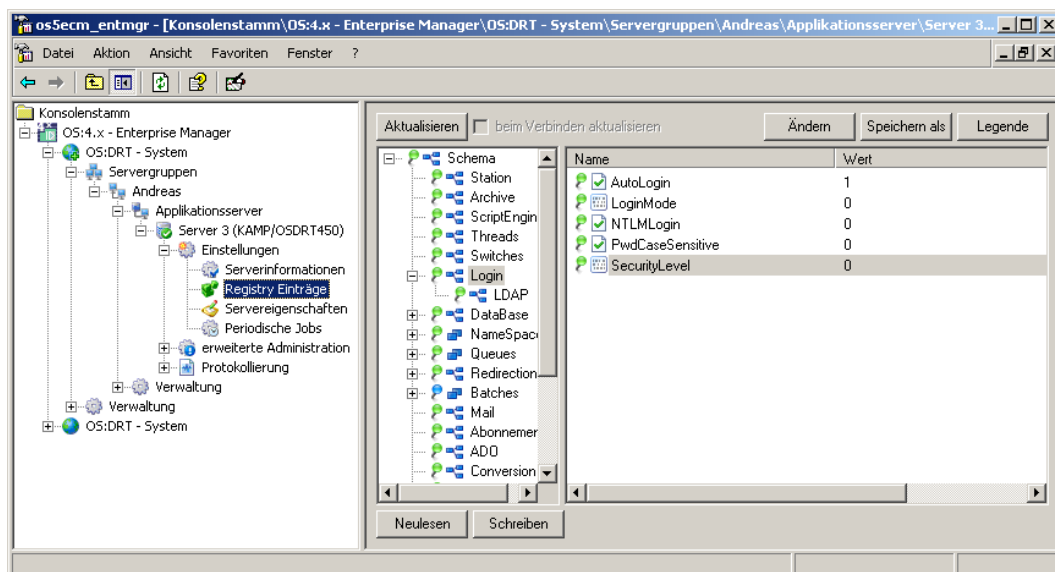


Abbildung 14: Parameter für die Festlegung des Login-Verhaltens

An dieser Stelle können auch weitere Einstellungen zum Login getätigt werden. Der Parameter 'AutoLogin' legt fest, ob automatisch der angemeldete Windows-Benutzer im System angemeldet wird. Über den Parameter 'LoginMode' kann zwischen Anmeldedialog und LDAP Anmeldung umgeschaltet werden. Auch eine Gross/Kleinschreibüberprüfung des Passwortes kann an dieser Stelle aktiviert oder deaktiviert werden.

Beim Anlegen eines Benutzerkontos im enaio® administrator im Menüpunkt Sicherheitssystem müssen dem Benutzerkonto Systemrollen zugewiesen werden. Diese Systemrollen bestimmen, welche Aktionen im enaio®-System unter diesem Benutzerkonto ausgeführt werden können.

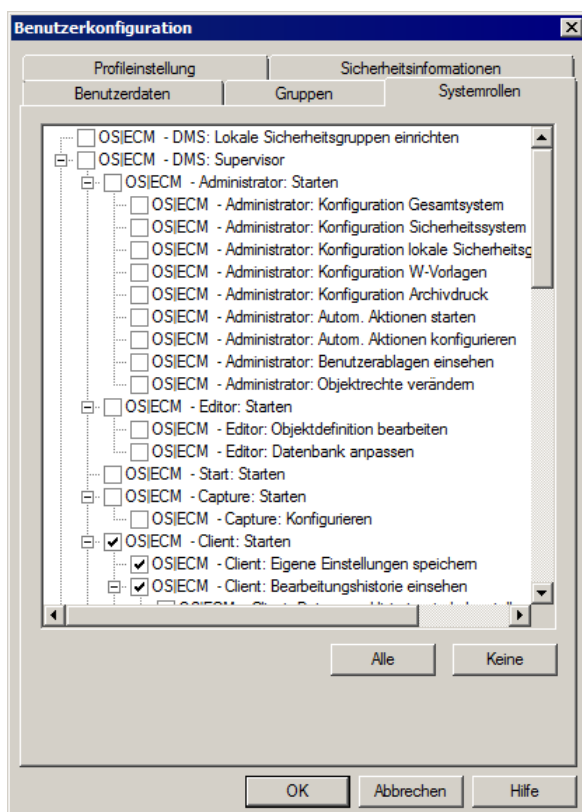


Abbildung 15: Systemrollen eines Benutzerkontos

Neben den Aktionsrechten gibt es noch Zugriffsrechte und -klauseln, die bestimmen, ob ein Benutzer Zugriffsrechte auf DMS-Objekte besitzt. Diese Rechte und Klausel lassen sich im enaio® administrator einstellen und gelten jeweils für einen Objekttyp (vgl. Abbildung 16, Abbildung 17).

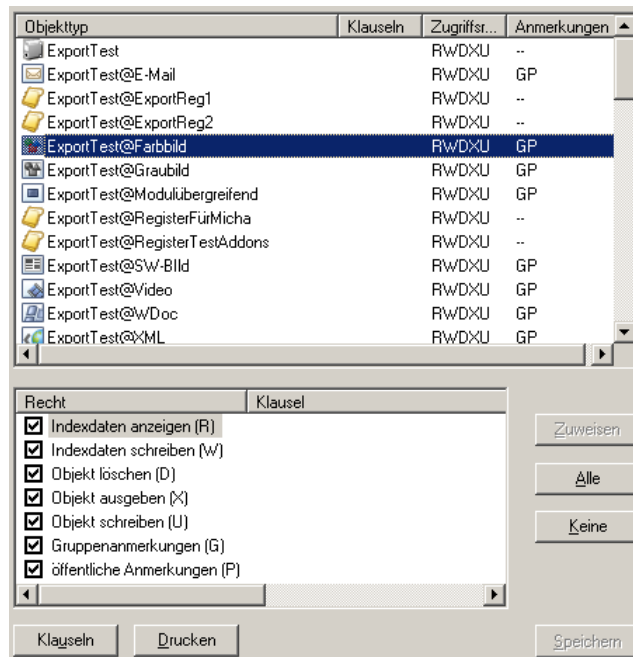


Abbildung 16: Gruppenabhängige Rechte im Sicherheitssystem

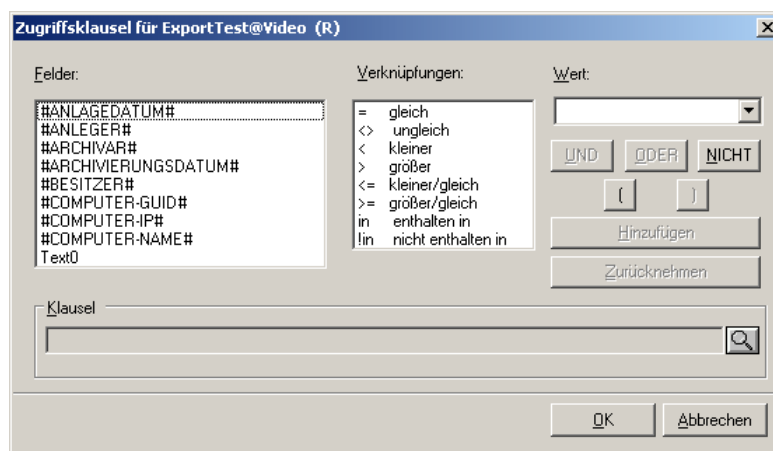


Abbildung 17: Dialog für das Anlegen von Zugriffsrechten (Klauseln) auf DMS-Objekte

Weitere Funktionen zur Datensicherheit

Zur weiteren Datensicherheit, zur Verhinderung nicht autorisierter Zugriffe ist die Verschlüsselung von Dokumenten möglich.

Die Verschlüsselung der Dokumentdateien erfolgt über die Microsoft Crypto-API - Bestandteil des Windows-Betriebssystems - mit dem Verschlüsselungsverfahren 'AES' und einer Schlüssellänge von 256 Bit.

Dokumentdateien können vom enaio® client und vom enaio® server verschlüsselt werden. Beide Verschlüsselungsarten können gemeinsam oder unabhängig voneinander eingesetzt werden und verwenden eigene symmetrische Schlüssel, die Bestandteil des Programmcodes sind.

Der enaio® client verschlüsselt mit dem Client-Schlüssel alle Dokumentdateien, die er zum enaio® server überträgt und entschlüsselt die Dokumentdateien, die er vom Server erhält. Die Dokumentdateien werden unverschlüsselt im Client-Cache gespeichert.

Der enaio® server verschlüsselt mit dem Server-Schlüssel alle Dokumentdateien, die vom enaio® client übertragen werden, bevor er sie speichert. Fordert ein Client ein Dokument an, entschlüsselt der enaio® server mit dem Server-Schlüssel die Dokumentdateien vor dem Übertragen.

Entsprechend der neuesten Entwicklungen im Bereich digitaler Dokumente und der Autorisierung über Chipkarten und Trustcenter wurde die Möglichkeit implementiert, Dokumente mit einer Elektronischen Signatur zu versehen, um so ihre Echtheit zu bezeugen und sie vor Fälschungen bzw. nicht autorisierter Änderung zu schützen.

Das Lizenzsystem unter enaio® kann unter anderem für die Erhöhung der Datensicherheit vom Benutzer systematisch eingesetzt werden. Indem nur bestimmte Stationen über Lizenzen für die Verschlüsselung oder Digitale Signatur verfügen, kann der nicht autorisierte Zugriff auf sicherheitsrelevante Daten oder sensible Dokumente vor allem in größeren Netzen in Einrichtungen mit Publikumsverkehr verhindert werden.

Installation von enaio®

Überblick

In diesem Kapitel werden Installationsszenarien von enaio® grob beschrieben. Eine genaue Beschreibung finden Sie in der detaillierten Installationsbeschreibung von enaio®.

Installationen dienen dazu, einen geplanten Systemzustand herzustellen. Technologische Basis des enaio®-Systems ist die gegebene PC- und Netzwerktechnologie, logisch ist das enaio®-System als Client-/Server-Architektur aufgebaut. Zur Abbildung der logischen Eigenschaften des Systems auf der technologischen Basis sind mehrere Rechner mit enaio®-Komponenten auszustatten. Zwar sind Installationen möglich, an denen alle Komponenten auf einem Rechner installiert sind. Diese sind aber Sonderfälle, die in Produktivsystemen höchstens bei Testsystemen vorkommen.

Daher besteht der Installationsvorgang für das gesamte DMS üblicherweise aus mehreren Einzel-Installationen an Server- und Arbeitsplatzrechnern. Serverkomponenten sind auf dedizierte Serverrechnern zu installieren und Client-Komponenten auf Arbeitsplatzrechnern. Dabei wird der Zustand des Gesamtsystems bestimmt durch den Stand der Installation auf den einzusetzenden Einzelrechnern.

Einzelinstallationen haben unterschiedliche Funktionen bei der Herstellung des DMS-Systems:

- § Herstellung eines Produktivzustands bei der Erstinstallation
- § Erweiterung des Systems
- § Reparatur von Server- oder Arbeitsplatz-Installationen
- § Update oder Upgrade von Servern und Arbeitsplätzen im Rahmen der Weiterentwicklung von enaio®

Die Installation von enaio® erfolgt mit einem Setup, das mit Installshield 7 erstellt wurde. Der Administrator, der die Installation durchführt, wird von einem Assistenten geführt. Entscheidungen in einem Dialog führen zu unterschiedlichen Folge-Dialogen. Damit werden umfangreiche Einstellungsdialoge vermieden, bei denen die Möglichkeit besteht, dass erforderliche Einstellungen vergessen werden. Stattdessen werden genau die Informationen abgefragt, die zu einer erfolgreichen Installation führen. Um den Installationsaufwand nach Möglichkeit zu minimieren, werden dem Administrator zusätzliche Hilfsmittel zur Verfügung gestellt, wie das *Silent Setup* oder der *Wartungsmodus*.

Mögliche Installationsvorgänge auf den einzelnen Rechnern werden einerseits bestimmt durch den Installationsstand des enaio®-Systems. So ist die Netzwerkinstallation von Client-Komponenten nur möglich, wenn im System mindestens ein Anwendungsserver läuft und ein Client-Fileserver zur Verfügung steht.

Andererseits werden die Installationsmöglichkeiten auch durch den Zustand des Rechners bestimmt, an dem das Setup gestartet wird. Die Installation von enaio®-Komponenten an einzelnen Rechnern ist abhängig davon, ob Komponenten schon auf diesem System installiert sind. Bei der Installation von enaio®-Komponenten werden in die Systemregistrierung Informationen über die Installation geschrieben. Damit wird der Wartungsmodus von Installshield verfügbar, der die Erweiterung, Aktualisierung und Reparatur der Installation auf dem Rechner stark vereinfacht. Der Wartungsmodus ist an einem Arbeitsplatzrechner nur verfügbar, wenn an diesem Rechner bereits eine enaio®-Installation durchgeführt wurde und der Wartungsmodus nicht explizit oder versehentlich abgeschaltet wurde. Der Wartungsmodus ist jedoch eingeschränkt, da die Erweiterung der Installation um einen Anwendungsserver damit nicht möglich ist. Wenn erforderlich, lässt sich der Wartungsmodus ausschalten.

Die Installationsszenarien folgen aus der Systemplanung. Danach müssen die verschiedenen enaio®-Systemkomponenten durch das Setup planungsgemäß auf ausgesuchten Rechner im Netzwerk installiert werden.

Ein enaio®-System kann folgende Software-Komponenten enthalten:

- § Anwendungsserver,
- § Verschiedene enaio®-Clients,
- § enaio® web-Komponenten,
- § Andere enaio®-Komponenten wie Office-Makros u. ä.,
- § Nicht-OS-Komponenten wie Datenbank, Hardware-Ansteuerungen etc.

Die Datenbank ist zwar eine zentrale Komponente für den Betrieb des enaio®-Systems, sie ist aber keine Komponente des enaio®-Systems. Einrichtung, Pflege und Sicherung der Datenbank gehört nicht zu den administrativen Tätigkeiten innerhalb des enaio®-Systems, wiewohl diese Tätigkeiten von großer Bedeutung für das System sind. Genauso gehören Hardwaretreiber für Scanner, Jukeboxen u. ä. nicht zum enaio®-System.

Diese Komponenten gilt es, auf die Zielrechner im Netzwerk so zu verteilen, dass die Mächtigkeit des enaio®-Systems genutzt werden kann. Die Flexibilität des enaio®-Systems erlaubt eine große Variationsbreite der möglichen Systemkonzepte. Sie reicht von einem Einzelplatzsystem, bei dem alle Komponenten auf einem Rechner installiert sind über ein einfaches System bis hin zu hoch komplexen Systemen. Diese drei Installationsmodelle sollen jetzt exemplarisch beschrieben und die erforderlichen Schritte zu ihrer Herstellung gezeigt werden.

Der Einsatz von Virensclannern hat Einfluss auf Anwendungen mit häufigen Zugriffen auf das Dateisystem. Insbesondere sind hiervon Arbeits- und Cachebereiche des enaio®-Servers, der Datenbanken und der E-Mail-Komponenten betroffen. In enaio® wird davon ausgegangen, dass Arbeits- und Cachebereiche ausschließlich und exklusiv von enaio®-Komponenten beschrieben und gelesen werden. Deshalb muss beim Einsatz von Virensclannern eine Abstimmung mit den umgebenden Anwendungskomponenten vorgenommen werden und ggf. einzelne Verzeichnisse von Scanläufen ausgenommen werden.

Ausbaustufen des enaio®-Systems

Die Flexibilität des enaio®-Systems und der Installationsroutinen erlauben eine große Variationsbreite bei der Dimensionierung des Systems. Von einem Einzelplatzsystem bis zu komplexen Systemen mit mehreren Netzsegmenten ist dabei alles möglich. Drei der möglichen Ausbaustufen sollen kurz vorgestellt

werden: Ein Einzelplatzsystem, das in der Praxis selten ist, eine typische Installation mit einem Anwendungsserver und ein komplexes System,

Einzelplatzsystem

Bei diesem System sind alle Komponenten auf einem Rechner installiert. Dies ist kein typisches enaio®-System, es kommt in der Praxis höchstens als Testsystem vor. Die folgende Abbildung zeigt ein solches System und die darauf installierten Komponenten.

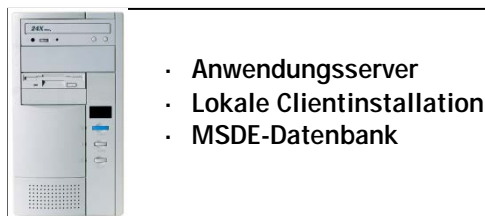


Abbildung 18: Einzelplatzsystem

Die Installationsanforderungen für das enaio®-System sind bei einer Installation auf einem Rechner genau die gleichen wie bei den komplexeren Systemen: Es sind Lizenzen für den Betrieb erforderlich, die verwendete Hardware und Software muss tauglich sein.

An diesem Rechner werden die frei verfügbare MSDE-Datenbank, der Anwendungsserver, sowie administrative und produktive Clients betrieben. Daher muss der Rechner den Anforderungen für die Installation der einzelnen Komponenten genügen, mit der Abweichung, dass nicht unbedingt ein Serverbetriebssystem für diese Einsatzform erforderlich ist.

Bei einem Einplatz-System kann die komplette Installation in einem Schritt durchgeführt werden. Dazu wird als Installations-Aktion *Benutzerdefiniert* ausgewählt. Danach werden alle erforderlichen Komponenten installiert.

Nach der Installation werden dann die erforderlichen administrativen Tätigkeiten zur Inbetriebnahme des Systems durchgeführt.

Einfaches System mit einem Anwendungsserver

Dieses ist ein typisches enaio®-System. Die Komponenten sind auf mehrere Rechner verteilt, die über ein TCP/IP-Netzwerk verbunden sind. Die folgende Abbildung zeigt ein solches System und die installierten Komponenten.

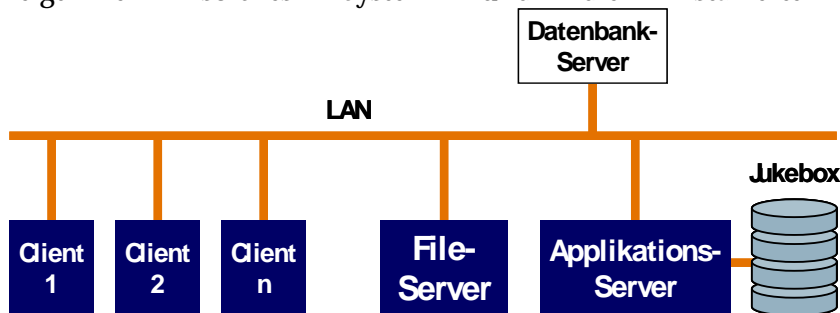


Abbildung 19: Einfaches System mit einem Anwendungsserver

Die Installationsanforderungen für das System sind differenziert, da Clientkomponenten andere Anforderungen haben als Serverkomponenten. Im Netz bietet ein geeigneter Datenbankserver seine Dienste unabhängig an.

Bei einem System einfacher Ausbaustufe mit einem Anwendungsserver und einem Fileserver sind mehrere Schritte durchzuführen:

- § Die Datenbank muss als erstes verfügbar gemacht werden.
- § Dann wird ein Anwendungsserver installiert.
- § Danach wird ein Fileserver installiert.
- § Schließlich werden alle erforderlichen Clients per Netzwerkinstallation installiert.

Nach der Installation werden dann wiederum die erforderlichen administrativen Tätigkeiten zur Inbetriebnahme des Systems durchgeführt.

Komplexes System mit mehreren Servergruppen

Dieses ist ein enaio®-System mit hoher Ausbaustufe. Die Komponenten sind auf mehrere Rechner verteilt, die über ein TCP/IP-Netzwerk verbunden sind. Das Gesamtsystem ist auf mehrere Netzwerksegmente verteilt, die weit auseinander liegen. Einige Clients sind möglicherweise per DFÜ angebunden. Das System enthält folgende Komponenten:

- § Einer Anzahl von Anwendungsservern mit mehreren Jukeboxen, Balanced Server und Servercluster
- § mehrere Servergruppen
- § mehrere Fileserver
- § eine große Anzahl von installierten Clients mit Einzelplatzscannern
- § eine Anzahl von enaio® capture-Clients mit mehreren Hochleistungsscannern
- § installiertes enaio® web

Die folgende Abbildung zeigt ein solches System und die installierten Komponenten.

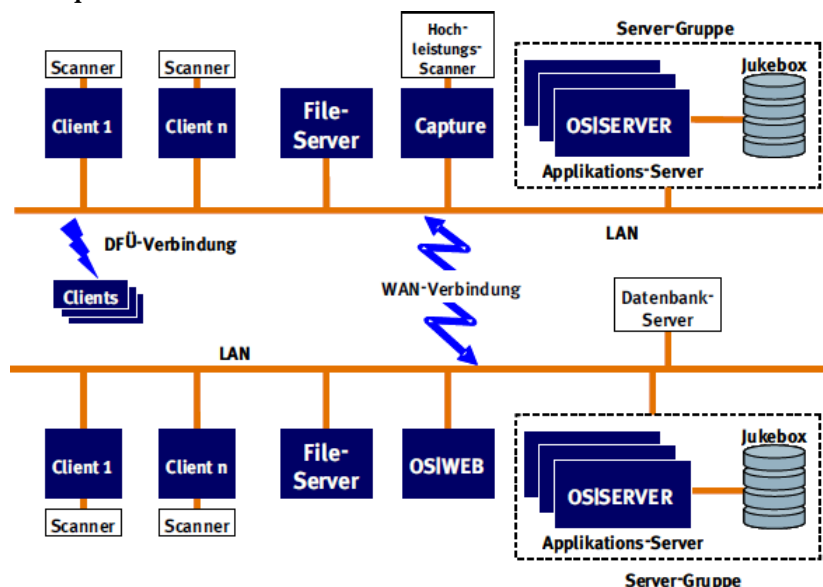


Abbildung 20: Komplexes enaio-System

Je nach Komplexität des geplanten Systems sind unterschiedliche Installationsschritte zur Herstellung des Systems erforderlich. Die meisten Schritte sind naturgemäß bei komplexen Systemen mit mehreren Anwendungsservern in mehreren Servergruppen erforderlich. Der Ablauf könnte wie folgt aussehen:

- § Zunächst wird ein Anwendungsserver installiert, implizit wird die erste Servergruppe installiert. Die erforderliche Treiber-Software für die Jukeboxen ist zu installieren

- § Zur ersten Servergruppe wird die geplante Anzahl von Anwendungsservern hinzugefügt
- § Es werden weitere Servergruppen hinzugefügt
- § Zu jeder der Servergruppen wird die gewünschte Anzahl von Anwendungsservern hinzugefügt
- § Dann werden die Fileserver installiert
- § Es werden alle erforderlichen Clients installiert. Zum Betrieb der Scanner ist die erforderliche Treiber-Software zu installieren
- § Es werden alle erforderlichen enaio® capture-Clients installiert
- § Es wird enaio® web installiert

Bei allen Installations-Aktionen ist das Vorhandensein einer geeigneten und freigegebenen Datenbank erforderlich.

Die Reihenfolge der einzelnen Installationen ist bei komplexen Systemen variabel. Das strikte Einhalten der beschriebenen Reihenfolge ist nicht unbedingt der effizienteste und schnellste Weg zur Einrichtung des Systems. Fest steht jedoch, dass die Installation eines Anwendungsservers in allen Fällen der erste Installationsschritt sein muss. Außerdem gibt es weitere Abhängigkeiten. So kann z.B. die Netzwerkinstallation erst durchgeführt werden, wenn eine Fileserver-Installation existiert und das Fileserver-Verzeichnis freigegeben ist.

Zur Verkürzung der Installationszeit kann parallel zur Installation der für das gewünschte Zielsystem erforderlichen Komponenten mit der Einrichtung des Systems begonnen werden. Sobald mindestens ein Anwendungsserver und ein administrativer Client installiert sind, kann in enaio® editor die Objektdefinition entworfen werden. Mit dem enaio® administrator können dann weitere erforderliche Konfigurationen vorgenommen werden.

Anlegen von Dokumenten

Allgemeine Voraussetzungen

Für das Anlegen von Ordnern, Registern und Dokumenten benötigt der Client die Objektdefinition. Im enaio® editor werden folgende Objektdefinitionen erzeugt und definiert:

- § die Objekttypen im DMS
- § die Dialogelemente der Verschlagwortungsmasken zu den Objekttypen
- § für Dokumenttypen zusätzlich die Module, über die zugeordnete Dokumente erzeugt werden
- § und die Objektrelationen, über die vorgegeben wird, auf welcher hierarchischen Ebene eines Ordnerstyps welche Register- und Dokumenttypen angelegt werden können

Der enaio® editor erzeugt ebenfalls die an die Objektdefinition angepassten Datenbanktabellen, in denen folgende Daten gespeichert werden:

- § Verschlagwortung
- § weitere automatisch mitgeführte Daten wie 'Anleger' und 'Erstellungsdatum' – die Basisparameter
- § Daten für die Bearbeitungshistorie
- § Objektrelationen

Beim Start liest der Client die Objektdefinition und die gruppenspezifischen Zugriffsrechte auf DMS-Objekte aus der Datenbank, ermittelt die Rechte des angemeldeten Benutzers und stellt dem Benutzer die Funktionen zum Zugriff auf DMS-Objekte und zum Erstellen von DMS-Objekten zur Verfügung.

Folgende Daten werden im enaio® administrator verwaltet und in der Datenbank gespeichert:

- § Benutzer
- § Zuordnung von Benutzern zu Gruppen
- § gruppenspezifischen Zugriffsrechte auf Objekttypen

Legt ein Benutzer im enaio® client ein neues DMS-Objekt an, öffnet der Client die Verschlagwortungsmaske des gewählten Objekttyps. Der Benutzer speichert die Verschlagwortung und der Client trägt die Daten, Verschlagwortung, Standort und eine eindeutige ID, die er vom Server erhält, in die Datenbank ein.

Dokumente, denen Benutzer Dateien zuordnen, werden durch einen Eintrag in der Datenbank gesperrt. Andere Benutzer können zu diesem Zeitpunkt keine Dateien hinzufügen, die Verschlagwortung kann allerdings von anderen Benutzern eingesehen und auch editiert werden.

Danach ordnet der Benutzer Dokumenten Dateien zu. Dieser Vorgang unterscheidet sich für Bilddokumente, Windows-Dokumente, E-Mails, Film-Dokumente und XML-Dokumente.

Der Client verwaltet die Dokumentendateien in einem benutzerspezifischen Arbeits- und CACHE-Bereich. Legt der Benutzer Dokumentendateien ab, werden sie zum Server übertragen und vom Server im WORK-Bereich gespeichert.

Die Objektdefinition in der Datenbank

Die Objektdefinition wird im enaio® editor erzeugt und in der Datenbank gespeichert:

- § die Tabelle `osobjdef` enthält die Definition der DMS-Objekttypen
- § die Tabelle `osobjfields` enthält die Definition der einzelnen Verschlagwortungsfelder der DMS-Objekttypen
- § die Tabelle `osconf` enthält u.a. Konfigurationseinträge von Verschlagwortungsfeldern, beispielsweise die Daten von Katalogen
- § die Tabelle `osproperties` enthält ebenfalls Eigenschaften von DMS-Objekttypen

Die Tabelle `osobjdef` enthält in der Spalte `tablename` die Bezeichnung der Tabelle, in der die Verschlagwortung eines Objekts dieses Typs gespeichert wird.

Objektrelationen in der Datenbank

Objektrelationen geben vor, auf welcher hierarchischen Ebene eines Ordnerstyps Register- und Dokumenttypen angelegt werden können. Die Objektrelationen werden im enaio® editor erstellt und in der Tabelle `osobjparrel` gespeichert.

Diese Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|-------------------------|------------------------------------------------------------------------------------------------------------------|
| <code>objtype</code> | der Objekttyp eines Ordner- oder Registertyps |
| <code>parobjtype</code> | der Objekttyp der Objekte (Ordner/Register), in denen Einschränkungen auf die Instanzierung von ‚objtype‘ gelten |
| <code>maxcount</code> | Maximale Anzahl der anzulegenden Objekte des Typs |

Tabelle 3

Bei der Recherche und bei der Anzeige von Inhaltlisten werden Objektrelationen nicht beachtet.

Konfigurationseinträge für Kataloge, AddOns und das Dialogelement 'Tabelle'

Im enaio® editor können Dialogelementen Kataloge und AddOns zugeordnet werden. Kataloge sind einfache oder strukturierte Listen, aus denen Benutzer Einträge für die Verschlagwortung auswählen. AddOns sind Bibliotheken, über die zusätzliche Funktionen, beispielsweise Funktionen für eine strukturierte Verschlagwortung, zur Verfügung gestellt werden.

Die Tabelle `osobjfields` mit der Definition der einzelnen Verschlagwortungsfelder enthält die Spalte `flags`, in die eingetragen ist, ob und welcher Katalog oder welches AddOn zugeordnet ist.

Für viele Kataloge und AddOns sind Konfigurationsangaben notwendig. In diesem Fall wird in die Spalte `osconfid` der Tabelle `osobjfields` eine ID für die Konfigurationsangaben eingetragen. In der Tabelle `osconf` sind in der Spalte `osconfval` für jede `osconfid` die Konfigurationsangaben, so wie sie im enaio® editor als Text angezeigt werden, in einem BLOB-Feld gespeichert. Der `osconftyp` von Katalog- und AddOn-Einträgen in der Tabelle `osconf` ist 6.

AddOns werden ohne Pfad gespeichert, sie werden im Verzeichnis \clients\client32 gesucht. Strukturbaum-Dateien, externe Programme und Skripte werden ebenfalls dort gesucht. Falls sie dort nicht gefunden werden, wird über den in der Tabelle `osconf` in der Spalte `osconfval` gespeicherten Pfad mit zugeordneter Laufwerksangabe gesucht.

Für das Dialogelement 'Tabelle' werden ebenfalls Konfigurationsangaben - Spaltenbezeichnungen und Eigenschaften von Spalten - verwaltet. Wie bei Katalogen und AddOns wird in die Tabelle `osobjfields` eine ID für die Konfigurationsangaben eingetragen. Die Daten selbst werden über die Tabelle `oslistctrl` verwaltet. Dort sind für jede Spalte die Bezeichnung im Client, Datentyp, Feldlänge und die Bezeichnung der Spalte, in die in der Tabelle `object#list#` die Verschlagwortung eingetragen wird, verzeichnet. (# ist die fortlaufende Nummer)

Verschlagwortung der Ordner-, Register-, Dokumenttypen

Der enaio® editor erzeugt beim Anpassen der Tabellen für jeden DMS-Objekttyp eine Verschlagwortungstabelle. Die Tabelle enthält für jedes angelegte Verschlagwortungsfeld eine Spalte und weitere Spalten für Systemdaten und für Basisparameter. Der Standort der DMS-Objekte innerhalb der hierarchischen Struktur aus Ordnern und Register# wird in Standorttabellen gespeichert.

Die Tabellen von Ordnerarten tragen die Bezeichnung `stamm#`, die Tabellen von Registertypen die Bezeichnung `register#` und die Tabellen von Dokumenttypen die Bezeichnung `object#`.

Die Tabellenbezeichnungen werden im enaio® editor im Eigenschaftsdialog der DMS-Objekte angezeigt, die Feldbezeichnungen in der Datenbankansicht von enaio® editor.

Die Tabellen `stamm#` (Ordner) haben folgende Struktur:

| Spalte | Inhalt |
|--------------------------|---------------------------------------------------------------|
| <code>id</code> | die eindeutige ID eines DMS-Objekts |
| <code>zeitstempel</code> | der genaue Zeitpunkt, zu dem das DMS-Objekt angelegt wurde |
| <code>links</code> | Kennzeichnung, ob Notizen existieren |
| <code>anleger</code> | der Benutzer, der das Objekt angelegt hat |
| <code>angelegt</code> | der Zeitpunkt des Anlegens |
| <code>modifyuser</code> | der Benutzer, der das Objekt zuletzt geändert hat |
| <code>modifytime</code> | der Zeitpunkt der letzten Änderung |
| <code>deleted</code> | Löschflag für das Objekt |
| <code>osowner</code> | Besitzer des Objekts |
| <code>feld#</code> | die Verschlagwortung des DMS-Objekts in den einzelnen Feldern |

Tabelle 4

Die Spaltenbezeichnung für die Dialogelemente wird in der Datenbankansicht von enaio® editor angezeigt und steht ebenfalls in der Spalte `fieldname` der Tabelle `osobjfields`. Die Feldbezeichnungen, die der enaio® editor vergibt, korrespondieren mit den Datentypen. Der Editor verwendet `feld`, `real`, `datum`, `zahl` und `list` für das Dialogelement 'Tabelle'.

Die Tabellen `register#` (Register) haben folgende Struktur:

| Spalte | Inhalt |
|-----------------|-----------------------------------|
| <code>id</code> | die eindeutige ID eines Registers |

| | |
|-------------|--------------------------------------------------------------------------------------|
| stamm_id | die ID des Ordnerstyps, dem das Register zugeordnet ist |
| parent_id | die eindeutige ID des Registers, falls ein Register in einem anderen Register liegt. |
| links | Kennzeichnung, ob Notizen existieren |
| anleger | der Benutzer, der das Register angelegt hat |
| angelegt | der Zeitpunkt des Anlegens |
| modifyuser | der Benutzer, der das Register zuletzt geändert hat |
| modifytime | der Zeitpunkt der letzten Änderung |
| zeitstempel | der genaue Zeitpunkt, zu dem das Register angelegt wurde |
| deleted | Löschflag für das Register |
| osowner | Besitzer des Registers |
| feld# | die Verschlagwortung des Registers in den einzelnen Feldern |

Tabelle 5

Die Tabellen object# (Dokument) haben folgende Struktur:

| Spalte | Inhalt |
|-------------|---------------------------------------------------------------------------------------------------------|
| id | die eindeutige ID eines Dokuments |
| zeitstempel | der genaue Zeitpunkt, zu dem das Dokument angelegt wurde |
| haupttyp | der Haupttyp kennzeichnet das Modul des Dokumenttyps |
| untertyp | der Untertyp kennzeichnet den Dokumenttyp, aus Haupt- und Untertyp ergibt sich der eindeutige Objekttyp |
| anzahl | die Anzahl der Dateien des Dokuments |
| angelegt | der Zeitpunkt des Anlegens |
| anleger | der Benutzer, der das Dokument angelegt hat |
| archiviert | das Archivierungsdatum für die Objektinformation |
| archivar | der Archivar für die Objektinformation |
| flags | Kennzeichnung des Archivierungsstatus |
| links | Kennzeichnung, ob Notizen existieren |
| medium_doc | die ID des Bereiches, in dem das Dokument gespeichert ist |
| name_doc | der Pfad und die Bezeichnung eines archivierten Dokument |
| medium_dia | die ID des Bereiches, in dem das Dia zum Dokument gespeichert ist |
| name_dia | der Pfad und die Bezeichnung eines archivierten Dias |
| version | Kennzeichnung von Varianten bei Windows-Dokumenten |
| lockuser | der Benutzer, der das Dokument ausgecheckt hat |
| systemid | Kennzeichnung eines anderen Archivsystems |
| foreignid | die eindeutige ID in einem anderen Archivsystem |
| modifyuser | der Benutzer, der das Objekt zuletzt geändert hat |
| modifytime | der Zeitpunkt der letzten Änderung |
| deleted | Löschflag des Dokuments |
| osowner | Besitzer des Dokuments |
| feld# | die Verschlagwortung des Objekts in den einzelnen Feldern |

Tabelle 6

Für jeden Ordner-, Register- und Dokumenttyp wird eine weitere Tabelle angelegt, eine Schattentabelle. Die Tabellen tragen die gleiche Bezeichnung mit einem angehängten 's'. Erhält ein DMS-Objekttyp im enaio® editor die Eigenschaft

'Indexdatenhistorie erzeugen', werden die Historiendaten in die Schattentabelle eingetragen.

Die Schattentabellen haben folgende Struktur:

| Spalte | Inhalt |
|--------|----------------------------------------------------------------------------|
| osguid | die eindeutige ID der Änderung |
| feld# | die Verschlagwortung des Objekts in den einzelnen Feldern vor der Änderung |

Tabelle 7

Die Daten der Änderung, beispielsweise Zeitpunkt, Benutzer, Arbeitsstation, Typ der Änderung, werden in die Tabelle `osobjhist` eingetragen und sind dort ebenfalls durch diese eindeutige ID gekennzeichnet.

Besitzt ein Dokumenttyp im enaio® editor die Eigenschaft 'Dokumentenhistorie erzeugen', speichert enaio® server die Dokumente vor der Änderung in einem Unterverzeichnis im WORK-Bereich. Das Unterverzeichnis trägt den Namen der Objekt-ID (hexadezimal).

Verschlagwortung des Dialogelements 'Tabelle'

Die Verschlagwortung des Dialogelements 'Tabelle' wird ebenfalls, analog zu Mehrfachfeldern, in eigenen Tabellen gespeichert, die der enaio® editor anlegt. Die Bezeichnung der Tabellen hat folgende Struktur: `object#list#`. Beispiel: Das Verschlagwortungsfeld `list3` des Dokumenttyps `object27` ist eine Tabelle. Die Datenbanktabelle trägt die Bezeichnung `object27list3`.

Die Datenbanktabellen haben folgende Struktur:

| Spalte | Inhalt |
|--------|------------------------------------------|
| id | die eindeutige ID des DMS-Objekts |
| line | die Zeilennummer |
| feld# | die Verschlagwortung in der #-ten Spalte |

Tabelle 8

Zu den Datenbanktabellen für Tabellen werden ebenfalls Schattentabellen für die Indexdatenhistorie angelegt. Sie erhalten in der Bezeichnung ein angehängtes 's'. Die angezeigten Spaltenbezeichnungen im Client und die Eigenschaften der Spalten werden in der Tabelle `oslistctrl` gespeichert.

Der Standort von Registern und Dokumenten

Dokumente werden in enaio® in einer hierarchischen Struktur aus Ordnern und Registern gespeichert. Die oberste Ebene bilden die Ordner. In Ordnern können Register und Dokumente angelegt werden, in Registern können weitere Register und Dokumente angelegt werden. Register befinden sich immer in einem Ordner oder in einem anderen Register, Dokumente befinden sich ebenfalls immer in einem Ordner oder einem Register. Dieser Standort wird für Register in der Tabelle `osparregrel` und für Dokumente in der Tabelle `sdrel` gespeichert.

Standort der Register

Der Standort eines Registers wird in der Tabelle `osparregrel` gespeichert.

Die Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|------------|----------------------------------------------------------|
| folderid | die eindeutige ID des Ordners, in dem das Register liegt |
| foldertype | die ID des Ordnertyps, in dem das Register liegt |

| | |
|--------------|----------------------------------------------------------|
| registerid | die eindeutige ID des Registers |
| registertyp | die ID des Registertyps |
| parentregid | die eindeutige ID des Registers der nächst höheren Ebene |
| parentregtyp | die ID des Registertyps der nächst höheren Ebene |

Tabelle 9

Standort der Dokumente

Der Standort eines Dokuments innerhalb des Archivs wird in der Tabelle `sdrel` gespeichert.

Die Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|-------------|------------------------------------------------------------|
| loeschen | zur Zeit nicht verwendet |
| zeitstempel | der genaue Zeitpunkt, zu dem das DMS-Objekt angelegt wurde |
| stamm_id | die eindeutige ID des Ordners, in dem das Dokument liegt |
| object_id | die eindeutige ID des Dokuments |
| objekttyp | die ID des Dokumenttyps |
| register | die eindeutige ID des Registers der nächst höheren Ebene |
| regtyp | die ID des Registertyps der nächst höheren Ebene |

Tabelle 10

Der Ablageort der Dokumentendateien ist nicht in der Datenbank gespeichert. Er ergibt sich aus Untertyp und Haupttyp und der ID des Dokuments.

Die Protokollierung in der Datenbank

Jede Aktion wird vom Client über die Tabelle `osobjhist` in der Datenbank protokolliert.

Die Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|-----------|--------------------------------------------------------------------------------------------------|
| osguid | die eindeutige ID der Aktion |
| osid | die eindeutige ID des DMS-Objekts, auf das die Aktion bezogen ist |
| ostype | die ID des Objekttyps |
| osuser | die eindeutige ID des Benutzers, der die Aktion ausgeführt hat |
| ostime | der Zeitstempel |
| osaction | eine Aktions-ID |
| osstation | die GUID der Arbeitsstation, an der die Aktion ausgeführt wurde |
| osinfo | der Info-Text, der in der Bearbeitungshistorie zusätzlich zur Aktionsbeschreibung angezeigt wird |

Tabelle 11

Diese Protokollierung der Aktionen ist unabhängig von der OXRPT-Protokollierung, die für alle enaio®-Komponenten konfiguriert werden kann. Den Aktions-IDs ist in der Tabelle `oshistact` eine Bezeichnung und eine Detail-Beschreibung zugeordnet, die in der Bearbeitungshistorie angezeigt werden. Im

Folgenden ist eine unvollständige Übersicht über Aktionstypen, die protokolliert und in der Bearbeitungshistorie angezeigt werden, gegeben:

| osaction | osinfo (Bezeichnung) | osexinfo (Detailbeschreibung) |
|----------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Digitale Signatur | Das Dokument wurde digital signiert. Über die Ansichtsoptionen besteht die Möglichkeit, die digitale Signatur zu verifizieren. |
| 2 | Objekt angelegt | Das Objekt wurde durch Funktionalitäten des Clients oder durch einen Import erzeugt. |
| 3 | Indexdaten geändert | Die Indexdaten des Objekts oder dessen Status wurden durch Funktionalitäten des Clients oder durch ein Update mittels des Imports geändert. |
| 4 | Dokument geändert | Das Dokument wurde durch Funktionalitäten des Clients oder durch ein Update mittels des Imports geändert. |
| 5 | Dokument archiviert | Das Dokument wurde rechtssicher archiviert. Eine Änderung ist nicht mehr möglich. |
| 6 | Dokument gelöscht | - |
| 7 | Dokument ausgegeben | Das Dokument wurde vom Anwender gelesen, gedruckt oder anderweitig ausgegeben. Eine Änderung erfolgte nicht. |
| 8 | Dokumentstatus geändert | Das Dokument wurde mit dem Status archivierbar versehen. |
| 9 | Dokumentstatus geändert | Das Dokument wurde mit dem Status nicht archivierbar versehen. |
| 10 | Dokument erzeugt | Das Dokument wurde durch Funktionalitäten des Clients oder durch einen Import erzeugt. |
| 11 | Verknüpfung hergestellt | - |
| 12 | Verknüpfung aufgelöst | - |

Tabelle 12

Die Aktionen 3 (Indexdaten geändert), 4 (Dokument geändert), 11 (Verknüpfung hergestellt) und 12 (Verknüpfung aufgelöst) werden als Modifizierungsdaten geführt.

Dokumentensperrung in der Datenbank

Legt ein Benutzer ein Dokument an, wird es zuerst verschlagwortet. Der Client schreibt die Daten in die Tabelle `object#` und trägt den Standort in die Tabelle `sdrel` ein. Der Client trägt das Dokument ebenfalls in die Tabelle `doclock` ein. Dadurch kann kein anderer Benutzer Dokumentendateien zuordnen. Die Sperrung des Dokuments wird bei jedem Bearbeiten dieses Dokuments durch einen Benutzer ausgelöst. Im enaio® client wird dieses Dokument dann durch ein Schloss gekennzeichnet. Hat der Benutzer das Bearbeiten des Dokuments abgeschlossen, wird das Dokument wieder aus der Tabelle `doclock` ausgetragen und ist damit zum Bearbeiten freigegeben.

Die Tabelle `doclock` hat folgende Struktur:

| Spalte | Inhalt |
|-------------|--------------------------------------------------------------|
| object_id | die eindeutige ID des DMS-Objekts |
| object_type | die ID des Objekttyps |
| user_id | die eindeutige ID des Benutzers, der das Dokument bearbeitet |
| user_time | der Zeitstempel |
| info | Statusinformation |
| fileinfo | Änderungsdatum für Windows-Dokumente |
| station | Arbeitsstation des Benutzers |

Tabelle 13

Verschlagwortung und Standort von DMS-Objekten

Ordner, Register und Dokumente werden auf folgenden Wegen angelegt:

- § Import von Daten
- § enaio® capture
- § enaio® client

Der enaio® client liest beim Start die Objektdefinition, die Objektrelationen und die Katalog-, AddOn- und Tabellen-Konfigurationseinträge aus der Datenbank.

Werden Objektdefinitionen, Objektrelationen und Konfigurations-Einträge nach dem Start von enaio® client geändert, muss der enaio® client neu gestartet werden, um die aktuellen Daten zu erhalten. Es müssen ebenfalls alle enaio®-Server neu gestartet werden oder bei allen enaio®-Servern die betroffenen Exekutoren ('std' und 'dms') neu geladen werden.

Der enaio® client ermittelt beim Starten die Rechte des angemeldeten Benutzers aus den Zugriffsrechten und Gruppenzugehörigkeiten in der Datenbank.

Werden diese Daten nach dem Start von enaio® client geändert, kann der enaio® client diese Daten über 'Umschalttaste-Strg-F5' aktualisieren.

Der enaio® client liest ebenfalls benutzerspezifische Einstellungen aus der Tabelle `osconf`. In der Tabelle werden für jeden Benutzer (Spalte `osconfuser`) als Typ 3 (Spalte `osconftype`) in der Spalte `osconfval` benutzerspezifische Einstellungen verwaltet. Diese und weitere Einstellungen werden beim Start ausgelesen und beim Beenden des Clients gespeichert.

Nicht benutzerspezifische Einstellungen werden über die Datei `as.cfg` verwaltet. Der Pfad zu dieser Datei ist in der Tabelle `osresources` eingetragen. Beim Start des Clients legt der Server eine Kopie der Datei `as.cfg` in das Verzeichnis

`\Temp\ostemp\'ServerName\'#\'Port\'\etc\.`

Der Client liest die Einstellungen aus dieser Datei.

Werden diese Daten nach dem Start von enaio® client geändert, muss der enaio® client neu gestartet werden, um die aktuellen Daten zu erhalten. Das Beenden eines beliebigen oder aller Clients kann im enaio® enterprise-manager vom Systemadministrator erzwungen werden.

Ordner, Register und Dokumente werden verschlagwortet und erhalten einen Standort im DMS. Die Verschlagwortung trägt der Benutzer manuell oder unterstützt durch Funktionen in die typspezifische Verschlagwortungsmaske ein, die der enaio® client aus den Daten, die er beim Start liest, erstellt. Speichert der Benutzer die Verschlagwortung, überprüft der Client die Daten. Daten, die den

Vorgaben nicht entsprechen werden gekennzeichnet und der Benutzer aufgefordert, die Daten zu korrigieren. Entspricht die Verschlagwortung den Vorgaben, werden die Daten in der Datenbank gespeichert.

AddOns können die Eigenschaft 'automatisch abfahren' erhalten. Dann werden sie beim Speichern der Verschlagwortung automatisch ausgeführt.

Der Standort eines Ordners ist immer die oberste Ebene in der hierarchischen Struktur. Legt ein Benutzer Register und Dokumente an, öffnet er zuerst einen Ordner und legt so den Standort fest. Dokumente können ebenfalls in der benutzerspezifischen Ablage angelegt werden. Bei Dokumenten, die über das Dateisystem übernommen werden, kann der Benutzer die Ablage oder einen zuvor geöffneten Ordner als Standort wählen.

Zwischenspeicherung von Verschlagwortungen

Benutzer können für jeden Objekttyp die zuletzt eingetragene Verschlagwortung zwischenspeichern und in einer neuen Verschlagwortungsmaske Voreintragen lassen. Diese Daten werden mit den anderen benutzerspezifischen Einstellungen in der Tabelle `osconf` gespeichert, beim Start ausgelesen und beim Beenden des Clients gespeichert.

Die Verschlagwortung des Dialogelements 'Tabelle' wird nicht zwischengespeichert. Beim Voreintragen wird nicht überprüft, ob die Daten den Vorgaben entsprechen. Die Daten werden über die Tabulatoren-Reihenfolge der Felder gespeichert. Wird die Reihenfolge im enaio® editor geändert, werden zwischengespeicherte Verschlagwortungen falsch eingetragen.

Anlegen von Ordnern und Registern

Legt ein Benutzer einen neuen Ordner an, öffnet der Client die Verschlagwortungsmaske. Speichert der Benutzer die Daten, fordert der Client vom Server eine ID für den Ordner an. Der Server ermittelt die nächste freie ID über die Tabelle `osnextindex` und trägt dort die vergebene ID, seine Servergruppen-ID und den Zeitstempel in die Tabelle ein und sendet die ID an den Client. Der Client trägt die Daten in die entsprechende Tabelle `stamm#` ein und protokolliert die Aktion in der Tabelle `osobjhist`.

Legt ein Benutzer ein neues Register an, trägt der Client zusätzlich den Standort des Registers in die Tabelle `osparregrel` ein.

Beispiel für die Tabellenstruktur nach dem Anlegen eines Registers:

Der Server trägt in die Tabelle `osnextindex` die jeweils zuletzt vergebene ID ein:

| id | serverid | time |
|------|----------|------------|
| 1311 | 2 | 1033551104 |

Der Client trägt in die Tabelle `register#` die Daten ein:

| id | stamm_id | parent_id | feld1 |
|------|----------|-----------|------------------|
| 1311 | 1203 | 1286 | Verschlagwortung |

Der Client trägt in die Tabelle `osparregrel` den Standort ein:

| folderid | foldertype | registerid | registertyp | parentregid | parentregtyp |
|----------|------------|------------|-------------|-------------|--------------|
| 1203 | 22 | 1311 | 6488079 | 1286 | 6488079 |

Die Aktion protokolliert der Client in der Tabelle `osobjhist`.

Anlegen von Bilddokumenten

Bilddokumente sind Dokumente, denen Grauwertbilder, Schwarz/Weißbilder oder Farbbilder zugeordnet werden können. Welcher Dateityp einem Dokumenttyp zugeordnet werden kann, wird beim Einrichten des Dokumenttyps im enaio® editor angegeben. Dieser Typ ist der Haupttyp eines Dokuments und legt das Bearbeitungsmodul der Dokumente fest und damit auch das Dateiformat. Innerhalb des DMS werden Bilddateien nicht über Dateiendungen identifiziert sondern über ihren Haupttyp.

| Haupttyp | Typ des Bilddokuments | Modulkürzel | Dateiformat |
|----------|-----------------------|-------------|--------------------------|
| 1 | Grauwertbilder | X | JPEG (maximale Qualität) |
| 2 | Schwarz/Weißbilder | D | TIFF G4 |
| 3 | Farbbilder | P | JPEG (maximale Qualität) |
| 4 | Windows | W | Anwendungsformat |
| 5 | Filmdokumente | M | MPEG, AVI |
| 6 | E-Mail | E | MIME, MSG |
| 7 | XML | | XML, XSL, XSLT |

Tabelle 14

Die Bearbeitungsmodule für Grauwertbilder, Schwarz/Weißbilder und Farbbilder im enaio® client übernehmen Daten über die TWAIN- oder KOFAX-Schnittstelle aus Scannern oder direkt Bilder aus dem Dateisystem und speichern sie im DMS-Dateiformat im Arbeitsbereich. Dort erhalten Sie die Endung `AS'Modulkürzel'`.

Über das Dateisystem können Bilddateien mit folgenden Formaten übernommen werden:

- § TIFF-Dateien (*.TIF) - unkomprimiert oder Fax G4,
- § JPEG-Dateien (*.JPG),
- § BMP-Dateien (*.BMP),
- § PCX-Dateien (*.PCX),
- § Dicom-Dateien (*.DCM),
- § Targa-Dateien (*.TGA),
- § Gif-Dateien (*.GIF) - nur unkomprimiert.

Die Dateien werden, falls notwendig, farblich konvertiert und im DMS-Dateiformat gespeichert. Das Modul für Grauwertbilder konvertiert Farbbilder in Graustufenbilder, das Modul für Schwarz/Weißbilder konvertiert Farb- und Graustufenbilder in Schwarz/Weißbilder.

Bilddateien beim Client

Hat der Benutzer alle Dateien zu den jeweiligen Seiten des Bilddokuments übernommen und legt das Dokument ab, speichert der Client die Dateien im benutzerspezifischen Cache.

Der Pfad und die Dateibezeichnungen haben folgende Struktur:

```
..\[benutzerid]\CACHE\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Seite]
```

[benutzerid] ist ein benutzerspezifisches Verzeichnis, das mit der hexadezimalen ID des Benutzers bezeichnet ist.

[Haupttyp] ist der hexadezimale Haupttyp des Dokuments.

[Untertyp] ist der hexadezimale Untertyp des Dokuments,
[mod 0x100] entspricht den zwei letzten Stellen der hexadezimalen Dokument-ID,
[DokID] ist die hexadezimale Dokument-ID,
[Seite] ist die Seitennummer, hexadezimal hochgezählt von 000.

Hat der Dokumenttyp die Eigenschaft 'Dias erzeugen', erzeugt der Client ebenfalls Quicklooks, kleine Seitenvorschauen der ersten Seite. Die Quicklook-Datei für ein Dokument wird im gleichen Verzeichnis wie die Bilddateien gespeichert und hat die gleiche Bezeichnung. Sie erhält die Endung DIA, das Bildformat ist TIF.

Beispiel für den Pfad und die Dateibezeichnung einer Bilddatei:

\Temp\ostemp\000000ED\CACHE\02\1B\36\00000136.001

\Temp\ostemp ist der Arbeitsbereich, \000000ED\CACHE ist der CACHE-Bereich des Benutzers mit der ID 237, 02 ist der Haupttyp (Schwarz-Weiß-Bild), 1B ist der Untertyp des Dokumenttyps, 36 sind die letzten beiden Stellen der Dokument-ID, 00000136 ist die eindeutige Dokument-ID des Dokuments, .001 ist die Dateiendung für die zweite Seite des Dokuments.

Bilddateien beim Server

Nachdem der Client das Dokument abgelegt hat, sendet er die Bilddateien und die Quicklook-Datei an den Server und löscht die Daten sowohl aus seinem Arbeitsbereich wie aus dem Cache. Der Server speichert die Bilddateien und die Quicklook-Datei im Work.

Pfad und die Dateibezeichnungen im Work haben die gleiche Struktur wie im Cache:

| |
|---------------------------------------------------------|
| \WORK\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Seite] |
|---------------------------------------------------------|

Tabellenstruktur beim Anlegen eines Bilddokuments

Analog zum Anlegen eines Registers erhält der Client vom Server eine ID für das Dokument und trägt die Verschlagwortung in die Tabelle `object#` ein. Hat der Benutzer das Dokument noch nicht abgelegt, ist seine Benutzer-ID in der Spalte `lockuser` verzeichnet. Das Dokument wird ebenfalls in der Tabelle `doclock` eingetragen. Zu diesem Zeitpunkt kann das Dokument von anderen Benutzern recherchiert werden, es hat aber noch keine Seiten und ist zum Bearbeiten gesperrt. Die Verschlagwortung ist nicht gesperrt und kann editiert werden.

Der Client trägt ebenfalls den Dokumentenstandort in die Tabelle `sdrel` ein.

Sobald der Benutzer das Dokument abgelegt hat und der Server das Dokument übernimmt, trägt der Server in die Tabelle die Anzahl der Seiten ein und die ID der Bereiche, in dem die Dateien und das Quicklook gespeichert sind. Die Daten aus der Tabelle `doclock` werden gelöscht. Das Dokument ist damit entsperrt.

Legt der Benutzer das Dokument nicht ab, sondern bricht die Aktion ab, werden die Sperr-Einträge in den Tabellen `object#`, `sdrel` und `doclock` gelöscht.

Beide Aktionen, das Anlegen des DMS-Objekts und das Erzeugen der Dokumentendateien, werden einzeln in der Tabelle `osobjhist` protokolliert und einzeln im Client in der Bearbeitungshistorie des Dokuments aufgeführt.

Scanner

Die Bearbeitungsmodule übernehmen Daten über die TWAIN- oder KOFAX-Schnittstelle aus Scannern. Ist der Scanner nach den Herstelleranweisungen installiert und der Zugriff erfolgreich getestet, können die Bearbeitungsmodule ebenfalls auf die Scanner zugreifen. Für den Test des Zugriffs über die KOFAX-Schnittstelle stellt KOFAX das Testprogramm VCDEMO.EXE zur Verfügung, der

Zugriff über die TWAIN-Schnittstelle kann beispielsweise mit dem Programm Imaging, Zubehör des Betriebssystems, getestet werden.

Bei der Installation von KOFAX-Komponenten wird der Pfad zum Komponentenverzeichnis `imgctls\bin` angegeben. Dieses Verzeichnis muss für die Bearbeitungsmodule über die Systemvariable PATH erreichbar sein. Ist das nicht der Fall, muss der Pfad zu diesem Verzeichnis nachgetragen werden. Für die Kommunikation über SCSI-Karten benötigen die Bearbeitungsmodule aktuelle ASPI-Treiber.

Scannereinstellungen

Scannereinstellungen werden benutzerspezifisch verwaltet. Über die Einstellungen im Client legt der Benutzer zusätzlich fest, ob die Scannereinstellungen arbeitsplatzspezifisch verwaltet werden sollen.

Die Daten werden in der Datenbank über die Tabelle `osconf` verwaltet. Dort sind benutzerspezifisch (Spalte `osconfuser`) unter der Konfigurationsbezeichnung `axscan` (Spalte `osconfname`) die Daten in der Spalte `osconfal` gespeichert.

Für den Zugriff auf diese Daten wird die Datei `axscan.ini` mit dem Inhalt des Datenbankfeldes erzeugt und im benutzerspezifischen Verzeichnis `\etc\user\Benutzername` gespeichert.

Die Datei enthält schnittstellen-, arbeitsplatz- und bearbeitungsmodulspezifische Abschnitte mit den Einstellungsdaten. Die KOFAX-Sektion enthält Verweise auf einzelne Konfigurationen, die ebenfalls in der Tabelle `osconf` gespeichert sind. Falls eine dieser Konfigurationen benötigt wird, werden die Daten ebenfalls in eine Datei `konfigurationsbezeichnung.ini` übertragen, im benutzerspezifischen Verzeichnis `\etc\user\Benutzername` gespeichert und für die Scannereinstellung verwendet.

Anlegen von Windows-Dokumenten

Windows-Dokumente sind Dokumente, die nicht mit Bearbeitungsmodulen des enaio® client, sondern mit externen Anwendungen, die der Client aufruft, erstellt werden. Für Windows-Dokumente werden Vorlagen eingerichtet, die der Benutzer auswählt und mit denen die verknüpften Anwendungen beim Anlegen von Dokumenten aufgerufen werden.

Über die W-Vorlagenverwaltung im enaio® administrator werden Vorlagen und Anwendung für Windows-Dokumenttypen verwaltet. Alle Vorlagen werden von enaio® server im Verzeichnis `\etc\Templates` gespeichert.

In der Datenbank werden die Daten der W-Vorlagenverwaltung in mehreren Tabellen gespeichert:

- § Vorlagen und ihre Verknüpfungen mit Anwendungen in der Tabelle `ostemplate`,
- § Vorlagen und ihre Verknüpfungen mit Dokumenttypen in der Tabelle `ostempobj`,
- § Vorlagen und ihre Verknüpfungen mit Gruppen in der Tabelle `ostempuser`.
- § Daten der Anwendungen in der Tabelle `osapplications`.

Die Tabelle `ostemplate` hat folgende Struktur:

| Spalte | Inhalt |
|---------------------|------------------------------------------------------------------------------------------------------------|
| <code>id</code> | die eindeutige ID der Vorlage |
| <code>app_id</code> | die eindeutige ID der Anwendung, die Daten zur Anwendung stehen in der Tabelle <code>osapplications</code> |

| | |
|-----------------|-------------------------------------------------------------------------------------------------------|
| viewer_id | die eindeutige ID des Viewers, die Daten zum Viewer stehen in der Tabelle <code>osapplications</code> |
| namespace_id | die ID des Namespace, die Namespacebezeichnung steht in der Tabelle <code>osnamespace</code> |
| alias_id | die ID des Alias, die Aliasbezeichnung steht in der Tabelle <code>osalias</code> |
| flags | Kennzeichen, ob die Vorlage im Template-Verzeichnis liegt |
| name | Dateiname der Vorlagen |
| extension | Dateiendung der Vorlage |
| checkoutdir | Verzeichnis, in das die Vorlage zum Bearbeiten geschrieben ist |
| checkoutstation | Arbeitsstation, an der die Vorlage bearbeitet wird |
| checkoutuser_id | Benutzer, der die Vorlage bearbeitet. |

Tabelle 15

Die Tabelle `ostempobj` hat folgende Struktur:

| Spalte | Inhalt |
|-----------|-------------------------------------------------|
| temp_id | die eindeutige ID der Vorlage |
| object_id | die ID des Dokumenttyps aus Haupt- und Untertyp |

Tabelle 16

Die Tabelle `ostempuser` hat folgende Struktur:

| Spalte | Inhalt |
|---------|----------------------------------------------------------------------------------------------|
| temp_id | die eindeutige ID der Vorlage |
| user_id | die eindeutige ID der zugeordneten Benutzergruppe oder eines zugeordneten Benutzers |
| flags | Kennzeichen, ob der Eintrag <code>user_id</code> eine Gruppe oder einen Benutzer bezeichnet. |

Tabelle 17

Die Tabelle `osapplications` hat folgende Struktur:

| Spalte | Inhalt |
|--------|-------------------------------------------------------------------------------|
| id | die eindeutige ID der Anwendung |
| flags | Kennzeichen, wie die Anwendung aufgerufen wird |
| name | Bezeichnung für die Anwendung |
| path | Pfad mit zugeordneter Laufwerksbezeichnung und Dateibezeichnung der Anwendung |

Tabelle 18

Der Client liest die Daten aus der Datenbank. Werden Daten der W-Vorlagenverwaltung im enaio® administrator geändert, muss der Client nicht neu gestartet werden.

Beispiel für die Daten zu einer Vorlage in der Datenbank:

Die Tabelle `ostemplate` enthält Vorlagendaten und die Anwendungszuordnung:

| id | app_id | viewer_id | namespace_id | alias_id | flags | name | extension | checkout |
|-----|--------|-----------|--------------|----------|-------|--------|-----------|----------|
| 300 | 139 | 139 | 240 | 241 | 0 | ef.doc | doc | |

Die Tabelle `osapplications` enthält die Anwendungsdaten für die Anwendung mit der ID 239.

| id | flags | name | path |
|-----|-------|--------|--------------------------------------------------|
| 139 | 2 | MSWord | C:\Programme\Microsoft Office\Office\WINWORD.EXE |

Die Tabelle `osnamespace` enthält die Bezeichnung des Namespaces mit der ID 240.

| id | name |
|-----|--------|
| 240 | Office |

Die Tabelle `osalias` enthält die Bezeichnung des Alias mit der ID 241.

| id | name |
|-----|---------------|
| 241 | Standardbrief |

Die Tabelle `ostempobj` enthält die Zuordnung der Vorlagen zu Dokumenttypen.

| temp_id | object_id |
|---------|-----------|
| 300 | 262284 |

Die Tabelle `ostempuser` enthält die Zuordnung der Vorlagen zu Benutzern.

| temp_id | user_id | flags |
|---------|---------|-------|
| 300 | 625 | 1 |

Dokumentendateien bei Windows-Dokumenten

Im Unterschied zu Bilddokumenten, werden für Windows-Dokumente Vorlagen verwendet und für ein Windows-Dokument wird genau eine Datei im Anwendungsformat im Archiv verwaltet. Legt ein Benutzer ein Windows-Dokument an, verschlagwortet er es zuerst und der Client trägt die Daten, wie bei Bilddokumenten, in die entsprechenden Tabellen ein. Dann wählt der Benutzer eine Vorlage aus.

Wählt der Benutzer keine Vorlage, sondern bricht an dieser Stelle ab, bleiben die Verschlagwortungsdaten als ein Dokument ohne Seiten erhalten.

enaio® server übernimmt die Vorlage aus dem Vorlagenverzeichnis `\templates`, überträgt sie zum Client, der sie im CACHE-Bereich speichert und zum Server zurücksendet. Der Server speichert die Vorlage als Dokument im WORK-Bereich. Pfad und die Dateibezeichnungen haben die gleiche Struktur wie bei Bilddokumenten, der Haupttyp von Windows-Dokumenten ist 4, die Dateiendung entspricht der Vorlage.

| |
|----------------------------------------------------------------------------|
| <code>\WORK\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Dateiendung]</code> |
|----------------------------------------------------------------------------|

Pfad und die Dateibezeichnungen beim Client haben ebenfalls diese gleiche Struktur:

| |
|--------------------------------------------------------------------------------------------|
| <code>..\[benutzerid]\CACHE\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Dateiendung]</code> |
|--------------------------------------------------------------------------------------------|

Der Client startet die Anwendung, wie in enaio® administrator bei der Konfiguration der Anwendungen angegeben, mit einem Aufruf des Programms über den angegebenen Pfad oder über die Anwendung, die für die Dateiendung registriert ist.

Speichert der Benutzer das Dokument, wird es im CACHE-Bereich gespeichert. Checkt der Benutzer das Dokument ein, wird es zum Server übertragen, vom Server im WORK-Bereich gespeichert und für andere Benutzer zum Bearbeiten freigegeben. Aus dem CACHE-Bereich des Clients wird es gelöscht.

Tabellenstruktur beim Anlegen eines Windows-Dokuments

Analog zum Anlegen der Bilddateien erhält der Client vom Server eine ID für das Dokument und trägt die Verschlagwortung in die Tabelle `object#` ein. Die Seitenzahl ist '0', der Benutzer ist in der Spalte `lockuser` verzeichnet. In der Tabelle `doclock` sind ebenfalls Dokumentendaten und Benutzerdaten zur Kennzeichnung des Auscheckens eingetragen.

Der Dokumentenstandort wird in der Tabelle `sdrel` eingetragen.

Wählt der Benutzer eine Vorlage aus, fordert der Client die Vorlage vom Server an, speichert sie im CACHE-Bereich und überträgt sie zum Server, der das Dokument mit der Vorlage aus dem Vorlagenverzeichnis anlegt und in der Tabelle `object#` die Seitenzahl '1' und die Medien-IDs einträgt. Der Client startet mit dem Dokument die Anwendung. Checkt der Benutzer das Dokument ein, wird es zum Server übertragen, im WORK-Bereich gespeichert und aus dem CACHE-Bereich gelöscht.

In der Tabelle `object#` wird der Benutzer ausgetragen, die Einträge in der Tabelle `doclock` werden gelöscht. Das Dokument ist damit entsperrt. Das Anlegen des DMS-Objekts, das Erzeugen der Dokumente aus der Vorlage beim Server und das Öffnen zum Bearbeiten mit der Anwendung werden als einzelne Aktionen in der Tabelle `osobjhist` protokolliert und einzeln im Client in der Bearbeitungshistorie des Dokuments aufgeführt.

Übernahme von Windows-Dokumenten

Statt eine Vorlage zu wählen, kann der Benutzer eine Datei aus dem Dateisystem übernehmen und dabei wählen, ob die Anwendung geöffnet werden soll. Auswählen kann der Benutzer nur Dateien, deren Endung einer Vorlage entspricht, die dem Dokumenttyp zugeordnet ist.

Wählt der Benutzer eine Datei, wird die Datei vom Client übernommen und zum Server übertragen. Der Server speichert die Datei im WORK-Bereich. Lässt der Benutzer das Dokument öffnen, überträgt der Server das Dokument wieder zum Client und checkt es aus. Der Client öffnet das Dokument mit der Anwendung, die der Endung zugeordnet ist.

Übernahme von E-Mails

Benutzer können E-Mails auf folgende Arten in das Archiv übernehmen:

- § aus dem Postfach
- § über Drag&Drop aus Microsoft Outlook
- § über enaio® document-storage

Der Dokumenttyp E-Mail wird, wie die anderen Dokumenttypen auch, mit dem enaio® editor eingerichtet. Die Daten werden in den Tabellen `osobjdef` und `osobjfields` gespeichert, die Verschlagwortung in der entsprechenden Tabelle `object#`.

Für die Übernahme von E-Mails aus dem Postfach unterstützt enaio® alle MAPI 1.0-konformen Windows-Mailsysteme, sowie IMAP-Systeme, bei denen der Versand über SMTP/MAPI erfolgt.

Über enaio® enterprise-manager können die Postfächer systemweit zwischen MAPI und IMAP zur Kommunikation mit dem Mailserver umgeschaltet werden. Die Daten werden im Format MIME erwartet und auch in diesem Format im Archiv verwaltet.

Der Client ermittelt über die benutzerspezifischen Einstellungen, ob Benutzer das Postfach verwenden. Diese Einstellung wird benutzerspezifisch in der Tabelle `osconf` gespeichert. In der Tabelle werden für jeden Benutzer (Spalte `osconfuser`)

als Typ 3 (Spalte `osconftype`) in der Spalte `osconfval` benutzerspezifische Einstellungen gespeichert, die der Client beim Start aus der Datenbank liest. Für die Initialisierung des Posteingangs steht in der Sektion `[CLIENT]` der Eintrag `MAPIINIT`. Beim Wert 1 wird der Posteingang initialisiert, beim Wert 0 nicht.

Angezeigt wird immer der Posteingang des an das Betriebssystem angemeldeten Benutzers.

Systemweit kann die Initialisierung der Posteingänge über einen Eintrag in der Konfigurationsdatei `as.cfg` festgelegt werden. Der Pfad zu dieser Datei ist in der Tabelle `osresources` eingetragen. Die Datei kann mit einem beliebigen Editor editiert werden. Der Eintrag steht in der Sektion `[SYSTEM]` und lautet ebenfalls `MAPIINIT`. Der Default-Wert ist 1. Die Postfächer sind also standardmäßig aktiviert. Allerdings muss dazu das System ebenfalls entsprechend lizenziert sein.

E-Mails enthalten Kennungsdaten, beispielsweise 'Von', 'An', 'Betreff'. Diese Daten können Verschlagwortungsfeldern des Dokumenttyps E-Mail zugeordnet werden. Die Zuordnung in dieser Konfigurationsdatei kann auch über die internen Namen erfolgen.

Postfach

Die Zuordnung bei der Übernahme von E-Mails aus dem Postfach erfolgt automatisch über die Vergabe von internen Namen beim Dokumenttyps E-Mail im enaio® editor.

Die Felder benötigen folgende Bezeichnung:

| Kenndaten | Interner Feldname |
|-----------|-------------------|
| Von | MAIL_FROM |
| An | MAIL_TO |
| CC | MAIL_CC |
| BCC | MAIL_BCC |
| Datum | MAIL_SUBMIT_TIME |
| Betreff | MAIL_SUBJECT |
| Nachricht | MAIL_BODY |

Tabelle 19

Groß/Kleinschreibung wird dabei nicht beachtet, durch die Verwendung von internen Namen können die Beschriftungen auf der Datenmaske beliebig sein. Werden für diese Kenndaten keine entsprechenden Verschlagwortungsfelder eingerichtet, werden die Daten nicht für die Verschlagwortung verwendet, aber im E-Mail-Modul angezeigt.

Drag&Drop aus Microsoft Outlook

Die Zuordnungen für die Übernahme von E-Mails über Drag&Drop aus Microsoft Outlook werden ebenfalls über die vergebenen internen Namen festgelegt. Aus Kompatibilitätsgründen mit früheren Versionen können diese Zuordnungen jedoch auch aus der Konfigurationsdatei `as.cfg` gelesen werden. Sind dort diese Einstellungen festgelegt, dann werden Zuordnungen über interne Namen nicht ausgewertet. (Beim Start des Clients legt der Server eine Kopie der Datei `as.cfg` in das Verzeichnis `\Temp\ostemp\'ServerName\'#\'Port\'etc\.`) Sind keine Zuordnungen zwischen E-Mail und Verschlagwortungsfeldern eingerichtet, so werden die Daten nicht für die Verschlagwortung verwendet, aber im E-Mail-Modul angezeigt.

Dokumentendateien bei E-Mail-Dokumenten

Wie bei den anderen Dokumenttypen, übernimmt der Client eine E-Mail in den Arbeitsbereich und überträgt sie nach der Verschlagwortung in den CACHE-Bereich.

Pfad und die Dateibezeichnungen haben folgende Struktur:

```
..\[benutzerid]\CACHE\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Dateiendung]
```

Die Dateiendung von E-Mails lautet `mim`, `ima` oder `msg`.

Danach überträgt der Client die Datei zum Server.

Pfad und die Dateibezeichnungen beim Server haben folgende Struktur:

```
\WORK\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Dateiendung]
```

Nach der Übertragung wird die Datei aus dem Arbeitsbereich und aus dem Cache gelöscht.

Tabellenstruktur bei der Übernahme von E-Mails

Wie bei anderen Dokumenttypen erhält der Client nach der automatischen oder manuellen Verschlagwortung der E-Mail vom Server eine ID für das Dokument und trägt die Verschlagwortung in die Tabelle `object#` ein. Der Dokumentenstandort wird in der Tabelle `sdrel` eingetragen. Danach wird das Dokument im WORK-Bereich gespeichert und aus dem CACHE-Bereich gelöscht.

Beide Aktionen, das Anlegen des DMS-Objekts in der Datenbank und das Speichern der Dokumentendateien, werden einzeln in der Tabelle `osobjhist` protokolliert und einzeln im Client in der Bearbeitungshistorie des Dokuments aufgeführt.

enaio® document-storage

enaio® document-storage stellt Funktionen zur Verfügung, die es ermöglichen, E-Mails direkt aus Outlook, GroupWise und Lotus Notes heraus in enaio® zu übernehmen.

Die Funktionen sind über eine Symbolleiste in der E-Mail-Anwendung eingebunden. Für die Konfiguration steht eine eigene übersichtliche Oberfläche zur Verfügung.

enaio® document-storage erlaubt es darüber hinaus, E-Mails auf verschiedene Arten automatisch zu kennzeichnen, Anlagen bei der Archivierung mit einzubeziehen oder auch auszuschließen und vieles mehr.

Zur Konfiguration von enaio® document-storage wird die Systemrolle 'Editor: Starten' benötigt.

Über die Konfigurationsverwaltung wird festgelegt, welchen Objekttypen in enaio® E-Mails zugeordnet sind und welche Kenndaten als Verschlagwortungsdaten übernommen werden. Festgelegt wird ebenfalls, wie Anlagen behandelt werden. Anlagen können als eigene Dokumente verwaltet werden. Dabei kann überprüft werden, ob eine Anlage bereits in enaio® vorliegt und – falls das der Fall ist – statt diese wiederholt zu speichern, ein Verweis erzeugt werden.

Weiter können mit der Übernahme von E-Mails Workflowvorgänge gestartet werden. Dabei können die Kenndaten als Vorgangsvariablen übergeben werden.

Die Konfiguration - `axvbdocstorage.xml` - wird im Verzeichnis `\etc` des Datenverzeichnisses gespeichert. In Umgebungen mit mehreren Servergruppen und damit Datenverzeichnissen kann die Konfigurationsdatei kopiert und in die anderen etc-Verzeichnisse übertragen werden.

Funktionsübersicht

| Funktion | Microsoft Outlook (ab 2007) | Novell Group-Wise (ab v7.01) | IBM Lotus Notes (ab v6.5.6) |
|-----------------------------------------------------|-----------------------------|------------------------------|-----------------------------|
| Separate Ablage der Anlagen erlauben | Ü | Ü | Ü |
| Ablage von E-Mails ohne Anlagen erlauben | Ü | Ü | Ü |
| In Archivierungsordner verschieben | Ü | | |
| Objekt archivieren und löschen | Ü | | |
| Anlagen entfernen | Ü | | Ü |
| OS-Linkdatei einfügen | Ü | | Ü |
| Archivierungs-ID-Kennung in Body einfügen | Ü | | Ü |
| Archivieren, senden und Outlook-Ablageort ermitteln | Ü | | |
| Bei erkanntem Ablageort automatisch ablegen | Ü | | |
| Objekt im MSG-Format speichern | Ü | | |
| Signierte E-Mails im MSG-Format speichern | Ü | | |
| E-Mail-Eigenschaften kopieren | Ü | Ü | |
| Objekt aktualisieren | Ü | | |
| Mit dem enaio-Client archivieren | Ü | Ü | Ü |
| Einzel übernehmen | Ü | Ü | Ü |
| Adressauflösung via LDAP | Ü | | Ü |
| Kooperationsmodus verwenden | Ü | | |
| Verweise erzeugen | Ü | | |
| Anlagen über Notizen verbinden | Ü | Ü | Ü |
| Deduplizierung | Ü | Ü | Ü |

Übernahme von Filmdokumenten

Mit dem Modul für Filmdokumente können Filme im Format MPEG oder AVI aus dem Dateisystem übernommen werden. Sie werden im gegebenen Format im Archiv verwaltet, erhalten aber im Arbeitsbereich die Endung ASM. und im CACHE-

und WORK-Bereich die Endung 000. Jedem Filmdokument kann nur eine Datei zugeordnet werden. Werden für den Dokumenttyp Dias erzeugt, wird das erste Bild extrahiert und im Format TIF mit der Dateiendung DIA gespeichert. Dateiverwaltung und Tabellenstruktur entsprechen denen der Bilddokumente. Der Haupttyp von Filmdokumententypen ist 5.

Zur Anzeige der Filme im Client muss der Windows Media Player auf dem Arbeitsplatzrechner installiert sein.

Übernahme von XML-Dokumenten

XML-Dokumente werden entweder importiert oder im Client über einen Dateiauswahldialog aus dem Dateisystem übernommen. Einem XML-Dokument kann nur eine Datei zugeordnet werden. Die Dateien werden direkt übernommen und nicht angezeigt. Für die Anzeige werden XML-Dokumententypen Stylesheets zugeordnet.

Legt ein Benutzer ein XML-Dokument an, verschlagwortet er es zuerst und der Client trägt die Daten in die Verschlagwortungstabelle `object#` ein und die Standortdaten in die Tabelle `sdrel`. Dann wählt der Benutzer über den Dateiauswahldialog eine XML-, XSL- oder XSLT-Datei aus. Bricht der Benutzer ab, werden die Daten wieder gelöscht. Die gewählte Datei wird in den Arbeitsbereich kopiert, von dort in den CACHE-Bereich und zum Server übertragen. Der Server speichert die Datei im WORK-Bereich. Dateiverwaltung und Tabellenstruktur entsprechen denen der anderen Dokumente. Der Haupttyp von XML-Dokumententypen ist 7.

Übernahme über das Dateisystem

Dateien für Windows-Dokumente, Bilddokumente und Filmdokumente können Benutzer direkt aus dem Dateisystem, beispielsweise dem Windows-Explorer, in einen geöffneten Ordner ziehen. Der Client kopiert die Datei in den Arbeitsbereich und erstellt anhand der Dateiendung eine Liste mit Dokumententypen, denen die Datei zugeordnet werden kann. Die Daten für die Zuordnung zu Windowsdokumententypen werden aus der Tabelle `ostemplate` gelesen.

Kann die Endung keinem Bearbeitungsmodul und keiner Windows-Anwendung zugeordnet werden, erhält der Benutzer eine Fehlermeldung.

Verschlagwortet der Benutzer das Dokument, trägt der Client die Daten in die Tabelle `objekt#` ein, den Standort in die Tabelle `sdrel`, kopiert die Datei in den Cache und überträgt sie zum Server.

Pfad und die Dateibezeichnungen beim Server haben folgende Struktur:

```
\WORK\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Dateiendung]
```

Pfad und die Dateibezeichnungen beim Client haben folgende Struktur:

```
..\[benutzerid]\CACHE\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Dateiendung]
```

Die Datei wird nicht geöffnet. Das Anlegen wird in der Tabelle `osobjhist` protokolliert.

Übernahme über enaio® printer

Bei der Installation von enaio® können Druckertreiber mit installiert werden, über die Benutzer aus jeder Anwendung mit Druckfunktion Dateien als Bilddokument oder als PDF-Dokument in das DMS übernehmen kann. PDF-Dokumente sind in das PDF-Format eingebundene TIFFs.

Der enaio® printer(SW) druckt Schwarz/Weiß im Format TIF G4 oder PDF, der enaio® printer(Color) druckt farbig im Format JPEG oder PDF. Schwarz/Weiß-Bilder werden einem Schwarz/Weiß-Dokumenttyp zugeordnet, Farbbilder einem Farbbild-Dokumenttyp und PDF-Dokumente einem Windows-Dokumenttyp.

Ist einem Windows-Dokumenttyp keine PDF-Anwendung zugeordnet, können PDF-Dokumente trotzdem zugeordnet werden. Der Benutzer erhält einen Hinweis. Die PDF-Dokumente können dann nur über die registrierte Anwendung schreibgeschützt geöffnet werden.

Der Schwarz/Weiß-Drucker kann zusätzlich Hintergrundbilder verwenden. Diese Hintergrundbilder müssen als Schwarz/Weiß-Bitmaps im Format TIFF G4 vorliegen und die gleiche Größe haben, wie die Dokumentenseiten. Format und Hintergrundbilder werden über enaio® administrator verwaltet. Wird das Format PDF gewählt, können Benutzer über ihre benutzerspezifischen Einstellungen einen Bestätigungsdialog einschalten, über den Sie statt PDF das entsprechende Bildformat wählen können.

enaio® administrator speichert die Einstellungen in der Konfigurationsdatei `as.cfg`. Der Pfad zu dieser Datei ist in der Tabelle `osresources` eingetragen.

Die Einträge in der Datei haben folgende Struktur:

```
[ASPRINT]
HINTERGRUND0101=1.Brief,C:\HG_Bilder\1brief.tif
HINTERGRUND0102=ff Brief,C:\HG_Bilder\2brief.tif
HINTERGRUND0201=1.Auftrag,C:\HG_Bilder\lauftrag.tif
HINTERGRUND0202=ff Auftrag,C:\HG_Bilder\2auftrag.tif
DDOCFORMAT=0
PDOCFORMAT=1
```

Die Sektion `[ASPRINT]` enthält eine Liste der Hintergrundbilder. Bilder mit 1 als letzter Ziffer werden für die erste Seite angeboten, Bilder mit 2 als letzter Ziffer für die folgenden Seiten. Die Pfadangaben werden mit zugeordneter Laufwerksangabe gespeichert. Alle Benutzer müssen genau so auf die Bilder zugreifen können.

Der Wert für `DDOCFORMAT` legt das Format für den Schwarz/Weiß-Drucker fest, 0 entspricht TIF, 1 entspricht PDF. Der Wert für `PDOCFORMAT` legt das Format für den Farb-Drucker fest, 0 entspricht JPEG, 1 entspricht PDF.

Die Druckertreiber lesen diese Daten aus der `as.cfg`, die beim Start des Clients in das Verzeichnis `\Temp\ostemp\'ServerName\'#\'Port\' \etc\` geschrieben wird. Den Pfad zu dieser Datei erhalten sie vom Client. Ist der Client nicht gestartet, erhalten die Benutzer eine Fehlermeldung.

Die Druckertreiber übergeben dem Client Pfad und Dateibezeichnung der Dateien. Der Benutzer legt den Standort im DMS fest, einen geöffneten Ordner oder die Ablage, gibt den Dokumenttyp an und verschlagwortet das Dokument. Der Client trägt die Daten, Verschlagwortung und Standort, in die Datenbank ein und überträgt die Dokumentendateien zum Server. Dateiverwaltung und Tabellenstruktur entsprechen denen der anderen Dokumente.

enaio® office-utilities

Über die enaio® office-utilities können Benutzer ebenfalls auf die Druckertreiber zugreifen und das Format wählen. Sind Hintergrundbilder eingerichtet, werden diese ebenfalls angeboten. Die enaio® office-utilities erzeugen die Konfigurationsdatei `asprint.ini` im Verzeichnis `WINNT`. In dieser Datei ist das vom Benutzer gewählte Format hinterlegt. Die Druckertreiber suchen nach dem Start zuerst nach einer Konfigurationsdatei und übernehmen die Formateinstellung aus der Datei.

Der Standort und die Verschlagwortung werden dann wie bei der direkten Wahl des Druckertreibers angegeben.

Modulübergreifende Dokumente

Dokumenttypen können im enaio® editor die Eigenschaft 'modulübergreifend' erhalten. Dann wählt der Benutzer nach dem Verschlagworten das Bearbeitungsmodul aus. Diese Eigenschaft wird über die Spalte `flags` des ersten Dialogelements des DMS-Objekts in der Tabelle `osobjfields` verwaltet. Modulübergreifende Dokumenttypen haben einen Haupttyp, je nachdem, welcher Haupttyp beim Anlegen im enaio® editor gewählt wurde. Dieser Haupttyp wird in der Spalte `haupttyp` der Tabelle `object#` eingetragen. Das Anlegen und die Tabellenstrukturen entsprechen denen der anderen Dokumenttypen.

Dokumente ohne Seiten

Dokumente ohne Seiten sind Dokumente eines Dokumenttyps, die nur aus der Verschlagwortung bestehen und denen keine Dateien zugeordnet sind. Dokumenttypen können im enaio® editor die Eigenschaft 'ohne Seiten' erhalten. Dann kann kein Benutzer Dateien zuordnen. Diese Eigenschaft wird über die Spalte `flags` des ersten Dialogelements des DMS-Objekts in der Tabelle `osobjfields` verwaltet. Benutzer können Dokumente ohne Seiten anlegen, wenn Sie nur Verschlagwortungsdaten speichern wollen oder erst später Dateien zuordnen wollen. Dann wird in die Spalte `anzahl` der Tabelle `object#` der Wert '0' eingetragen und kein Bearbeitungsmodul geöffnet.

Die Ablage

Benutzer können Dokumente entweder in einen geöffneten Ordner oder ein Register übernehmen oder in die Ablage legen. Die Ablage ist ein benutzerspezifischer Bereich, auf den nur der Benutzer zugreift. Objekte in der Ablage werden bei einer Recherche nicht berücksichtigt. Über den Menüpunkt 'Benutzerablagen einsehen' im enaio® administrator kann eine Inhaltsliste der Ablage jedes Benutzers eingesehen werden. Wird ein Benutzer im enaio® administrator gelöscht, kann der Inhalt der Ablage in die Ablage des Benutzers verschoben werden, der den Benutzer löscht. Dokumente, die ein Benutzer für die Ablage erzeugt, werden einem Dokumenttyp zugeordnet oder ohne Typ angelegt.

Dokumente mit Typ

Dokumente mit Dokumenttyp werden verschlagwortet, die Verschlagwortung wird in die Tabelle `object#` eingetragen. Der Standort wird nicht in die Tabelle `sdrel` eingetragen, sondern in die Tabelle `mdrel`. Diese Tabelle enthält alle Dokumente aus Ablagen und alle Dokumente, die in Mappen liegen und hat folgende Struktur:

| Spalte | Inhalt |
|--------------------------|------------------------------------------------------------|
| <code>loeschen</code> | zur Zeit nicht verwendet |
| <code>zeitstempel</code> | der genaue Zeitpunkt, zu dem das DMS-Objekt angelegt wurde |
| <code>mappe_id</code> | die eindeutige ID der Ablage/Mappe |
| <code>object_id</code> | die eindeutige ID des Dokuments |
| <code>stamm_id</code> | Ordner-ID (Mappe) |
| <code>objekttyp</code> | die ID des Dokumenttyps |

| | |
|----------|-------------------------------------------|
| modul | Haupttyp von Dokumenten ohne Typ |
| eingang | zur Zeit nicht verwendet |
| absender | der Anleger von Dokumenten ohne Typ |
| ausgang | zur Zeit nicht verwendet |
| anzahl | Anzahl der Seiten von Dokumenten ohne Typ |

Tabelle 20

Die Ablagen und Mappen werden über die Tabelle `mappe` verwaltet:

| Spalte | Inhalt |
|---------------|-------------------------------------------------------|
| id | die eindeutige ID der Mappe/Ablage |
| loeschen | zur Zeit nicht verwendet |
| zeitstempel | der genaue Zeitpunkt, zu dem Mappen angelegt wurden |
| angelegt | das Anlagedatum von Mappen |
| anleger | der Anleger von Mappen |
| empfaenger | der Empfänger von Mappen oder der Besitzer der Ablage |
| thema | das Thema von Mappen |
| typ | Kennzeichnung, ob eine Mappe öffentlich ist |
| links | Kennzeichnung für Mappen, ob Notizen existieren |
| anleger_id | zur Zeit nicht verwendet |
| empfaenger_id | zur Zeit nicht verwendet |

Tabelle 21

Die Dateiverwaltung von Dokumenten aus der Ablage entspricht denen der Dokumente aus Ordnern. Pfad und die Dateibezeichnungen beim Server haben folgende Struktur:

```
\WORK\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Dateiendung]
```

Pfad und die Dateibezeichnungen beim Client haben folgende Struktur:

```
..\[benutzerid]\CACHE\[Haupttyp]\[Untertyp]\[mod 0x100]\[DokID].[Dateiendung]
```

Dokumente ohne Typ

Dokumente, die Benutzer ohne Typ für die Ablage anlegen, werden nicht verschlagwortet. Der Benutzer wählt ein Bearbeitungsmodul und übernimmt Dateien. Der Client speichert die Dateien im Arbeitsbereich. Legt der Benutzer das Dokument ab, speichert der Client die Dateien im Cache und überträgt sie zum Server.

| Windows-Dokumente können nicht ohne Typ in der Ablage angelegt werden. |

Pfad und die Dateibezeichnungen beim Server haben folgende Struktur:

```
\WORK\[Haupttyp]\[DokID].[Dateiendung]
```

Pfad und die Dateibezeichnungen beim Client haben folgende Struktur:

```
..\[benutzerid]\CACHE\[Haupttyp]\[DokID].[Dateiendung]
```

Dokumente ohne Typ sind in der Datenbank nur in der Tabelle `mdrel` eingetragen. Das Anlegen von Dokumenten in der Ablage wird in der Tabelle `osobjhist` protokolliert.

Sollen Dokumente ohne Typ von der Ablage in das System verschoben werden, so müssen sie erst einem Dokumenttyp zugeordnet und entsprechend verschlagwortet werden.

In der Ablage wird das Rechtesystem beachtet. Legt ein Benutzer ein Dokument an, das er durch Einschränkungen der Zugriffsrechte nicht löschen kann, kann er es ebenfalls nicht aus seiner Ablage löschen.

Arbeiten mit Dokumenten

Allgemeine Voraussetzungen

Für das Arbeiten mit Dokumenten benötigt der enaio® client die aktuellen Objektdefinitionen, Objektrelationen und Konfigurations-Einträge, die er beim Start liest. Er benötigt ebenfalls die aktuellen Rechte des angemeldeten Benutzers. Diese Daten können im enaio® client über 'Umschalttaste-Strg-F5' aktualisiert werden.

DMS-Objekte sind in eine hierarchische Struktur eingebunden und werden verschlagwortet. Die Benutzer benötigen grundlegende Kenntnisse der hierarchischen Struktur und der objektspezifischen Verschlagwortungsstruktur, um zielgenau nach DMS-Objekten recherchieren zu können.

Benutzer recherchieren nach DMS-Objekten

- § indem sie über Recherchemasken Suchkriterien für die Verschlagwortung und die Basisparameter angeben und Trefferlisten erhalten
- § indem sie der hierarchischen Struktur folgen, also Ordner und Register öffnen und Inhaltsverzeichnisse einsehen
- § indem sie mit Suchausdrücken über den Volltextindex suchen, der sich über die Verschlagwortung und den Dokumenteninhalt erstrecken kann, und ebenfalls Trefferlisten erhalten
- § indem sie gespeicherte Anfragen ausführen und Trefferlisten erhalten

Für eine Recherche erstellt der enaio® client aus den Benutzerangaben und den Daten aus dem Rechtesystem SQL-Statements, die an enaio® server gesendet werden, der die Statements parst und die Datenbank anfragt.

Als Parser wird der allgemeine Parser `oxtrodbc.dll` verwendet. Für die Datenbank Oracle kann ebenfalls der Parser `oxoratnl.dll`, für Microsoft SQL Server der Parser `oxmstnl.dll` verwendet werden.

Für eine Volltextrecherche wird der Volltextindex-Server angefragt.

Der Client erhält eine Liste von DMS-Objekten, die er als Trefferliste oder in einem Ordnerfenster als Inhaltsliste anzeigt. Benutzer können Recherchestruckturen konfigurieren, speichern und anderen Benutzern zur Verfügung stellen. Die Sicht auf DMS-Objekte in Trefferlisten und Ordnerfenstern kann konfiguriert werden. DMS-Objekte können für den schnellen Zugriff in individuellen und öffentlichen Bereichen zusammengestellt werden.

Der enaio® client verwaltet benutzerspezifische Einstellungen über die Tabelle `osconf`.

Die Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|-------------------------|----------------------------|
| <code>osconfuser</code> | die eindeutige Benutzer-ID |

| | |
|-------------|---------------------------------------------------------------------------------------|
| osconfstate | zur Zeit nicht verwendet |
| osconftype | der Konfigurationstyp |
| osconfval | die Konfigurationsdaten |
| osconfid | die eindeutige ID einer Konfiguration |
| osconflink | die eindeutige Benutzer-ID des zugeordneten Profils |
| osconfname | Dateibezeichnung, falls Konfigurationsdaten als Datei an Komponenten übergeben werden |

Tabelle 22

Die folgenden Konfigurationstypen werden für Benutzereinstellungen verwendet:

| osconftype | Inhalt |
|------------|-----------------------------------------------------------------|
| 1 | gespeicherte Anfragen |
| 2 | Verknüpfungen zu externen Programmen |
| 3 | Voreinstellungen, Fensterpositionen, zwischengespeicherte Daten |
| 4 | Ordnerstruktur in der Objektsuche |
| 5 | SQL-Anfragen |
| 7 | DMS-Objekte in der Objektsuche |

Tabelle 23

Falls einem Benutzer ein Profil zugeordnet wird, werden für den Benutzer die Daten dieser Konfigurationstypen vom Profil auf den Benutzer übertragen und in die Datenbank eingetragen.

Hat ein Benutzer nicht die Systemrolle 'eigene Einstellungen speichern', kann er die Daten dieser Konfigurationstypen im Client zwar temporär ändern, der Client schreibt Änderungen aber nicht in die Datenbank zurück.

Nicht benutzerspezifische Einstellungen werden über die Datei `as.cfg` verwaltet. Der Pfad zu dieser Datei ist in der Tabelle `osresources` eingetragen. Beim Start des Clients legt der Server eine Kopie der Datei `as.cfg` in das Verzeichnis

`\Temp\ostemp\'ServerName\'#\'Port\'\etc\.`

Der Client liest die Einstellungen aus dieser Datei.

Fensterpositionen werden von Clientkomponenten in den Benutzereinstellungen in der `osconf` eingelesen. Diesen Benutzereinstellungen sind Einträge des Konfigurationstyps 3 zugeordnet.

Falls Benutzereinstellungen fehlerhaft im System gespeichert wurden, können Benutzer beim Start die Taste F8 gedrückt halten. Der Client liest dann die gespeicherten Fensterpositionen nicht aus, sondern initialisiert sich mit Default-Einstellungen.

Im Folgenden werden nur einige Aspekte in Hinblick auf das Systemverständnis erklärt. Eine genaue Anleitung für das Arbeiten mit Dokumenten finden Sie im enaio® client-Handbuch.

Recherchieren

Für eine Recherche erstellt der enaio® client aus den Benutzerangaben und den Daten aus dem Rechtesystem SQL-Statements mit denen er die Datenbank anfragt. Anfragen über den Volltextindex richtet er an den Volltextindex-Server.

Die Recherche ist eingegrenzt auf DMS-Objekte, die innerhalb der hierarchischen Struktur einem Ordnertyp zugeordnet sind. Übergreifende Recherchen sind nur mit Erweiterten Anfragen möglich.

Für die Recherche stellt der Benutzer ein, ob der Platzhalter für beliebig viele Zeichen automatisch vor oder nach den Suchkriterien eingesetzt wird. Diese benutzerspezifische Einstellung wird in der Tabelle `osconf` mit den Daten des Konfigurationstyps `3` gespeichert. Der Eintrag lautet: `Autostern='Wert'`

Suchkriterien in einer Recherchemaske werden immer durch das logische UND verknüpft.

Die Verschlagwortung des Dialogelements `Tabelle` steht für die Recherche nicht zur Verfügung.

Beispiel für eine Recherche:

Der Benutzer öffnet eine Dokumenten-Recherchemaske und trägt Suchkriterien ein.

Abbildung 21 Beispiel für eine Recherchemaske

Als Voreinstellung wird der Platzhalter für beliebig viele Zeichen vor und hinter die Einträge eingesetzt.

Der enaio® client bildet folgendes Statement:

```
select      o.feld1,o.feld3,o.feld4,o.id,o.links      from      objekt2      o
where ( {fn UCASE(o.feld1)} like '%Mustermann%' and
({fn UCASE(o.feld6)} < '300'));
```

Angefragt wird die Verschlagwortung der Felder, die in der Trefferliste angezeigt werden sollen. Anfragekriterium ist der Eintrag im Feld `Autor` (Dokument-Feld 1). Das zweite Anfragekriterium für das Feld `Kopienanzahl` (Dokument-Feld 6) stammt aus einer Klausel des Rechtesystems. Der anfragende Benutzer darf hier nur Dokument-Verschlagwortungen einsehen, bei denen die Anzahl der Kopien kleiner als 300 ist.

Trefferlisten

Falls Trefferlisten genau ein DMS-Objekt enthalten, kann dieses gleich geöffnet werden. Dies ist eine benutzerspezifische Einstellung und wird ebenfalls in der

Tabelle `osconf` mit den Daten des Konfigurationstyps 3 gespeichert. Bei einer reinen Volltextanfrage werden alle Treffer - Ordner, Register und Dokumente - in einer Trefferliste aufgeführt. Es wird nicht unterschieden, ob der Suchbegriff Bestandteil der Verschlagwortung oder des Inhalts ist.

Über die Verschlagwortungstabelle `objectn` ermittelt der Client, ob Dokumente von anderen Benutzern bearbeitet werden und damit gesperrt sind. Ist das der Fall, werden diese Dokumente in der Trefferliste gekennzeichnet.

Ordnerfenster

Für das Öffnen von Ordnern und Registern erzeugt enaio® client SQL-Statements, mit denen alle DMS-Objekte, deren Standort dem gewählten Ordner oder Register zugeordnet ist, angefragt werden. Objektrelationen werden dabei nicht beachtet. Über die Verschlagwortungstabelle `objectn` ermittelt der Client, ob Dokumente von anderen Benutzern bearbeitet werden und damit gesperrt sind. Ist das der Fall, werden diese Dokumente in der Trefferliste gekennzeichnet.

Filter werden mit den benutzerspezifischen Einstellungen für die Gestaltung der Ordnerfenster in der Tabelle `osconf` mit den Daten des Konfigurationstyps 3 gespeichert und können über die Profilverwaltung Benutzern zugeordnet werden.

Der Eintrag für Filter hat folgende Struktur:

```
[FILTER@2]
Configs=Rechnungen,Belege
[FILTER@2|Rechnungen]
...
```

Für jeden Ordnertyp wird eine Sektion `[FILTER@'Untertyp']` eingerichtet, in der die eingerichteten Filter aufgelistet sind.

Für jeden Filter wird eine Sektion `[FILTER@'Untertyp'|'Filterbezeichnung']` eingerichtet, der die Filterkriterien folgen.

Beispiel für einen Filter

Konfiguration eines Filters im enaio® client:

Abbildung 22 Konfiguration von Filterkriterien

Eintrag in der Datenbank:

```
[FILTER@1 | W-Doks ]
TypesToShow=262298 | 262284 | 262317 |
CreatedByUser=OSWEB
SortColumns=2 | 3 |
CHECK1=0
CHECK2=1
CHECK3=1
CHECK4=0
Active=0
TimeFrom=4294967295
TimeTo=4294967295
UseActDate=0
UseFromTo=0
UseLastMonths=1
UseLastDays=0
LastDaysMonths=1
ShowRegister=0
SortAscending=0
```

Gespeicherte Anfragen

Anfragen, die Benutzer mit einfachen Recherchemasken und Recherchemasken im Expertenmodus zusammenstellen, können gespeichert werden. Für gespeicherte Anfragen können Benutzer Felder mit Variablen belegen. Werden gespeicherte Anfragen ausgeführt, werden nur die mit Variablen belegten Felder angezeigt.

Einträge für eine Volltextrecherche werden nicht gespeichert.

Gespeicherte Anfragen werden vom enaio® client in der Tabelle `osconf` unter dem Konfigurationstyp 1 gespeichert und in der Objektsuche angezeigt. Sie können über die Profilverwaltung Benutzern zugeordnet werden.

Beispiel für eine gespeicherte Anfrage:

Gespeichert wird eine Ordneranfrage mit einer statischen Variable. Statische Variablen sind Variablen, bei denen der letzte Eintrag temporär zwischengespeichert wird. Beim nächsten Aufruf wird er voreingetragen, aber beim Beenden des Clients nicht gespeichert.

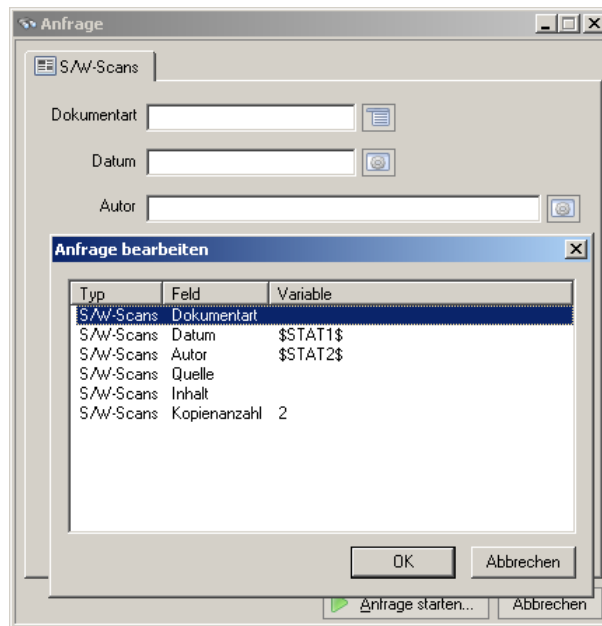


Abbildung 23

Die Anfrage wird mit der Bezeichnung Anfrage 'SW Scan' gespeichert und hat die Eigenschaft 'Standardaktion: öffnen'.

Der Eintrag in der Tabelle `osconf` sieht so aus:

```
[1@0]
#OSPOS000#=$STAT1$
#OSACT#=1
#OSPOS005#=1*
[SYSTEM]
NAME=S/W-Scans Kopienanzahl 2
IDENT=1509
VARREQUEST=1
DEFACTION=0
```

Die Sektion `[Objekttyp@0]` kennzeichnet die gespeicherte Anfrage für den Objekttyp. Die Einträge werden den Feldern über die Tabulatoren-Position (`#OSPOS000#`) zugeordnet.

Die Einträge werden über die Tabulatoren-Position der Felder gespeichert. Wird die Tab-Reihenfolge geändert, werden gespeicherte Anfragen falsch ausgeführt.

SQL-Anfragen

SQL-Anfragen sind Anfragen, mit denen Benutzer unabhängig von den Zugriffsrechten auf DMS-Objekte und unabhängig von der hierarchischen Struktur auf beliebige Daten aus der Datenbank zugreifen und VB-Skripte ausführen können.

Über SQL-Anfragen können weder mit SQL-Befehlen noch mit VB-Skripten Daten aus der Datenbank gelöscht oder geändert werden.

SQL-Anfragen können nur von Benutzern eingerichtet werden, denen über das Sicherheitssystem die Systemrolle 'Editor: starten' und 'Client: SQL-Anfragen bearbeiten' zugeordnet ist.

SQL-Anfragen werden vom enaio® client in der Tabelle `osconf` unter dem Konfigurationstyp 5 gespeichert und in der Objektsuche angezeigt. Sie können über die Profilverwaltung Benutzern zugeordnet werden oder Benutzern zugesendet werden.

Beispiel für eine Erweiterte Anfrage:

Angefragt werden einzelne Verschlagwortungen, Geschlecht (feld9) und Alter (feld10) eines Ordnerstyps (stamm2) für eine anonymisierte statistische Auswertung. Für die Ergebnisliste sind die Spalten bezeichnet (HEADER).

Der Benutzer kann über eine Schaltfläche (Export) ein VB-Skript starten (SCRIPT1), über das eine Microsoft Excel-Tabelle mit den Daten erzeugt wird. Er benötigt keine Zugriffsrechte auf den Ordnerstyp.

```
[P-Statistik]
IDENT=1250
SQL=select feld9,feld10 from stamm2
HEADER=Geschlecht,80;Alter,40
TREEPARENT=0
FLAG1=1
TEXT1=Export
FLAG2=0
TEXT2=
FLAG3=0
FLAG4=0
FLAG5=0
CHANGED=0
SCRIPT1=sub ExportToExcel()....
SCRIPT2=
SCRIPT3=
```

Anzeige von XML-Dokumenten

XML-Dokumenten werden Stylesheets zugeordnet, mit denen sie in einem Browser angezeigt werden. Die Zuordnung wird über die Konfigurationsdatei `as.cfg` verwaltet. Dort wird eine Sektion für XML-Dokumente eingerichtet.

Beispiel:

```
[XML]
objekt(458752)=Tabelle,Grafik
Tabelle=tabelle.xsl;html
Grafik=grafik.xsl;svg
default(458752)=Grafik
```

Beim Öffnen wird das Default-Stylesheet verwendet. Andere können über eine Liste ausgewählt werden. Die Stylesheets müssen im Verzeichnis `\etc\Templates` gespeichert werden. Benutzer können aber auch über einen Dateiauswahldialog beliebige Stylesheets wählen. Sind keine Stylesheets eingerichtet, werden die XML-Dateien angezeigt.

Ordner in der Objektsuche

Um Verweise auf DMS-Objekte, Verknüpfungen zu externen Programmen, gespeicherte Anfragen, SQL-Anfragen und Recherchemasken in der Objektsuche

übersichtlich zu verwalten, können Benutzer Ordner unterhalb des Standard-Ordners 'Desktop' einrichten.

Die Daten werden in der Tabelle `osconf` mit dem Konfigurationstyp 4 gespeichert.

Beispiel:

Ordner in der Objektsuche
des Clients:



Eintrag in der Datenbank:

```

[ 26279936 ]
#OSPOS000#=26279937
#OSPOS001#=26279939
[ 26279937 ]
NAME=Verzeichnis 1
#OSPOS000#=26279938
[ 26279938 ]
NAME=Verzeichnis 1.1
[ 26279939 ]
NAME=Verzeichnis 2
  
```

Der Ordner 'Desktop' wird automatisch angelegt.

Daten, die in der Objektsuche verwaltet werden, werden mit der Angabe der Ordernummer gespeichert.

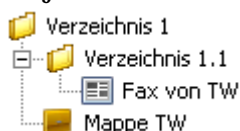
Verweise auf DMS-Objekte

Benutzer können Verweise auf DMS-Objekte in der Objektsuche in den Ordner 'Desktop' oder in einen selbst eingerichteten Ordner legen und zusätzlich auf der Navigation verwalten.

Die Daten werden in der Tabelle `osconf` mit dem Konfigurationstyp 7 gespeichert. Angegeben sind jeweils die eindeutige ID, der Objekttyp und die Zuordnung zu einem Ordner (`TREEPARENT`).

Beispiel:

Objektsuche:



Eintrag in der Datenbank:

```

[Fax von TW]
#OSIDENT#=1436
#OSTYPE#=131099
TREEPARENT=26279938
POS=0
SYSMODE=0
[Mappe TW]
#OSIDENT#=1524
#OSTYPE#=13303808
TREEPARENT=26279937
POS=1
SYSMODE=0
  
```

Beim Löschen von DMS-Objekten wird nicht überprüft, ob Verweise auf diese Objekte angelegt wurden.

Verknüpfungen zu externen Programmen

Benutzer können Verknüpfungen zu externen Programmen in der Objektsuche in den Ordner 'Desktop' oder in einen selbst eingerichteten Ordner und zusätzlich auf

die Navigation legen. Diese Verknüpfungen können auch im 'Senden an' - Menü angelegt werden. Ein externes Programm kann einfach gestartet werden oder mit der Übergabe von Daten gestartet werden.

Die folgenden Parameter werden für die Übergabe von Daten verwendet:

§ %o

Ein oder mehrere Pfade auf DMS-Dateien werden direkt übergeben. Als Trennzeichen dient das Leerzeichen. Wenn der Pfad oder Dateiname Leerzeichen enthält, so wird der Dateipfad in Anführungszeichen eingeschlossen.

§ %p

Ein Pfad auf eine Textdatei, die wiederum je Zeile einen Pfad auf eine DMS-Datei enthält.

§ %i

Ein Pfad auf eine Textdatei, die wiederum je Zeile die eindeutige ID und mit Komma getrennt den Objekttyp enthält. Dieser Parameter empfiehlt sich, wenn viele Objekte übergeben werden.

§ %f

Ein oder mehrere Pfade auf Dateien aus dem Archiv. Liegen die Dateien noch nicht im CACHE-Bereich des Clients, werden sie dorthin übertragen. Leerzeichen werden wie beim Parameter %o behandelt.

§ %g

Ein Pfad auf eine Textdatei, die wiederum je Zeile einen Pfad auf eine Datei im CACHE-Bereich enthält. Dieser Parameter empfiehlt sich, wenn viele Objekte übergeben werden. Liegen die Dateien noch nicht im CACHE-Bereich des Clients, werden sie dorthin übertragen.

Weiterhin können Funktionstasten zusammen mit einer Tastenkombination Strg-Umschalt festgelegt werden, um direkt über diese Tastenkombination das externe Programm aufzurufen. Ist in dem aufzurufenden Programm ein Icon definiert kann dieses in der Baumdarstellung für das externe Programm verwendet werden. Es ist allerdings auch möglich dem Programm ein anderes Icon zuzuweisen.

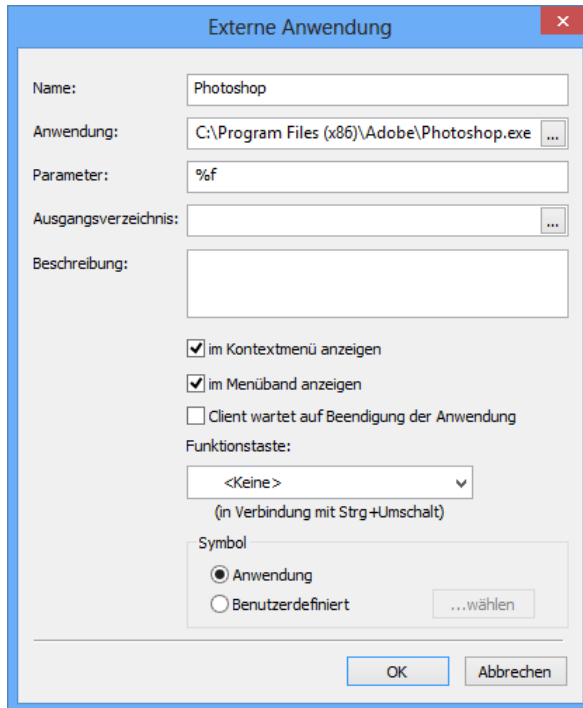
Die übergebenen Daten werden vom enaio® client nicht gelöscht. Dateien, die in den CACHE-Bereich des Clients übertragen werden, sind nicht als ausgecheckt gekennzeichnet. Sie werden beim Beenden des Clients automatisch aus dem CACHE-Bereich gelöscht.

Die Konfigurationsdaten werden in der Tabelle `osconf` mit dem Konfigurationstyp `2` gespeichert.

Beispiel:

Konfiguration:

Eintrag in der Datenbank:



[Photoshop]
EXECUTE=C:\Programme\Adobe\Photoshp.exe
PARAM=%f
SEND2MENU=1
ADDTOLBAR=1
TREEPARENT=26279936
POS=6

Mappen

Benutzer können DMS-Objekte in Mappen zusammenstellen. Öffentliche Mappen sind für alle Benutzer zugänglich, nicht öffentliche für den Ersteller, für den vom Ersteller angegebenen Adressaten und für Benutzer, denen die Mappe gesendet wird.

Öffnet ein Benutzer eine Mappe, werden die Zugriffsrechte des Benutzers nur teilweise beachtet:

- § Klauseln auf Zugriffsrechte für die Anzeige in der Mappe nicht beachtet.
- § DMS-Objekte, auf die der Benutzer nur keine Rechte hat, weil er die Zugriffsrechte auf den Ordner, in dem die Objekte liegen, nicht hat, werden angezeigt und sind bearbeitbar.

Über Mappen können also Benutzern DMS-Objekte zugänglich gemacht werden, die sie nicht selbst recherchieren können.

Mappen werden, zusammen mit den Ablagen über die Tabelle `mappe` verwaltet:

| Spalte | Inhalt |
|---------------|---------------------------------------------------------|
| id | die eindeutige ID der Mappe/Ablage |
| loeschen | zur Zeit nicht verwendet |
| zeitstempel | der genaue Zeitpunkt, zu dem Mappen angelegt wurden |
| angelegt | das Anlagedatum von Mappen |
| anleger | der Anleger von Mappen |
| empfaenger | der Empfänger von Mappen oder der Besitzer der |
| thema | Ablage |
| typ | das Thema von Mappen |
| links | öffentliche Mappe (0) oder nicht öffentliche Mappe (1) |
| anleger_id | Kennzeichnung, ob Notizen existieren (1) oder nicht (0) |
| empfaenger_id | zur Zeit nicht verwendet |

| | |
|--|--------------------------|
| | zur Zeit nicht verwendet |
|--|--------------------------|

Tabelle 24

Legt ein Benutzer ein DMS-Objekt in eine Mappe, trägt der enaio® client die Daten in die Tabelle `mdrel` ein. Diese Tabelle enthält ebenfalls alle Dokumente aus Ablagen und hat folgende Struktur:

| Spalte | Inhalt |
|-------------|------------------------------------------------------------|
| loeschen | zur Zeit nicht verwendet |
| zeitstempel | der genaue Zeitpunkt, zu dem das DMS-Objekt angelegt wurde |
| mappe_id | die eindeutige ID der Mappe/Ablage |
| object_id | die eindeutige ID des DMS-Objekts |
| stamm_id | Ordner-ID |
| objekttyp | die ID des Objekttyps |
| modul | Haupttyp von Dokumenten ohne Typ (Ablage) |
| eingang | zur Zeit nicht verwendet |
| absender | der Anleger von Dokumenten ohne Typ (Ablage) |
| ausgang | zur Zeit nicht verwendet |
| anzahl | Anzahl der Seiten von Dokumenten ohne Typ (Ablage) |

Tabelle 25

Öffentliche Mappen können von allen Benutzern bearbeitet werden, nicht öffentliche nur vom Ersteller. Das Recht zum Bearbeiten schließt ein, eine Mappe zu löschen, DMS-Objekte aus der Mappe zu entfernen und die Mappe zu versenden. Benutzer, denen eine nicht öffentliche Mappe gesendet wird, können DMS-Objekte aus der Mappe entfernen, die Mappe selbst aber nicht löschen. Das Datenblatt einer Mappe kann nur vom Ersteller editiert werden.

Löscht ein Benutzer Daten aus einer Mappe, werden die Einträge aus der Tabelle `mdrel` gelöscht. Löscht ein Benutzer Mappen, werden die Daten aus der Tabelle `mappe` gelöscht und ebenfalls alle Einträge in der Tabelle `mdrel`, die auf die Mappe bezogen sind.

Wiedervorlagen

Benutzer können Dokumenten, Mappen, Registern und Ordnern einen Zeitpunkt für eine Wiedervorlage zuordnen. Wiedervorlagen werden für Benutzer oder Benutzergruppen eingerichtet. Adressaten finden Wiedervorlagen im Abonnement-Bereich auf der Registerkarte 'Wiedervorlage', wenn Sie die entsprechenden Zugriffsrechte besitzen.

Wiedervorlagendaten werden vom Client in der Tabelle `osrevisit` gespeichert. enaio® server informiert alle Benutzer, denen das Objekt wiedervorgelegt wird.

Die Tabelle `osrevisit` hat folgende Struktur:

| Spalte | Inhalt |
|------------|-----------------------------------------------------------------|
| id | die eindeutige ID des DMS-Objekts |
| objekttyp | der Objekttyp |
| user_id | die ID des Adressaten |
| firstvisit | Zeitpunkt der Wiedervorlage |
| lastvisit | Zeitpunkt, an dem das DMS-Objekt den Status 'bearbeitet' erhält |
| infotext | Anmerkung des Anlegers |

| | |
|-------------|------------------------------------------------------------------------------------------|
| set_user_id | die ID des Anlegers |
| set_time | Anlagedatum |
| osinformed | Zeitpunkt, zu dem die Wiedervorlage einem Benutzer auf der Registerkarte angezeigt wurde |
| ostype | Kennzeichnung für Abonnements |
| osabotype | Actionflag für Abonnements |

Tabelle 26

Benutzer, denen die Systemrolle 'Konfiguration Sicherheitssystem' zugeordnet ist, können im Abonnement-Bereich auf der Registerkarte 'Wiedervorlage' die Wiedervorlagen aller Benutzer einsehen. Kennzeichnet ein Benutzer eine Wiedervorlage als 'bearbeitet', trägt der Client in der Spalte `lastvisit` das aktuelle Datum ein. Nur wenn dort ein Eintrag vorhanden ist, kann eine Wiedervorlage gelöscht werden.

Abonnements

Benutzer können Dokumente, Register und Ordner abonnieren. Abonnierte Objekte werden im Abonnementbereich aufgelistet, sobald Daten geändert wurden. Dabei kann gewählt werden, ob bei Änderungen an der Verschlagwortung oder für Dokumente bei Änderungen an einer Datei das DMS-Objekt im Abonnementbereich aufgeführt werden soll. Abonnementdaten werden vom Client in der Tabelle `osabonnement` gespeichert.

Die Tabelle `osabonnement` hat folgende Struktur:

| Spalte | Inhalt |
|------------|-------------------------------------------------------------|
| id | eine eindeutige ID für das Abonnement |
| object_id | die eindeutige ID des DMS-Objekts |
| actionflag | Änderungen der Verschlagwortung (3), Änderung der Datei (4) |
| produkt | zur Zeit nicht verwendet |
| benutzer | Benutzername |
| station | zur Zeit nicht verwendet |
| channel | Kennzeichnung, ob eine E-Mail gesendet werden soll |
| mail | angegebene E-Mail Adresse |
| alias | Infotext |
| osobjtype | die ID des Objekttyps |

Tabelle 27

Werden abonnierte DMS-Objekte geändert, schreibt enaio® server Daten des geänderten DMS-Objekts in die Tabelle `osrevisit`. In die Spalte `ostype` trägt er 1 ein, in die Spalte `osabotype` das Actionflag 3 oder 4. Er informiert alle Benutzer, die das Objekt abonniert haben.

Öffnet ein Benutzer die Registerkarte 'Abonnement', listet der Client alle DMS-Objekte aus der Tabelle `osrevisit` auf, die dem Benutzer zugeordnet sind und die in der Spalte `ostype` mit 1 gekennzeichnet sind. Unbearbeitete Abonnements werden dabei durch Fettdruck hervorgehoben.

Löscht ein Benutzer ein Objekt auf der Registerkarte 'Abonnement' wird der Eintrag in der Tabelle `osrevisit` gelöscht, das Abonnement aber nicht aufgehoben. Löscht der Benutzer ein Abonnement auf der Registerkarte 'Abonnierte Objekte', wird das Objekt aus der Tabelle `osabonnement` gelöscht, aber nicht überprüft, ob dieses Objekt in die Tabelle `osrevisit` steht.

Notizen und Verknüpfungen

Benutzer können zu DMS-Objekten, Dokumenten, Registern, Ordnern und Mappen Textnotizen und Verknüpfungen mit anderen DMS-Objekten verwalten. Textnotizen werden mit einem Notizeditor erstellt. Verknüpfungen werden erzeugt, indem ein DMS-Objekt mit der Maus in das Notizfenster gezogen wird. Für dieses DMS-Objekt wird automatisch ebenfalls eine Verknüpfung auf das zugeordnete DMS-Objekt im Notizfenster erzeugt.

Fügt ein Benutzer Textnotizen oder Verknüpfungen in das Notizfenster eines DMS-Objekts ein, schreibt der Client die Anzahl in die Spalte `links` der Verschlagwortungstabelle des DMS-Objekts.

Verknüpfungen werden in die Tabelle `hyplnk` eingetragen.

Die Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|-----------------------|-------------------------------------------------|
| <code>obj_id1</code> | eine eindeutige ID des einen DMS-Objekts |
| <code>obj_typ1</code> | der Objekttyp des DMS-Objekts |
| <code>obj_id2</code> | eine eindeutige ID des zugeordneten DMS-Objekts |
| <code>obj_typ2</code> | der Objekttyp des DMS-Objekts |
| <code>medium</code> | zur Zeit nicht verwendet |
| <code>name</code> | zur Zeit nicht verwendet |

Tabelle 28

Textnotizen werden in die Tabelle `remarks` eingetragen

Die Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|-------------------------|------------------------------------------------------------|
| <code>id</code> | eine eindeutige ID für die Notiz |
| <code>parent</code> | die eindeutige ID des zugeordneten DMS-Objekts |
| <code>parenttype</code> | der Objekttyp des DMS-Objekts |
| <code>medium</code> | die ID des Mediums, auf der die Notizdatei gespeichert ist |
| <code>type</code> | die Farbkennzeichnung der Notiz |
| <code>stamp1</code> | Zeitpunkt der Erstellung |
| <code>user1</code> | Ersteller |
| <code>stamp2</code> | Zeitpunkt der letzten Bearbeitung |
| <code>user2</code> | Bearbeiter |

Tabelle 29

Exportieren von Daten

Benutzer können die Verschlagwortung von DMS-Objekten und die Dateien von Dokumenten exportieren. Die Verschlagwortung von DMS-Objekten gleichen Typs kann in die Zwischenablage kopiert und aus der Zwischenablage in einer anderen Anwendung eingefügt werden. Benutzer können aber auch Konfigurationen erstellen und die Verschlagwortung in eine Textdatei oder eine XML-Datei exportieren.

Dateien von Dokumenten werden im gespeicherten Format exportiert. Bei Bilddokumenten können einzelne Seiten oder alle Seiten exportiert und dabei das Exportformat angegeben werden.

Verschlagwortung exportieren

Für den Export der Verschlagwortung wählt der Benutzer, ob er für die markierten Objekte gleichen Typs die in der Trefferliste oder dem Ordnerfenster angezeigte Verschlagwortung exportieren will oder eine gespeicherte Konfiguration verwendet. In einer Konfiguration werden alle Felder des Objekttyps angegeben, deren Verschlagwortung exportiert werden soll. Für Register können zusätzlich die Ordnerfelder angegeben werden, für Dokumente zusätzlich die Register- und Ordnerfelder.

Die Verschlagwortung kann in die Zwischenablage oder in eine Datei exportiert werden.

Die Konfigurationsdaten werden in der Tabelle `osconf` mit dem Konfigurationstyp 3 gespeichert.

Beispiel:

Konfiguration:

Eintrag in der Datenbank:

```
[ ZWAL#Emailexport ]
FLAGS=3
FILE=C:\daten\emailexport.txt
ORDTYPE=1
DOCTYPE=393218
FIELD00=o.feld11
FIELD01=o.feld4
FIELD02=o.feld3
FIELD03=d.datum1
FIELD04=d.feld1
FIELD05=d.feld2
FIELD06=d.feld4
```

Datensätze werden durch ein Zeilenumbruchzeichen getrennt, die einzelnen Daten in der Zeile durch ein Tab-Zeichen.

Exportierte XML-Dateien haben folgende Struktur:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <!-- optimal systems GmbH -->
- <Exportliste>
- <Zeile>
  <HFeld>SpaltenüberschriftFeld1</HFeld>
  <HFeld>SpaltenüberschriftFeld2</HFeld>
</Zeile>
- <Zeile>
  <DFeld>VerschlagwortungFeld1</DFeld>
  <DFeld>VerschlagwortungFeld2</DFeld>
</Zeile>
- <DFeld>VerschlagwortungFeld1</DFeld>
  <DFeld>VerschlagwortungFeld2</DFeld>
</Zeile>
</Exportliste>
```

Geben Benutzer eine XSL-Datei an, wird die XML-Datei über die XSL-Datei in HTML formatiert.

Dateien exportieren

Werden aus Trefferlisten oder Ordnerfenstern die Dateien von Dokumenten exportiert, werden die Dateien im modulspezifischen Dateiformat gespeichert, Windows-Dokumentdateien im Anwendungsformat.

Schwarz/Weiß-Bilder werden im modulspezifischen Dateiformat TIFF G4 gespeichert, mehrseitige Dokumente als Multipage TIFFs.

Die Bezeichnung wird automatisch nach folgendem Schema erzeugt:

| |
|-----------------------------------------|
| DOC[Nummer][Seitennummer].[Dateiendung] |
|-----------------------------------------|

[Nummer] ist eine dreistellige Nummerierung der exportierten Dateien.

[Seitennummer] ist die hexadezimale dreistellige Seitenzahl. Bei Schwarz/Weiß-Bildern und Film-Dokumenten steht hier immer '000'. Bei Windows-Dokumenten, die immer aus einer Datei bestehen, fällt dieser Teil weg.

Beim Export wird nicht überprüft, ob im angegebenen Verzeichnis schon Dokumente mit gleicher Bezeichnung liegen. Ist das der Fall, werden diese überschrieben. Die exportierten Dateien enthalten Anmerkungen auf Folien, die der Benutzer nicht bearbeiten darf. Anmerkungen, die der Benutzer bearbeiten darf, werden nicht mit exportiert.

Bei Schwarz/Weiß-, Graustufen- und Farbbildern können aus dem Anzeigemodul zusätzlich einzelne Seiten exportiert werden und kann das Format gewählt werden. Über die Funktion 'Seite speichern unter' aus dem Menü 'Datei' wird die angezeigte Seite unter der angegebenen Bezeichnung und im gewählten Format gespeichert.

Über die Funktion 'Seiten exportieren' öffnen Benutzer den Exportdialog und geben detailliert an, welche Seiten exportiert werden sollen, nach welchem Namensschema die Seiten bezeichnet werden, welches Format verwendet wird und ob Anmerkungen, die der Benutzer bearbeiten kann, zusätzlich zu den Anmerkungen, die er nicht bearbeiten kann, eingebrannt werden.

Das Exportieren von Dateien wird in der Tabelle `osobjhist` protokolliert und in der Bearbeitungshistorie mit der Information 'Objekt schreibgeschützt ausgegeben' angezeigt.

Versenden von DMS-Objekten

Für das Versenden von DMS-Objekten aus dem Menü oder Kontextmenü 'Senden an...' innerhalb von Trefferlisten öffnet der enaio® client den installierten (MAPI-) E-Mail Client und übergibt Dateien als Anhang für eine E-Mail. Ist Microsoft Outlook installiert, können die Benutzer auch direkt Microsoft Outlook für das Versenden auswählen. Dann steht ebenfalls das Outlook-AddIn mit der Funktion 'Senden und Archivieren' zur Verfügung.

Benutzer können DMS-Objekte intern versenden und Dokumente zusätzlich extern versenden.

Für das interne Versenden erzeugt der enaio® client Textdateien mit der Endung `os`. Die Endung `os` wird durch den enaio® client registriert.

Gespeicherte Anfragen, SQL-Anfragen und Verknüpfungen zu externen Programmen werden intern als OS-Datei versendet. Ein Empfänger, der den enaio® client geöffnet hat, kann diese in die Objektsuche übernehmen und ausführen.

Diese Daten werden so, wie sie auch in der Tabelle `osconf` gespeichert sind, als OS-Datei versendet.

Verweise auf Dokumente, Register, Ordner und Mappen können ebenfalls intern als OS-Datei versendet werden. Ein Empfänger, der auf den enaio® client zugreifen kann, öffnet über die OS-Datei das DMS-Objekt. Die Zugriffsrechte werden beachtet.

Beispiel für eine OS-Datei mit einem Verweis auf ein Dokument:

```
[OSAS]
DOKUMENTTYP=Fax
SCHRANK=Adressen
[Fax]
#OSID#=1436
[Adressen]
#OSID#=250
```

Werden Dokumente extern versendet, erzeugt der enaio® client die Dateien und fügt sie in den Anhang einer E-Mail ein. Das Bezeichnungsschema der Dateien und das Dateiformat entsprechen denen beim Export von Dokumenten aus Trefferlisten und Ordnerfenstern.

Das Versenden von Dokumenten, Registern und Ordnern wird in der Tabelle `osobjhist` protokolliert und in der Bearbeitungshistorie angezeigt.

Ausdruck von Daten

Für den Ausdruck von Datenblättern, Trefferlisten und E-Mails erzeugt der enaio® client XML-Dateien, formatiert sie über Stylesheets zu HTML-Dateien und druckt diese dann aus.

Die Dateien werden im benutzerspezifischen Arbeitsbereich `temp\ostemp` gespeichert und beim Beenden des Clients gelöscht. Die Stylesheets befinden sich im Verzeichnis `clients\client32`.

Trefferlisten werden über das Stylesheet `print1.xslt` formatiert, Datenblätter über `print2.xslt` und E-Mails über `print3.xslt`.

Beim Ausdruck von Datenblättern werden Daten des Dialogelements 'Tabelle' und Daten der Mehrfachfelder nicht mit gedruckt. Haben Objekte mehrere Standorte, wird nur der Standort mit gedruckt, der auf der Registerkarte angezeigt ist. Dokumente werden vor dem Ausdruck in den CACHE-Bereich übertragen.

Windows-Dokumente werden aus der geöffneten Anwendung gedruckt. XML-Dokumente werden, wie beim Anzeigen, über die zugeordneten Stylesheets als HTML-Dateien gespeichert und ausgedruckt. Filmdokumente können nicht gedruckt werden. Bilddokumente können aus Trefferlisten und Ordnerfenstern gedruckt werden. Anmerkungen auf Folien, die ein Benutzer nicht bearbeiten darf, werden vor dem Übertragen in den CACHE-Bereich eingebrannt.

Für den Druck von Schwarz/Weiß-Bildern und den Druck von Farb- oder Graustufenbildern werden benutzerspezifisch unterschiedliche Einstellungen verwaltet.

Die Daten werden mit den Daten des Konfigurationstyps 3 in der Tabelle `osconf` gespeichert.

Für den Schwarz/Weiß-Druck werden die Daten in der Sektion [ASD] gespeichert, für den Farb- und Graustufendruck in der Sektion [ASP]. Beim Druck werden Anmerkungen, die Benutzer bearbeiten dürfen, nicht mit gedruckt.

Ist ein Bilddokument geöffnet, steht eine weitere Druckfunktion zur Verfügung. Diese Druckfunktion ermöglicht, Anmerkungen, die Benutzer bearbeiten dürfen,

mit auszudrucken. Der Druckdialog wird über den Tastatur-Kurzbefehl 'Alt-P' aufgerufen. Der Ausdruck nimmt allerdings mehr Zeit in Anspruch.

Der Ausdruck von Dokumenten wird in der Tabelle `osobjhist` protokolliert und in der Bearbeitungshistorie angezeigt.

XML-Daten

Die XML-Daten werden über die Stylesheets aus dem Verzeichnis `clients\client32` formatiert. Die Stylesheets können mit einem beliebigen Editor bearbeitet werden.

XML-Daten für Trefferlisten

Das Stylesheet `print1.xslt` formatiert XML-Trefferlisten.

Beispiel für eine XML-Trefferliste:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- optimal systems GmbH -->
<Listen>
<Trefferliste>
<Anfrageliste>
<Zeile>
<DFeld>Adressen</DFeld>
<DFeld>Kurzname = dros</DFeld>
</Zeile>
</Anfrageliste>
<Datenliste>
<Zeile>
<HFeld>Kurzname</HFeld>
<HFeld>Name</HFeld>
<HFeld>Vorname</HFeld>
</Zeile>
<Zeile>
<DFeld>Drossel</DFeld>
<DFeld>Drossel-Beyer</DFeld>
<DFeld>Paul</DFeld>
</Zeile>
<Zeile>
<DFeld>Drosse</DFeld>
<DFeld>Drosse-Meierstein</DFeld>
<DFeld>Paula</DFeld>
</Zeile>
</Datenliste>
<Allgemein>
<Titel>Dokumentliste</Titel>
<Benutzer>THOMAS</Benutzer>
<Dokumenttyp>Adressen</Dokumenttyp>
<Anzahl>2</Anzahl>
</Allgemein>
</Trefferliste>
```

Die XML-Daten für Trefferlisten enthalten den Bereich `<Anfrageliste>` mit den Anfragekriterien für die Recherche, den Bereich `<Datenliste>` mit den Daten der Trefferliste und den Bereich `<Allgemein>` mit den Angabe 'Anzahl der Treffer', 'Benutzer', 'Titel' und 'Objektyp'.

Die Datenliste enthält die Verschlagwortungsdaten, für die Spalten in der Trefferliste angezeigt werden.

XML-Daten für Datenblätter

Das Stylesheet `print2.xslt` formatiert XML-Datenblätter.

Beispiel für ein XML-Datenblatt:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- optimal systems GmbH -->
<Datenblatt>
  <Allgemein>
    <Titel>Datenblatt</Titel>
    <Benutzer>THOMAS</Benutzer>
  </Allgemein>
  <Karten>
    <Karte>
      <Dokumenttyp>Adressen</Dokumenttyp>
      <Zeile>
        <DFeld>Name</DFeld>
        <DFeld>Drossel </DFeld>
      </Zeile>
      <Zeile>
        <DFeld>Vorname</DFeld>
        <DFeld>Paul</DFeld>
      </Zeile>
      <Zeile>
        <DFeld>Strasse</DFeld>
        <DFeld>Hindburgdamm 205</DFeld>
      </Zeile>
      <Zeile>
        <DFeld>Titel</DFeld>
        <DFeld>Dr. <DFeld/>
      </Zeile>
      <Zeile>
        <DFeld>Telefon</DFeld>
        <DFeld>030 1234567</DFeld>
      </Zeile>
      <Zeile>
        <DFeld>Wohnort</DFeld>
        <DFeld>Berlin</DFeld>
      </Zeile>
      <Zeile>
        <DFeld>PLZ / Ort</DFeld>
        <DFeld>10200</DFeld>
      </Zeile>
      <Zeile>
        <DFeld>Fax</DFeld>
        <DFeld />
      </Zeile>
      <Zeile>
        <DFeld>E-mail</DFeld>
        <DFeld />
      </Zeile>
      <Zeile>
        <DFeld>Anrede</DFeld>
        <DFeld>Herrn</DFeld>
      </Zeile>
      <Zeile>
        <DFeld>WWW</DFeld>
        <DFeld />
      </Zeile>
    </Karte>
  </Karten>
</Datenblatt>
```

Das folgende Ordner-Datenblatt wurde ausgedruckt:

Beim Ausdruck von Datenblättern wird für jede angezeigte Registerkarte ein Bereich <Karte> ausgedruckt. Der Ausdruck eines Registerdatenblatts enthält also immer auch einen Ordner-Bereich, der Ausdruck eines Dokumentendatenblatts enthält immer einen Ordner und einen Basisparameterbereich.

XML-Daten für E-Mails

Das Stylesheet `print3.xslt` formatiert XML-E-Mails.

Beispiel für eine XML-E-Mail:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- optimal systems GmbH -->
<E-Mail>
<Common>
<User>THOMAS</User>
</Common>
<Data>
<From>Jürgen Nitz</From>
<Date>23.10.2002 16:17</Date>
<To>Entwicklung; Consulting; Support</To>
<Cc />
<Subject>Release Warning</Subject>
<Body><ole0.bmp>> Betrifft Versionen: Update von 3.60 SPVII auf OS:4x SPIII
Symptome: Ab 3.60 SPVII wird die Datei axindex.cfg aufgesplittet. OS:4x Versionen
können bisher nicht damit umgehen. Lösung: axindex.exe 4.00.569 als Hotfix im SP-
Verzeichnis</Body>
</Data>
</E-Mail>
```

Druckkennzeichnung für DMS-Dokumente

Über den enaio® administrator kann eine Druckkennzeichnung für Dokumente eingerichtet werden. Beim Ausdruck von Bilddokumenten werden die dort konfigurierten Fuß- und Kopfzeilen gedruckt.

Die Konfiguration wird in der Datei `as.cfg` gespeichert.

Beispiel:

```
[PRINTHEADER]
FONT=Arial
FONTSIZE=12
POSITION=2
```

| |
|------------------------------------------------------|
| TEXT=Archiv-Ausdruck TEXTCOLOR=3 TRANSPARENT=0 |
|------------------------------------------------------|

Die Konfiguration gilt für alle Bilddokumente und alle Benutzer.

Formulardruck

Mit enaio® report-generator können Verschlagwortungsdaten aus unterschiedlichen DMS-Objekten zusammengestellt und formatiert ausgedruckt werden. Für den Formulardruck wird das Programm `axvbform.exe` als externes Programm eingebunden. Das Programm benötigt eine Konfigurationsdatei `axvbform.cfg`, die Zuordnungen von DMS-Objekttypen zu Formularen und Formularsets enthält.

Wird das Programm mit der Übergabe von DMS-Objekten gestartet, ermittelt es über die Konfigurationsdatei die benötigten Daten und fragt diese über die COM-Schnittstelle beim enaio® client an. Über die COM-Schnittstelle kann auf Daten zugegriffen werden, auf die der Benutzer keine Zugriffsrechte hat. Der Client beachtet die Zugriffsrechte nicht, wenn DMS-Objekttypen im enaio® editor die Eigenschaft 'Sicherheitssystem nicht beachten' erhalten. Die Daten werden entweder über ein temporär erzeugtes Microsoft Word-Dokument oder über eine List&Label-Formatierungsdatei ausgedruckt. Statt Daten so zu drucken, können die Formulare mit den Daten ebenfalls über enaio® printer als Dokument in das DMS übernommen werden.

Bearbeiten von Dokumenten

Allgemeine Voraussetzungen

Für das Bearbeiten von Dokumenten benötigt der enaio® client die aktuellen Objektdefinitionen, Objektrelationen und Konfigurations-Einträge, die er beim Start liest. Er benötigt ebenfalls die aktuellen Rechte des angemeldeten Benutzers. Die Rechte können im enaio® client über Tastaturkombination 'Umschalttaste-Strg-F5' aktualisiert werden.

Änderungen an der Verschlagwortung von DMS-Objekten können über das Rechtesystem und über Eigenschaften von Dialogelementen sehr detailliert zugelassen oder unterbunden werden. Änderungen an Dokumentendateien sind generell nach dem Archivieren nicht mehr möglich. Von archivierten Dokumenten können aber bearbeitbare Kopien, bei Windows-Dokumenten auch Varianten erstellt werden. Änderungen an DMS-Objekten können weitreichend in einer Bearbeitungshistorie gespeichert werden, über die Änderungen eingesehen und rückgängig gemacht werden können. Das Löschen von DMS-Objekten kann nicht rückgängig gemacht werden.

Änderung der Verschlagwortung

Die Verschlagwortung eines Ordners, Registers oder Dokuments wird über das Datenblatt geändert. Datenblätter von Registern enthalten zusätzlich eine Registerkarte der Ordnersverschlagwortung, Datenblätter von Dokumenten enthalten zusätzlich eine Registerkarte der Ordnersverschlagwortung, eine Registerkarte der Basisparameter des Dokuments und gegebenenfalls eine Registerkarte der Registerverschlagwortung.

Abbildung 24

Hat ein Benutzer beispielsweise die Zugriffsrechte auf ein Register, aber nicht auf den Ordner, in dem das Register liegt, enthält das Datenblatt des Registers keine Registerkarte der Ordnerverschlagwortung.

Basisparameter werden automatisch geführt und sind nicht editierbar. Alle anderen Felder auf Registerkarten eines Datenblatts sind editierbar, falls das Rechtesystem oder Eigenschaften von Dialogelementen dies zulassen. Nicht bearbeitbare Dialogelemente sind grau gekennzeichnet. Öffnet ein Benutzer das Datenblatt eines Dokuments, kann er über die Ordnerregisterkarte ebenfalls die Verschlagwortung des Ordners und Registers ändern.

Datenblätter sind editierbar, unabhängig davon, ob andere Benutzer ebenfalls diese Datenblätter geöffnet haben und bearbeiten. Änderungen an der Verschlagwortung werden vom Client in der Tabelle `osobjhist` protokolliert.

Die Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|------------------------|--------------------------------------------------------------------------------------------------|
| <code>osguid</code> | die eindeutige ID jeder Aktion |
| <code>osid</code> | die eindeutige ID des Objekts, auf das die Aktion bezogen ist |
| <code>ostype</code> | die ID des Objekttyps |
| <code>osuser</code> | die eindeutige ID des Benutzers, der die Aktion ausgeführt hat |
| <code>ostime</code> | der Zeitstempel |
| <code>osaction</code> | eine Aktions-ID |
| <code>osstation</code> | die GUID der Arbeitsstation, an der die Aktion ausgeführt wurde |
| <code>osinfo</code> | der Info-Text, der in der Bearbeitungshistorie zusätzlich zur Aktionsbeschreibung angezeigt wird |

Tabelle 30

Die Aktions-ID für Änderungen der Verschlagwortung ist 3.

Diese Protokollierung der Aktionen ist unabhängig von der OXRPT-Protokollierung.

Der Benutzer, der die Verschlagwortung ändert, wird ebenfalls in die Spalte `modifyuser` der Verschlagwortungstabelle eingetragen, der Zeitpunkt in die Spalte `modifytime`. Im enaio® administrator kann eine systemweite Sicherheitsabfrage für das Speichern der Verschlagwortung eingeschaltet werden. Die Sicherheitsabfrage wird auch beim Neuanlegen von DMS-Objekten gestellt, allerdings nicht beim Wechsel zwischen Registerkarten. Diese systemweite Einstellung wird in der Datei `as.cfg` gespeichert.

Indexdatenhistorie

Hat ein DMS-Objektyp im enaio® editor die Eigenschaft 'Indexdatenhistorie erzeugen' erhalten, wird die komplette alte Verschlagwortung vor der Änderung in die Schattentabelle übertragen.

Schattentabellen tragen die gleiche Bezeichnung wie die Verschlagwortungstabellen mit einem angehängten 's'.

Die Schattentabellen haben folgende Struktur:

| Spalte | Inhalt |
|---------------------|---------------------------------------------------|
| <code>osguid</code> | die eindeutige ID der Änderung |
| <code>feld#</code> | die Verschlagwortung des Objekts in den einzelnen |

| | |
|--|--------------------------|
| | Feldern vor der Änderung |
|--|--------------------------|

Tabelle 31

Die weiteren Daten der Änderung werden in die Tabelle `osobjhist` eingetragen und sind dort ebenfalls durch die eindeutige ID (Spalte `osguid`) gekennzeichnet. Über die 'Bearbeitungshistorie' können Änderungen eingesehen und Zustände wiederhergestellt werden.

Ordner-, Register-, Dokumentverschlagwortung

Die Verschlagwortung wird über das Datenblatt geändert. Geben Benutzer Daten an, die den Verschlagwortungsvorgaben nicht entsprechen, werden die Dialogelemente gekennzeichnet und der Benutzer aufgefordert, die Daten zu korrigieren. Erst wenn die Verschlagwortung den Vorgaben entspricht, werden die Daten in der Datenbank gespeichert.

Wurden Vorgaben nach dem Anlegen eines DMS-Objekts geändert, kann es notwendig werden, bereits bestehende Verschlagwortungen an die geänderten Vorgaben anzupassen, um andere Änderungen vorzunehmen.

Benutzer können Daten eingeben, die Auswirkungen auf das Rechtesystem haben. Beispiel: Ein Benutzer trägt in ein Feld 'Auftragsvolumen' 300000 ein, hat aber selbst über Klauseln keinerlei Zugriff auf Objekte, die mit Werten ≥ 300000 verschlagwortet sind. Sobald er die Verschlagwortung speichert, aktualisiert der Client die Ansicht. Das Objekt ist für den Benutzer nicht mehr sichtbar.

AddOns mit der Eigenschaft 'automatisch abfahren' werden beim Speichern der Verschlagwortung automatisch ausgeführt. Beim Quickfinder-AddOn und beim Aktenplan-AddOn wird nicht überprüft, ob der Eintrag in verbundenen Verschlagwortungsfeldern geändert wurde.

Ändert der Benutzer auf einer Registerkarte des Datenblatts die Verschlagwortung und wechselt zu einer anderen Registerkarte, wird vor dem Wechsel die Verschlagwortung automatisch in der Datenbank gespeichert oder ein Bestätigungsdialog eingeblendet. Diese benutzerspezifische Einstellung über die Registerkarte 'Bestätigen' wird in der Tabelle `osconf` mit den Daten des Konfigurationstyps 3 gespeichert. Auch wenn eine systemweite Sicherheitsabfrage für das Speichern der Verschlagwortung eingeschaltet ist, wird beim Wechsel der Registerkarten nur ein Bestätigungsdialog angezeigt, wenn der Benutzer dieses eingestellt hat.

Neben der Verschlagwortung wird der Benutzer, der die Verschlagwortung ändert, ebenfalls in die Spalte `modifyuser` der Verschlagwortungstabelle eingetragen, der Zeitpunkt in die Spalte `modifytime`. Die Aktion trägt der enaio® client in der Tabelle `osobjhist` ein. Für die Indexdatenhistorie wird die alte Verschlagwortung in die Schattentabelle eingetragen.

Modifizierungsdaten von Ordnern und Registern werden über die Funktion 'Objektinformation' angezeigt. Modifizierungsdaten von Dokumenten ebenfalls über das Datenblatt auf der Registerkarte Basisparameter.

Ist das DMS-Objekt abonniert, trägt der Client Objektdaten in die Tabelle `osrevisit` ein.

Sammeländerungen

Eine Sammeländerung ist eine gleichzeitige Änderung der Verschlagwortung mehrerer DMS-Objekte gleichen Typs. Bei Sammeländerungen überprüft der enaio® client nicht, ob die Daten, die der Benutzer einträgt den

Verschlagwortungsvorgaben entsprechen. Überprüft wird ebenfalls nicht, ob Verschlagwortungsfelder schreibgeschützt sind. Events werden nicht ausgeführt, AddOns beim Speichern nicht automatisch abgefahren. Überprüft werden nur direkt bei der Eingabe der Datenbanktyp und die erlaubte Länge.

Die Verschlagwortung von Mehrfachfeldern und die Verschlagwortung des Dialogelements 'Tabelle' kann durch eine Sammeländerung nicht geändert werden.

| Sammeländerungen können zu inkonsistenten Daten führen. |

Über den enaio® administrator müssen Sammeländerungen explizit für einzelne Benutzer über die Systemrolle 'Client: Sammeländerung durchführen', erlaubt werden. Wird diese Systemrolle nicht zugeordnet, so sind im System keine Sammeländerungen möglich.

Ändern des Standorts

Der Standort eines Dokuments wird unabhängig von der Verschlagwortung verwaltet. Die Standortdaten sind in der Tabelle `sdrel` gespeichert. Dort steht die ID des zugeordneten Ordners und die ID des Registers der nächst höheren Ebene.

Die Standortdaten von Registern werden in der Tabelle `osparregrel` gespeichert und ebenfalls in der Verschlagwortungstabelle `register#`. In beiden Tabellen stehen die ID des zugeordneten Ordners und die ID des Registers der nächst höheren Ebene. Ordner liegen auf der obersten Ebene der Hierarchie und haben keine Standortdaten.

Verschiebt ein Benutzer ein Register, werden die Standortdaten in der Verschlagwortungstabelle und in der Standorttabelle `osparregrel` aktualisiert. Verschoben werden ebenfalls alle Dokumente, die im Register liegen. Die Standortdaten dieser Dokumente ändern sich, wenn ein Register in einen anderen Ordner verschoben wird. Dann werden die Standortdaten in der Tabelle `sdrel` ebenfalls aktualisiert.

Wird ein Dokument verschoben, werden die Standortdaten in der Tabelle `sdrel` aktualisiert. Bei Dokumenten, die aus der Ablage in einen Ordner verschoben werden, wird der Standort aus der Tabelle `mdrel` gelöscht und der neue Standort in die Tabelle `sdrel` eingetragen.

Dokumenten ohne Typ müssen Benutzer zuerst einen Typ zuordnen und verschlagworten, bevor sie in einen Ordner verschoben werden können. Neben dem Standort wird auch der Speicherort der Dokumentdateien geändert. Standortänderungen werden nicht in der Bearbeitungshistorie protokolliert.

Über den enaio® administrator kann das Verschieben von Dokumenten oder das Verschieben von archivierten Dokumenten unterbunden werden. Die Einstellung wird über die Datei `as.cfg` verwaltet.

| Das Verschieben kann zu inkonsistenten Daten führen, beispielsweise bei paginierten Dokumenten oder bei Verschlagwortungen über das Quickfinder- und das Aktenplan-AddOn. |

Mehrere Standorte durch Verweise

Dokumente können mehrere Standorte erhalten. Dokumente mit mehreren Standorten haben mehrere Einträge in der Tabelle `sdrel`. Der Zugriff auf solche Dokumente ist von allen Standorten aus gleichberechtigt. Die Verwaltung der Verschlagwortung und der Dokumentendateien entspricht der von Dokumenten mit einem Standort. Löscht ein Benutzer ein Dokument mit mehreren Standorten,

wird nur der Verweis, der Eintrag für den aktuellen Standort, aus der Tabelle `sdrel` gelöscht.

Sowohl das Erzeugen eines neuen Standorts wie auch das Löschen eines Standorts wird nicht in der Tabelle `osobjhist` protokolliert und nicht in der Bearbeitungshistorie angezeigt.

Benutzer erzeugen einen weiteren Standort, indem sie ein Dokument mit der linken Maustaste und gedrückter Strg-Taste in einen anderen Ordner oder ein anderes Register ziehen oder indem sie mit der rechten Maustaste ein Dokument in einen anderen Ordner oder ein anderes Register ziehen und über das Kontextmenü die Funktion 'Hierher kopieren' wählen.

Über den enaio® administrator kann das Erzeugen von Verweisen unterbunden werden. Die Einstellung wird über die Datei `as.cfg` verwaltet. Ordner und Register haben immer nur einen Standort. Existieren zu einem Dokument mehrere Verweise, wird auf dem Datenblatt des Dokuments die Anzahl auf den Ordner- und Registerkarten-Reitern angezeigt. Über einen Schalter auf den Registerkarten kann zwischen den Verschlagwortungen gewechselt werden. Das Anlegen von Verweisen wird nicht in der Bearbeitungshistorie protokolliert.

Änderung von Dokumentdateien

Bei Bilddokumenten können Seiten hinzugefügt und gelöscht werden.

Filmdokumenten ist die entsprechende digitale Videosequenz zugeordnet.

Windows-Dokumente werden zum Bearbeiten in der zugeordneten Anwendung geöffnet. In dieser Anwendung kann die Datei beliebig bearbeitet und dann gespeichert werden.

Bilddokumente, Filmdokumente und Windows-Dokumente, die ein Benutzer zum Bearbeiten öffnet, werden ausgecheckt. In der Verschlagwortungstabelle `objectn` sind sie durch den Benutzernamen (`lockuser`) gekennzeichnet und in die Tabelle `doclock` eingetragen. Dadurch können andere Benutzer die Dokumente nur schreibgeschützt öffnen.

Die Tabelle `doclock` hat folgende Struktur:

| Spalte | Inhalt |
|--------------------------|--------------------------------------------------------------|
| <code>object_id</code> | die eindeutige ID des DMS-Objekts |
| <code>object_type</code> | die ID des Objekttyps |
| <code>user_id</code> | die eindeutige ID des Benutzers, der das Dokument bearbeitet |
| <code>user_time</code> | der Zeitstempel |
| <code>info</code> | Statusinformation |
| <code>fileinfo</code> | Änderungsdatum für Windows-Dokumente |
| <code>station</code> | Arbeitsstation des Benutzers |

Tabelle 32

Beim Einchecken eines Dokuments wird der Benutzernamen (`lockuser`) aus der Verschlagwortungstabelle `objectn` gelöscht und der Eintrag aus der Tabelle `doclock` entfernt.

Änderungen an Dateien werden vom Client in der Tabelle `osobjhist` protokolliert und in der Bearbeitungshistorie angezeigt.

Ist die Dokumentenhistorie angeschaltet, so speichert der Server die verschiedenen Dokumentenversionen im WORK-Bereich in einem Verzeichnis, das auf gleicher

Ebene wie die Dokumentendateien angelegt wird und als Bezeichnung die Dokument-ID trägt. Die Versionen tragen folgende Bezeichnung:

| |
|----------------------------------------|
| <code>[osguid]_[DocID].[Endung]</code> |
|----------------------------------------|

[osguid] ist die eindeutige ID der Änderung aus der Tabelle `osobjhist`.

[DocID] ist die eindeutige ID des Dokuments.

Das Öffnen von Dokumenten zum Bearbeiten und das Ändern von Dokumentdateien wird in der Bearbeitungshistorie protokolliert.

Veränderungen an Bild-Dokumenten

Öffnet ein Benutzer ein Bild-Dokument zum Bearbeiten, wird es ausgecheckt, vom Server in den CACHE-Bereich des Benutzers übertragen und im Bearbeitungsmodul geöffnet. Im Bearbeitungsmodul können Seiten hinzugefügt und Seiten gelöscht werden.

Löscht ein Benutzer Seiten, werden diese sofort aus dem CACHE-Bereich entfernt, hinzugefügte Seiten werden in den Arbeits-Bereich übernommen. Speichert der Benutzer die Änderungen, werden die Dateien zum Server übertragen und aus dem CACHE- und WORK-Bereich gelöscht. Das Dokument wird eingeecheckt. Der Server speichert alle Dateien mit dem gleichen Schema wie beim Neuanlegen.

Werden Seiten gedreht und das Dokument gespeichert, wird die gedrehte Seite neu gespeichert. Diese Änderung wird in der Bearbeitungshistorie protokolliert. Die alte Version wird ebenfalls für die Dokumentenhistorie gespeichert.

Dreht ein Benutzer eine Seite, werden Anmerkungen auf Folien für diese Seite nicht automatisch gedreht.

Speichert der Benutzer Änderungen nicht, wird das Dokument eingeecheckt, die Dateien werden aus dem CACHE-Bereich gelöscht aber nicht zum Server übertragen.

Veränderungen an Film-Dokumenten

Öffnet ein Benutzer ein Film-Dokument zum Bearbeiten, wird es ebenfalls ausgecheckt, die Datei vom Server in den CACHE-Bereich des Benutzers übertragen und im Bearbeitungsmodul geöffnet.

Über das Bearbeitungsmodul kann der Benutzer eine andere Datei auswählen. Die Datei wird in den Arbeitsbereich kopiert. Speichert der Benutzer die Änderung, wird die Datei zum Server übertragen und aus dem Arbeitsbereich gelöscht. Die alte Version wird aus dem CACHE-Bereich gelöscht. Der Server speichert die Datei mit dem gleichen Schema wie beim Neuanlegen.

Die Aktionen, zum Bearbeiten öffnen und Ändern, werden in der Bearbeitungshistorie protokolliert. Speichert der Benutzer Änderungen nicht, wird das Dokument eingeecheckt, die Datei wird aus dem CACHE-Bereich gelöscht aber nicht zum Server übertragen.

Veränderungen an Windows-Dokumenten

Windows-Dokumente werden nicht in einem Bearbeitungsmodul bearbeitet, sondern in der zugeordneten Anwendung. Öffnet ein Benutzer ein Windows-Dokument nicht schreibgeschützt, wird es ausgecheckt und die Datei vom Server in den CACHE-Bereich des Benutzers übertragen.

Über die Tabellen `ostemplate` ermittelt der Client die der Endung zugeordnete Anwendung und über die Tabelle `osapplications` den Pfad zur Anwendung, falls

die Anwendung nicht über die registrierte Anwendung gestartet werden soll. Die Datei wird in der Anwendung geöffnet und kann vom Benutzer bearbeitet werden. Speichert der Benutzer die Datei, muss sie unter der vorgegebenen Bezeichnung und dem vorgegebenem Pfad gesichert werden. Die Datei muss ebenfalls geschlossen werden. Solange Sie noch geöffnet ist, kann der Client sie nicht zum Server übertragen und einchecken.

Schließt der Benutzer das Dokument, ohne zu speichern, bleibt das Dokument ebenfalls ausgecheckt. Der Client ermittelt nicht, ob das Dokument in der Anwendung geändert wurde. Hat der Benutzer die Datei geschlossen, kann er sie einchecken oder das Auschecken zurücknehmen. Checkt er das Dokument ein, wird die Datei zum Server übertragen und aus dem CACHE-Bereich gelöscht. Die Daten in der Verschlagwortungstabelle und in der Tabelle `doclock` werden aktualisiert. Nimmt er das Auschecken zurück, wird die Datei aus dem CACHE-Bereich gelöscht und die Daten in der Datenbank ebenfalls aktualisiert.

Checkt der Benutzer das Dokument nicht ein, sondern öffnet es erneut, wird es aus dem CACHE-Bereich geöffnet.

Die Aktionen, zum Bearbeiten öffnen und Ändern, werden in der Bearbeitungshistorie protokolliert.

Aus Microsoft Office-Anwendungen kann über die enaio® office-utilities das Einchecken automatisiert werden.

Beim Beenden des Clients werden alle ausgecheckten Dokumente automatisch eingchecked. Sind ausgecheckte Dokumente noch in der Anwendung geöffnet, wird der Benutzer aufgefordert, die Dokumente zu schließen und die Dokumente einzuchecken oder das Auschecken zurückzunehmen.

Variantenverwaltung

Zu Windows-Dokumenten können Varianten erstellt werden.

Varianten sind Kopien von Windows-Dokumenten, die keinen eigenen Standort erhalten sondern nur über das Fenster 'Variantenverwaltung' des Originaldokuments zugänglich sind.

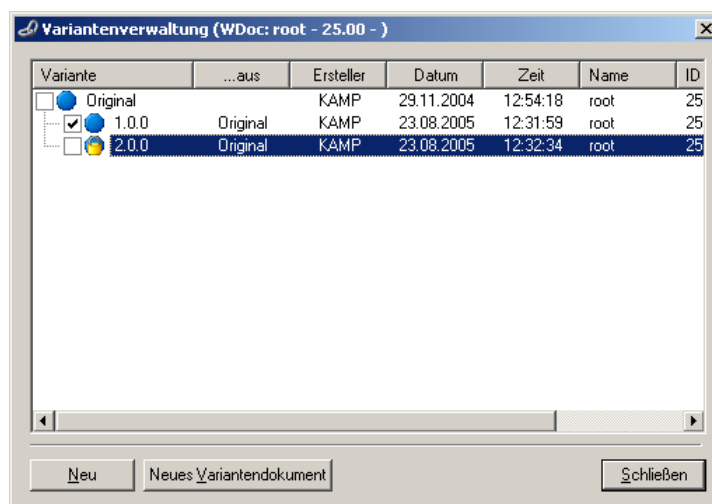


Abbildung 25

Erstellt ein Benutzer eine Variante, wird eine Kopie des Ausgangsdokumentes erzeugt, ausgecheckt und geöffnet. Das Dokument muss nach dem Speichern geschlossen und eingchecked werden. Von Varianten können weitere Varianten erstellt werden. Jede Variante hat eine eigene Bearbeitungshistorie.

Die Variante ist verschlagwortet wie das Ausgangsdokument. Die Verschlagwortung ist in der Tabelle `object#` gespeichert. Sie kann geändert werden, ist aber nicht recherchierbar. Das Windows-Dokument, von dem die erste Varianten erstellt wird, erhält in der Spalte `version` der Verschlagwortungstabelle `object#` den Eintrag 1. Innerhalb der Variantenverwaltung ist es als Original bezeichnet. Nur das Original ist recherchierbar. Nur über das Original kann auf die Variantenverwaltung zugegriffen werden.

Bei Varianten ist in der Spalte `version` der Verschlagwortungstabelle die eindeutige ID des Originals eingetragen. Varianten haben keinen eigenen Standort, also keinen Eintrag in der Tabelle `sdrel`.

Die weiteren Daten für die Variantenverwaltung stehen in der Tabelle `osdocver`.

Die Tabelle hat folgende Struktur:

| Spalte | Inhalt |
|-------------------------|-----------------------------------------|
| <code>doc_id</code> | die eindeutige ID des Dokuments |
| <code>doc_parent</code> | die eindeutige ID des Ausgangsdokuments |
| <code>doc_act</code> | Kennzeichnung des aktuellen Dokuments |
| <code>doc_ver</code> | die Variantenbezeichnung |
| <code>doc_root</code> | die eindeutige ID des Originals |

Tabelle 33

Für Originale ist in der Spalte `doc_parent` eine 0 eingetragen.

Ein Dokument aus der Variantenverwaltung kann als aktuelles Dokument gekennzeichnet werden. Für dieses Dokument in der Spalte `doc_act` eine 1 eingetragen. Das so gekennzeichnete Dokument wird geöffnet, wenn aus einer Trefferliste oder einem Ordnerfenster das Windows-Dokument geöffnet wird. In Trefferlisten oder Ordnerfenstern selbst wird immer die Verschlagwortung des Originals angezeigt.

Dateiverwaltung und Speicherort entsprechen denen anderer Dokumente.

Wird über eine Trefferliste oder das Ordnerfenster das Original gelöscht, werden auch alle Varianten gelöscht. Über das Fenster 'Variantenverwaltung' können die Varianten gelöscht werden, allerdings nicht das Original.

Anmerkungen auf Folien

Schwarz/Weiß-, Graustufen- und Farbbilddokumenten können mit Anmerkungen auf Folien versehen werden. Das Anlegen und Bearbeiten von Folien wird nicht in der Bearbeitungshistorie protokolliert.

Öffnet ein Benutzer eine Datei, der eine Folie zugeordnet ist, die er nicht bearbeiten darf, wird die Folie vor dem Übertragen in den CACHE-Bereich des Benutzers in die Datei eingebrannt. Die Folie kann nicht entfernt werden. Die Datei wird so versendet, gedruckt und exportiert. Kann ein Benutzer Dokumentdateien ändern, deren Folien er nicht bearbeiten darf, bearbeitet er die Dateien aus dem CACHE-Bereich mit eingebrannten Folien. Die Dateien werden so zum Server übertragen und gespeichert. Diese Möglichkeit sollte über das Rechtesystem ausgeschlossen werden. Bearbeitet ein Benutzer eine Folie, wird sie nicht ausgecheckt. Ein anderer Benutzer erkennt nicht, dass die Folie bearbeitet wird und kann sie ebenfalls bearbeiten. Gespeichert wird die Folie in der Version, die zuletzt gespeichert wird.

Folien werden in der Datenbank über die Tabellen `annotations` und `ann_groups` verwaltet. Die Tabelle `annotations` enthält die Daten, die Tabelle `ann_groups` enthält für Gruppenfolien die Gruppenzuordnung.

Die Tabelle `annotations` hat folgende Struktur:

| Spalte | Inhalt |
|---------------|-------------------------------------------------|
| annotation_id | die eindeutige ID der Folie |
| object_id | die eindeutige ID des Dokuments |
| page | die Seite, der die Folie zugeordnet ist |
| user_id | der Benutzer, der die Folie angelegt hat |
| flag | der Typ der Folie |
| zeitstempel | der Zeitpunkt, zu dem die Folien angelegt wurde |
| timeout | nicht verwendet |
| annotation | die Folien- und Anmerkungsdaten |

Tabelle 34

Die Tabelle `ann_groups` hat folgende Struktur:

| Spalte | Inhalt |
|---------------|----------------------------------------------------------------------|
| gruppen_id | die eindeutige ID einer Gruppe, der die Gruppenfolie zugeordnet ist. |
| annotation_id | die eindeutige ID der Gruppenfolie |

Tabelle 35

Anmerkungen auf Folien sind nicht Bestandteil der Bearbeitungshistorie. Dokumente werden dort immer mit Folien in der aktuellen Version angezeigt.

Elektronische Signatur

Für die elektronische Signatur wird am Arbeitsplatz ein installiertes Kartenlesegerät, eine Signaturkarte, der PIN und ein installierter Adobe Reader sowie die entsprechende Lizenz benötigt.

Der enaio® client signiert Dokumente, die im PDF-Format vorliegen, über eine Zusammenarbeit mit dem Adobe Reader. Für diese Signatur kann ein Signaturtyp im enaio® administrator eingerichtet werden.

In der Datenbank werden die Daten der Signaturtypen in der Tabelle `ossigtext` verwaltet. Diese Tabelle hat die folgende Struktur:

| Spalte | Inhalt |
|-----------|--------------------------------------------------------|
| ostextid | die eindeutige ID eines Signatortextes |
| oslabel | eine Kurzbezeichnung für den Signatortext |
| ostext | der Signatortext |
| osdeleted | Kennzeichnung, ob der Signatortext noch verwendet wird |

Tabelle 36

Will ein Benutzer ein Dokument digital signieren, so markiert er das entsprechende Dokument und startet die Digitale Signatur über das Kontextmenü oder den Tastaturkürzbefehl STRG-S. Dabei werden die Dateien in den CACHE-Bereich übertragen. Anmerkungen auf Folien, die der Benutzer nicht bearbeiten darf, werden eingebrannt.

PDF-Dokumente werden im Signaturmodul geöffnet.

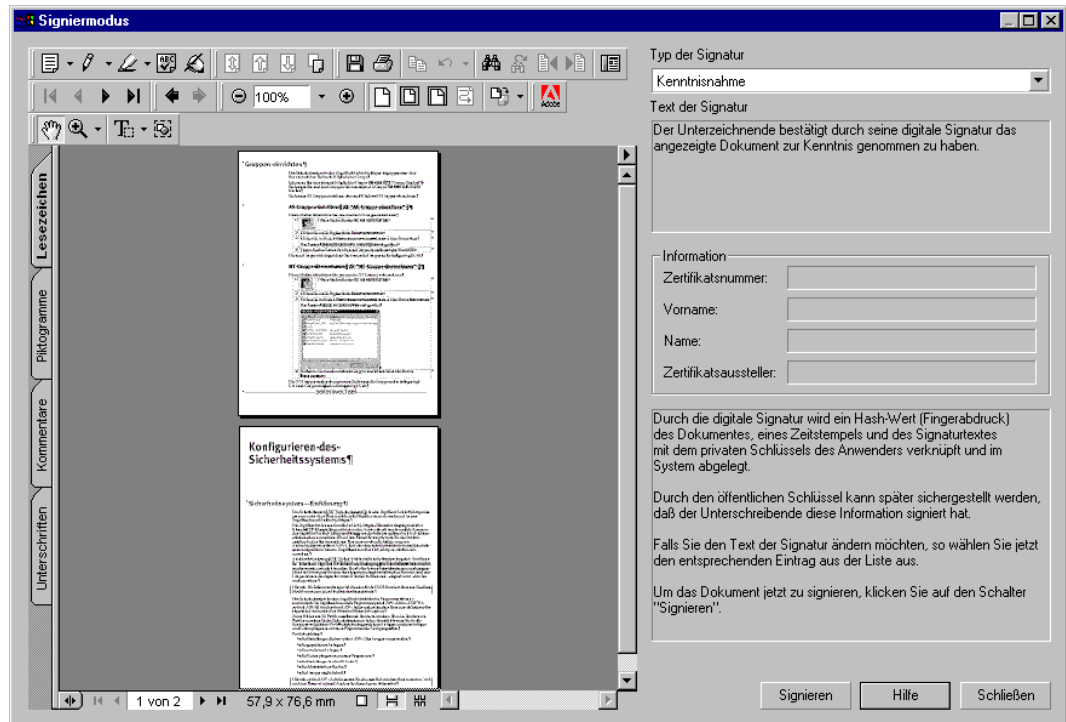


Abbildung 26

Der Benutzer wählt den Typ der Signatur aus und bestätigt die elektronische Signatur durch seine Signaturkarte und den PIN.

Die Signatur wird anschließend in der erzeugten PDF-Datei gespeichert.

Die PDF-Datei wird, wie andere Versionen auch, beim Server in einem Verzeichnis gespeichert, das auf gleicher Ebene wie die Dokumentendateien angelegt wird und als Bezeichnung die Dokument-ID trägt. Die Datei selbst erhält, ebenfalls wie andere Versionen, als Bezeichnung die eindeutige ID der Änderung aus der Tabelle `osobjhist`, gefolgt von der eindeutigen ID des Dokuments.

Elektronische Signaturen von PDF-Dateien werden über die Bearbeitungshistorie verifiziert. Für die Verifizierung wird die PDF-Datei im Signaturmodul geöffnet und Signaturtext und Signaturheader aus der Datenbank angezeigt. Die Signatur wird mit dem public key entschlüsselt. Überprüft wird, ob der Hash-Wert über Datei, Signaturtext und Signaturheader mit dem entschlüsselten Hash-Wert aus der Datenbank übereinstimmt.

Wird ein Dokument gelöscht, von dem eine digital signierte Version existiert, wird die digital signierte Version ebenfalls gelöscht.

Für die elektronische Signatur von W-Dokumenten, E-Mail-Dokumenten und XML-Dokumenten stehen keine Signaturtypen zur Verfügung. Für Dokumente dieser Dokumenttypen wird eine externe Signatur - eine PKCS7-Datei - erzeugt und zusammen mit dem signierten Dokument im WORK-Bereich gespeichert.

Bei der Signatur kann das zu signierende Dokument nicht gleichzeitig angezeigt werden. Der Benutzer erhält einen entsprechenden Hinweis.

Verifiziert werden externe Signaturen ebenfalls über die Bearbeitungshistorie.

Über einen Eintrag in der Datei `as.cfg` kann der Algorithmus - SHA-256, SHA-386 oder SHA-512 - eingestellt werden.

Löschen von Objekten

Die Zugriffsrechte auf Ordner, Register und Dokumente im enaio® administrator legen die Löschrechte fest. Wie alle Zugriffsrechte können die Löschrechte über Klauseln abhängig von der Verschlagwortung sein. Für das Löschen von DMS-Objekten können benutzerspezifisch Bestätigungsdialoge eingeschaltet werden. Es wird vom Benutzer in den Papierkorb gelöscht. Jeder Benutzer hat einen eigenen Papierkorb.

Einstellungen für Bestätigungsdialoge auf der Registerkarte 'Bestätigen' werden in der Tabelle `osconf` mit den Daten des Konfigurationstyps 3 gespeichert und können über die Profilverwaltung verteilt werden.

Beim Löschen von Ordnern und Registern wird rekursiv der gesamte Inhalt ebenfalls gelöscht. Deshalb benötigt der Benutzer die Löschrechte für alle enthaltenen Objekte. Archivierte Dokumente können nicht gelöscht werden. Enthalten Ordner oder Register archivierte Dokumente, müssen diese vorher verschoben werden.

Dokumente, die ausgecheckt sind, können von allen Benutzern gelöscht werden. Unabhängig von den Einstellungen für Bestätigungsdialoge wird der Benutzer informiert, welcher Benutzer wann und an welcher Arbeitsstation das Dokument zum Bearbeiten geöffnet hat und muss das Löschen bestätigen.

In einem nächsten Schritt können Objekte auch aus dem Papierkorb gelöscht werden. Dazu gibt es einen Systempapierkorb, in dem die Papierkörbe der einzelnen Benutzer zusammengefasst sind. Benutzer mit speziellen Rechten können sich den Systempapierkorb anschauen und aus diesem heraus die Objekte vollständig aus dem System löschen. Beim Löschen von DMS-Objekten löscht der enaio® client alle Datenbankinträge, die sich auf die Objekte beziehen.

Die folgenden Tabellen werden aktualisiert:

- § die Verschlagwortungstabellen `stamm#`, `register#`, `object#`,
- § die Schattentabellen `stamm#s`, `register#s`, `object#s`,
- § die Standorttabellen `sdrel` und `osparregrel`,
- § die Tabelle `mdrel`, falls Objekte in Mappen verwaltet werden,
- § die Tabelle `osdigsigs` für signierte Dokumente,
- § die Tabelle `osdocver` für Varianten,
- § die Tabelle `remarks` für Textnotizen,
- § die Tabelle `hyplnk` für Verknüpfungen,
- § die Tabellen `annotations` und `ann_groups` für Anmerkungen auf Folien,
- § die Tabelle `doclock`, falls Dokumente ausgecheckt sind,
- § die Tabelle `osrevisit` für Wiedervorlagen und aktuelle Abonnements.

Bei Verweisdokumenten wird nur der Standorteintrag in der Tabelle `sdrel` gelöscht.

Die Tabelle `osabonnement` für abonnierte Objekte wird von enaio® server aktualisiert.

Die Tabelle `osdigsigs` für signierte Dokumente wird nicht aktualisiert.

Textnotizen und Dokumentdateien, Versionen, signierte Versionen und Varianten werden vom Server gelöscht. Ebenfalls alle Dateien, die für die Volltextindexierung erstellt wurden.

Dateien aus dem CACHE-Bereich löscht der Client erst beim Beenden.

Überprüft wird nicht, ob in der Objektsuche oder auf der Suchleiste Verweise auf gelöschte Objekte existieren. Überprüft wird ebenfalls nicht, ob Verschlagwortungen gelöschter Objekte mit Verschlagwortungen anderer Objekte verknüpft sind.

Bearbeitungshistorie

Die Bearbeitungshistorie eines Objekts zeigt die Einträge für das Objekt aus der Tabelle `osobjhist` an und erlaubt, Verschlagwortungsversionen, bei Dokumenten ebenfalls Dateiversionen wieder herzustellen, falls Objekttypen die entsprechenden Eigenschaften im enaio® editor erhalten. Die Historien können im enaio® editor global für den Objekttyp aktiviert/deaktiviert werden, es ist aber auch eine Einstellung möglich, die es jedem Benutzer (sofern er die nötige Systemrolle besitzt, wird im Eigenschaftsfenster des Dokuments die entsprechende Schaltfläche eingeblendet) ermöglicht für jedes Dokument jederzeit beide Historien zu aktivieren oder zu deaktivieren.

Bei digital signierten Dokumenten kann über die Bearbeitungshistorie die Signatur verifiziert werden.

Wiederherstellen der Verschlagwortung

Änderungen an der Verschlagwortung sind in der Tabelle `osobjhist` mit der Aktions-ID 3 protokolliert und werden in der Bearbeitungshistorie mit dem Eintrag 'Indexdaten geändert', Zeitpunkt und Benutzer aufgeführt.

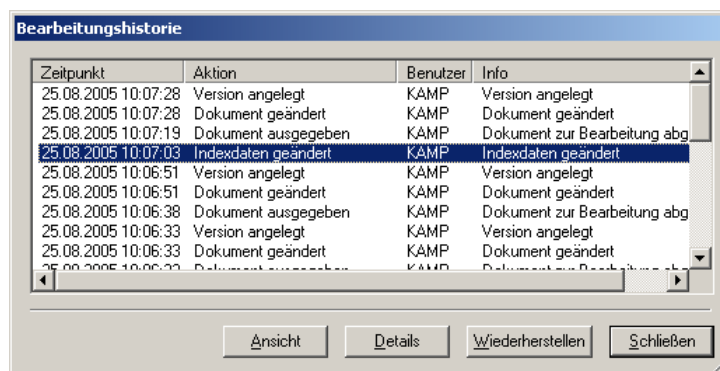


Abbildung 27

Ist dieser Eintrag markiert, kann die Verschlagwortung vor der Änderung über die Schaltfläche Ansicht eingesehen werden. Sie wird aus der Schattentabelle gelesen. Über die Schaltfläche 'Wiederherstellen' wird die Verschlagwortung vor der Änderung auf dem Datenblatt editierbar angezeigt. Sie kann bearbeitet und gespeichert werden.

Beim Speichern wird die Verschlagwortung in die Verschlagwortungstabelle eingetragen, die Verschlagwortung vor der Änderung in die Schattentabelle. Die Änderung wird in die Tabelle `osobjhist` mit der Aktions-ID 3 und der Information (`osinfo`) 'Wiederherstellung der Indexdaten vom 'Datum' 'Uhrzeit' eingetragen.

Wiederherstellen von Versionen

Änderungen an Dokumentendateien sind in der Tabelle `osobjhist` mit der Aktions-ID 4 protokolliert und werden in der Bearbeitungshistorie mit dem Eintrag 'Version angelegt', Zeitpunkt und Benutzer aufgeführt.

| Zeitpunkt | Aktion | Benutzer | Info |
|---------------------|---------------------|----------|------------------------------|
| 25.08.2005 10:07:28 | Version angelegt | KAMP | Version angelegt |
| 25.08.2005 10:07:28 | Dokument geändert | KAMP | Dokument geändert |
| 25.08.2005 10:07:19 | Dokument ausgegeben | KAMP | Dokument zur Bearbeitung abg |
| 25.08.2005 10:07:03 | Indexdaten geändert | KAMP | Indexdaten geändert |
| 25.08.2005 10:06:51 | Version angelegt | KAMP | Version angelegt |
| 25.08.2005 10:06:51 | Dokument geändert | KAMP | Dokument geändert |
| 25.08.2005 10:06:38 | Dokument ausgegeben | KAMP | Dokument zur Bearbeitung abg |
| 25.08.2005 10:06:33 | Version angelegt | KAMP | Version angelegt |
| 25.08.2005 10:06:33 | Dokument geändert | KAMP | Dokument geändert |
| 25.08.2005 10:06:33 | Dokument ausgegeben | KAMP | Dokument zur Bearbeitung abg |

Abbildung 28

Ist dieser Eintrag markiert, kann die Version vor der Änderung über die Schaltfläche 'Ansicht' eingesehen werden. Die Version wird schreibgeschützt in den Arbeitsbereich `temp\ostemp` übertragen und geöffnet.

Über die Schaltfläche 'Wiederherstellen' wird das aktuelle Dokument als Version gespeichert und die Version vor der Änderung als aktuelles Dokument gespeichert. Die Änderung wird in die Tabelle `osobjhist` mit der Aktions-ID 4 und der Information (`osinfo`) 'Wiederherstellung des Dokuments vom 'Datum' 'Uhrzeit'' eingetragen.

Volltextsuche

Überblick

enaio® bietet die Möglichkeit, Dokumente in Zusammenarbeit mit Indexdiensten zu indexieren und aus dem enaio® client über den Volltextindex zu recherchieren. Dabei können sowohl Dokumente beim Neuanlegen automatisch, wie auch bereits bestehende Dokument nachträglich über eine automatische Aktion indexiert werden.

Indexiert werden können sowohl Inhalt als auch die Verschlagwortung der Dokumente. Sie können ebenfalls die Verschlagwortung von Ordnern und Registern indexieren.

In Zusammenarbeit mit der Texterkennungssoftware FineReader können auch Bild-Dokumente, insbesondere D-Dokumenttypen, mit Hilfe der Texterkennung in ASCII-Dateien gewandelt und dann indexiert werden.

Der von enaio® unterstützten Indexdienste ist Elasticsearch.

Genaue Informationen finden Sie im Handbuch 'enaio® fulltext'.

Sicherheit

Überblick

DMS verwalten zum Teil Daten und Dokumente, die im Idealfall nicht mehr in Papierform vorliegen. Somit ist die Sicherheit der Aufbewahrung und der Wiederverfügbarkeit der digitalen Daten von großer Bedeutung.

Dokumente in digitaler Form sind folgenden Risiken ausgesetzt:

- § Es besteht die Gefahr, dass das DMS oder die verwalteten Daten durch Hardwaredefekte, Software-Fehlfunktionen oder externe Angriffe beschädigt werden, so dass eine Reproduktion der Daten unmöglich ist. Dem kann begegnet werden, indem Mechanismen eingesetzt werden
 - § zur Wiederherstellung der Daten und Dokumentdateien (Datenbank- und WORK-Verzeichnis-Backup)
 - § und zur Wiederherstellung des DMS (Programmverzeichnis-Backup)
- § Eine weitere Gefahr besteht darin, dass Dokumentdateien oder Indexdaten verfälscht oder mit DMS-eigenen Mitteln gelöscht werden können. Dem kann folgendermaßen begegnet werden:
 - § Eindeutige Identifizierung von Dokumentdateien (Elektronische Signatur)
 - § Wiederherstellung älterer Objektzustände (Objekthistorie)
- § Die verwalteten Objekte sollen nur berechtigten Personen zugänglich sein, Außenstehende sollen diese nicht sehen dürfen. Es kann von unberechtigter Seite durchaus Interesse daran bestehen, diese Objekte einzusehen. Daher sind die Objekte gegen unerlaubte externe Reproduktion zu schützen. Folgende Schutzmechanismen können zu diesem Zweck eingesetzt werden:
 - § Kapselung der Bereiche, die Dokumentdateien enthalten (Serverarchitektur)
 - § Verschlüsselung der Dokumentdateien
 - § Absicherung auf Netzwerk-, Domänen- und Betriebssystem-Ebene
 - § Authentifizierung beim Zugriff auf das DMS
 - § Authentifizierung beim Zugriff auf die Datenbank
- § Eine weitere Gefährdung besteht durch den illegitimen internen Zugriff, wobei korrekt authentifizierte Benutzer Objekte sehen, die sie nicht sehen sollen. Gegenmaßnahmen sind:
 - § Eine nach Objekten strukturierte Authentifizierung der Benutzer
 - § Modifizierung der Zugriffsrechte mit logischen Klauseln

Die Datensicherheit lässt sich einerseits durch Maßnahmen erhöhen, die außerhalb des enaio®-Systems liegen. Andererseits sind zur Sicherstellung eines Maximums an Datensicherheit im enaio®-System eine Reihe an internen Systemfunktionen vorhanden.

Schutz von Daten und Dokumentdateien

Absicherung auf Netzwerk-, Domänen- und Betriebssystem-Ebene

Weitere externe Vorgangsweisen, die nicht durch das DMS beeinflusst werden können, betreffen die Absicherung des Systemumfelds, in dem das enaio®-System eingebettet ist. Dies sind insbesondere das Netzwerk, die Domäne und die einzelnen Stationen, an denen enaio®-Komponenten betrieben werden. Diese Maßnahmen sollen nur kurz angesprochen werden. Detaillierte Informationen dazu finden sich in einschlägiger Literatur.

Das vorhandene Netzwerk ist vor allem gegenüber dem Internet abzusichern, hierfür bieten sich Firewalls an, die den Datenverkehr zwischen Netzwerk und Internet überwachen, sowie Firewall- und Antivirus-Software, die zentral oder auf den einzelnen Stationen installiert ist. Eine wichtige Maßnahme ist die Organisation von Arbeitsstationen in Domänen, die eine zentrale Benutzerverwaltung haben. Diese Benutzerverwaltung erschwert es unbefugten Benutzern, in verbotene Bereiche, z.B. den WORK-Bereich eines Anwendungsservers, einzudringen.

Voraussetzung ist die korrekte Einrichtung der Benutzerverwaltung, in der die Benutzer gezwungen werden, sichere Passwörter zu verwenden. Dasselbe gilt für die einzelnen Server- und Arbeitsstationen, an denen ebenfalls darauf zu achten ist, dass die vorhandenen Benutzerkonten ausreichend geschützt sind.

Absicherung durch das enaio®-System

Kapselung von Server-Dateibereichen

Der gemeinsame Zugriff auf Dateien wird in Netzwerken üblicherweise über Verzeichnisfreigaben geregelt. Da die Verzeichnisse für jeden einzelnen Benutzer, der Zugriff haben soll, freigegeben werden müssen, ist dieses Verfahren einerseits mit Risiken behaftet, und andererseits pflegeaufwendig. Mit der Kapselung der Dateibereiche von Anwendungsservern wird für beide Problembereiche die Situation verbessert.

Die wichtigen Dateibereiche sind der WORK-Bereich, der CACHE-Bereich und die Archivmedien. Bei einem einzelnen Anwendungsserver können alle Bereiche lokal vorhanden sein. Work- und CACHE-Bereich sind dabei Unterverzeichnisse des Serververzeichnisses, die Jukebox mit den Archivmedien ist lokal angeschlossen und über einem Laufwerksbuchstaben ansprechbar. Für diese Konstellation ist keine Verzeichnisfreigabe erforderlich. Den Zugriff auf die Dokumentdateien vermittelt der Anwendungsserver über den freigegebenen IP-Port. Alle Dokumentdateien werden durch den Anwendungsserver übertragen.

Sobald ein zweiter Anwendungsserver in derselben Servergruppe, aber an einem anderen Rechner installiert ist, ist auch in enaio® eine Verzeichnisfreigabe für den Work- und den Serverbereich, sowie je eine Freigabe für jede DMS-Jukebox erforderlich. Diese Freigaben beschränken sich jedoch auf das Benutzerkonto, mit dem die Anwendungsserver-Dienste gestartet werden. Durch die Beschränkung der Freigabe auf ein Benutzerkonto werden sowohl die Risiken minimiert, als auch die Verwaltung vereinfacht.

Detaillierte Informationen zur Kapselung der Server-Dateibereich erhalten Sie im Kapitel 'Systemarchitektur' und im Handbuch zum enaio® server.

Verschlüsselung von Dokumenten

Für jeden enaio®-Dokumenttyp kann die Verschlüsselung der Dokumentdateien eingerichtet werden. Dazu ist lediglich im enaio® editor die Verschlüsselungseigenschaft des Objekttyps entsprechend einzurichten.

Die Dokumentdateien werden in einem symmetrischen Verfahren verschlüsselt. Dokumente, für die die Verschlüsselung eingeschaltet ist, werden von der erstellenden Komponente verschlüsselt, zum Anwendungsserver übertragen und vom Anwendungsserver in den Dateibereichen abgelegt. Der Anwendungsserver kann ebenfalls Dokumentdateien, die zu ihm übertragen werden, in einem symmetrischen Verfahren verschlüsseln. Beide Arten der Verschlüsselung verwenden eigene Schlüssel und können unabhängig oder zusammen eingesetzt werden.

Wenn ein autorisierter Benutzer ein solches Dokument sehen oder bearbeiten will, dann wird es vom Anwendungsserver verschlüsselt zur Clientanwendung übertragen, vom Client entschlüsselt und angezeigt.

Damit ist sichergestellt, dass jede Übertragung des Dokuments im Netzwerk verschlüsselt erfolgt. Angreifer, die unerlaubt Zugang zur Kommunikation im Netzwerk haben, haben dadurch keinen Zugriff auf die übertragenen Dokumente.

Lizenzsystem

Obwohl das enaio®-Lizenzsystem nicht primär dafür ausgelegt ist, so hat es doch einen für die Datensicherheit positiven Effekt. Die Lizenzierung erlaubt es, dass Anwendungsserver nur auf durch GUID eindeutig identifizierte Serverrechner ausgeführt werden können. Die eigentliche Containerdatei, die die Lizenzen enthält, ist verschlüsselt. Beides macht die Nachbildung eines konkreten enaio®-Systems sehr schwierig, wenn nicht unmöglich.

Darüber hinaus kann das enaio®-Lizenzsystem so eingerichtet werden, dass auch die Stationen, an denen Clientanwendungen ausgeführt werden sollen, durch GUID identifiziert werden. Bei jedem Start einer Clientanwendung prüft der Anwendungsserver dann, ob die Station, an der die Anwendung ausgeführt wird, zur Ausführung berechtigt ist. Wenn nicht, wird die Lizenz verweigert. Die Kommunikation mit dem Anwendungsserver ist dann nicht mehr möglich. Das Lizenzsystem ist somit ein Authentifizierungs-System, das die Anmeldung von Anwendungen an den Anwendungsserver erzwingt – ein weiterer Sicherheitsfaktor.

enaio®-Authentifizierung

enaio® bietet ein Sicherheitssystem, mit dem differenzierte Zugriffsberechtigungen auf Objekttypen und Anwendungen vergeben werden können. Anwendungsrechte, genannt Systemrollen, werden an einzelne Benutzer vergeben. Berechtigungen für Objekttypen werden über Benutzergruppen vergeben.

Benutzer

Jeder Benutzer muss eigens angelegt werden, es sei denn, es erfolgt eine Übernahme von Benutzern aus einem LDAP-Verzeichnisdienst. Jeder Benutzer erhält einen Namen und ein vorgegebenes Passwort, sowie verschiedene Eigenschaften. Dazu gehört die Zuordnung zu Benutzergruppen, die Festlegung von Systemrollen, und, ob der Benutzer als Profilbenutzer dienen soll.

Wenn Benutzer aus einem LDAP-Verzeichnisdienst übernommen werden, erfolgt die Kennwortverwaltung nicht im enaio®-System, sondern im LDAP-Dienst. Die weiteren Eigenschaften, d.h. Zuordnung zu Benutzergruppen usw., müssen weiterhin im enaio®-System eingestellt werden.

Benutzerprofile

Eingerichtete Benutzer können als Benutzerprofile definiert und neuen Benutzern zugeordnet werden. Mit der Zuordnung eines Benutzers zu einem Benutzerprofil werden Einstellungen des Profilbenutzers an den neuen Benutzer übergeben. Dazu gehören z.B. die Systemrollen, Gruppenzugehörigkeit, aber auch Einstellungen im enaio® client und enaio® capture.

Systemrollen

In den Einstellungen zu Systemrollen werden Ausführungsrechte in den enaio®-Anwendungen enaio® administrator, enaio® start, enaio® editor, enaio® capture und enaio® client festgelegt. Im enaio® client kann z. B. über Systemrollen gesteuert werden, ob ein Benutzer Notizen bearbeiten oder ob er Bearbeitungshistorien einsehen darf.

Benutzergruppen

Benutzer können in Benutzergruppen zusammengefasst werden. Über Benutzergruppen wird die Erteilung von Objektrechten geregelt. Wenn ein Benutzer keiner Benutzergruppe zugeordnet wird, dann erhält er auch keine Rechte auf DMS-Objekte. Dies kann z.B. für Benutzer sinnvoll sein, die reine Administrationstätigkeiten durchführen, aber keine DMS-Objekte bearbeiten sollen. Benutzer können mehreren Benutzergruppen zugeordnet sein. Solche Benutzer erhalten Rechte auf Objekte aus allen zugeordneten Benutzergruppen, wobei immer die umfangreichsten Rechte übernommen werden.

Objektrechte

Für jeden erstellten Objekttyp und damit für alle Objekte dieses Objekttyps gibt es eine Reihe von Rechten. Allen Objekten sind die Rechte Indexdaten anzeigen (R), Indexdaten schreiben (W), Objekt löschen (D), Objekt ausgeben (X) und Objekt schreiben (U) gemeinsam. Bei Dokumenten gibt es zusätzlich die Rechte Gruppenanmerkungen (G) und Öffentliche Anmerkungen (P).

Mit diesen Objektrechten ist eine differenzierte Regelung für den Zugriff von Benutzern auf DMS-Objekte möglich.

Objektrechte sind nur für Objekttypen einstellbar. Für einzelne DMS-Objekte können keine Rechte erstellt werden. Private Objekte sind daher mit diesen Mitteln nicht erstellbar. Dies lässt sich mit Klauseln erreichen, die weiter unten beschrieben werden.

Klauseln

Für jedes einzelne Zugriffsrecht auf einen Objekttyp können Klauseln formuliert werden, mit denen die Rechte weiter eingeschränkt werden können. Wenn z.B. für alle Rechte eines Objekts eine passende Klausel formuliert wird, dann darf nur der Anleger des Objekts das Objekt sehen und bearbeiten. Die Klausel für diesen Zweck würde wie folgt lauten:

#ANLEGER# = #BENUTZER#

Kontensperrung

Fehlgeschlagene Anmeldeversuche können ein Indikator für Angriffsversuche auf das System sein. Daher ist für enaio® die Behandlung von fehlgeschlagenen Anmeldeversuchen einstellbar. Im Gesamtsystem des enaio® administrator sind für das Startverhalten folgende drei Sicherheitsstufen einstellbar:

§ Keine Einschränkung.

Es erfolgt keinerlei Reaktion auf fehlgeschlagene Anmeldeversuche, d.h. der Anmeldedialog wird immer wieder – mit dem entsprechenden Hinweis auf fehlerhafte Benutzernamen oder Passwörter – angezeigt.

§ Beendigung der Anwendung nach 3 Fehlversuchen.

Nach drei fehlgeschlagenen Anmeldeversuchen wird bei Start von Clientanwendungen der Anmeldedialog nicht mehr angezeigt. Die jeweilige Anwendung muss erneut gestartet werden. Bei der Neuansmeldung im enaio® client wird die Anmeldung abgebrochen und der aktuell angemeldete Benutzer bleibt angemeldet. Beim Abmelden aus dem Client wird nach dreimaligem Fehlversuch der Client beendet.

§ Sperrung des Benutzerkontos nach 3 Fehlversuchen.

Gesperrte Benutzerkonten können nur im enaio® administrator wieder freigegeben werden.

Detaillierte Informationen zum enaio®-Sicherheitssystem erhalten Sie im Handbuch zum enaio® administrator und im Kapitel 'Administration' dieses Systemhandbuches.

Objektidentität

Die wichtigste Funktion eines DMS-Systems ist es sicherzustellen, dass die verwalteten Objekte, also Daten und Dokumentdateien, unverfälscht reproduziert werden können. Das setzt einiges voraus, da die DMS-Objekte während ihrer Verarbeitung in ihrer elektronischen Repräsentation an ganz verschiedenen Stellen lokalisiert sind. Sie werden im Arbeitsspeicher eines Computers angelegt, auf Festplatte zwischengespeichert, über das LAN zum Anwendungsserver übertragen, dort im WORK-Bereich des Servers auf Festplatte geschrieben. Beim Öffnen geht es den Weg zurück. Irgendwann werden die Dokumentdateien archiviert, d.h. auf optische Medien übertragen.

Mit der Verschlagwortung ist es ähnlich, sie wird auch zunächst an einem Client erzeugt, über das Netz zum Anwendungsserver übertragen, weiter über das Netz an die Datenbank gereicht usw. Im Falle einer Havarie, z.B. des Defekts der Serverplatte, existiert das Objekt plötzlich nur noch auf einem Sicherungsband. Man kann von einer verteilten Präsenz der DMS-Objekte sprechen.

Damit die Objekte bei dieser Art von Präsenz überhaupt unverfälscht existieren können, gelten zahlreiche Voraussetzungen, die nicht detailliert überprüfbar sind. Der Arbeitsspeicher muss funktionieren, das Betriebssystem und die Clientanwendung müssen die Daten korrekt verwalten, das Netzwerkprotokoll und die Netzwerkhardware müssen die Daten korrekt übertragen, die Festplatten, Jukeboxen und optischen Medien müssen funktionieren und die Datenbank muss die Daten fehlerfrei verwalten.

All diese Voraussetzungen sind nicht im Einzelfall überprüfbar. Unzählige Erfahrungen im Umgang mit diesen Komponententypen zeigen jedoch, dass die Voraussetzungen normalerweise erfüllt sind.

Daher sollen im Folgenden nur die Funktionen beschrieben werden, die enaio® zur Sicherstellung der Objektidentität bereitstellt. Diese sind die Datenhaltung in der Datenbank, die Elektronische Signatur, die Dokumenthistorie und die Archivierung.

Datenhaltung in der Datenbank

Systeminformationen und Objektdaten werden in einem leistungsfähigen SQL-Datenbanksystem gespeichert. enaio® unterstützt MS SQL Server und Oracle. Dies sind bewährte Datenbanksysteme, die ein Höchstmaß an Datenintegrität bieten.

Dennoch ist auch bei diesen Datenbanksystemen auf ausreichende Authentifizierung und Datensicherung zu achten. Das sind Maßnahmen, die wiederum außerhalb des Einflussbereichs des enaio®-Systems liegen.

Es wird dringend davon abgeraten, den Datenbankzugriff über einen Datenbank-Benutzer zu erlauben, für den kein Passwort erforderlich ist. Mit einem solchen Benutzerkonto steht die Datenbank im Netz offen, alle Daten sind ungeschützt. Darüber hinaus sollten nur sichere Passwörter mit ausreichender Länge und Komplexität zugelassen werden.

Dokumentdateien: Elektronische Signatur

Die Elektronische Signatur ist die elektronische Entsprechung der Unterschrift auf einem Papierdokument, mit der die Echtheit eines elektronischen Dokuments bestätigt wird.

Technische Voraussetzungen

Wenn die technischen Voraussetzungen gegeben sind, ist es in enaio® möglich, jedes Dokument mit Ausnahme von XML-Dokumenten, Film-Dokumenten und E-Mails elektronisch zu signieren und elektronisch signierte Dokumente zu verifizieren. Um die Elektronische Signatur nutzen zu können, ist ein installiertes Kartenlesegerät, eine Signaturkarte, die PIN (Geheimzahl), die entsprechende enaio®-Lizenz und ein installierter Adobe Reader erforderlich. Voraussetzungen für die Erzeugung von Elektronischen Signaturen sind:

- § Installierte PKCS#11-Bibliothek `pkcs11api.dll` Version 2.10, basierend auf dem Cryptographic Token Interface Standard PKCS#11
- § Via PCSC auf dem System installierter Smart Card Reader
SignTrust hat für Ihre PKCS#11-Bibliothek folgende Kartenleser getestet:
Utimaco CardMan, Siemens-Nixdorf B1 V4.0 und Siemens B1 Snuggle Desktop. (Während der Realisierungsphase wird ein CARDMAN 2010 eingesetzt).
- § Signaturkarte eines Trustcenter (Während der Realisierungsphase stehen Karten von Medizon zur Verfügung)
- § HTMLDOC zur Konvertierung von HTML-Dokumenten nach PDF
- § PDF-Creator für Office zur Konvertierung von Office-Dokumenten nach PDF, alternativ der enaio® printer
- § Installierter Adobe Reader ab Version 4.0

Erstellung der Elektronischen Signatur

Die Elektronische Signatur wird wie in folgender Abbildung gezeigt, erstellt:

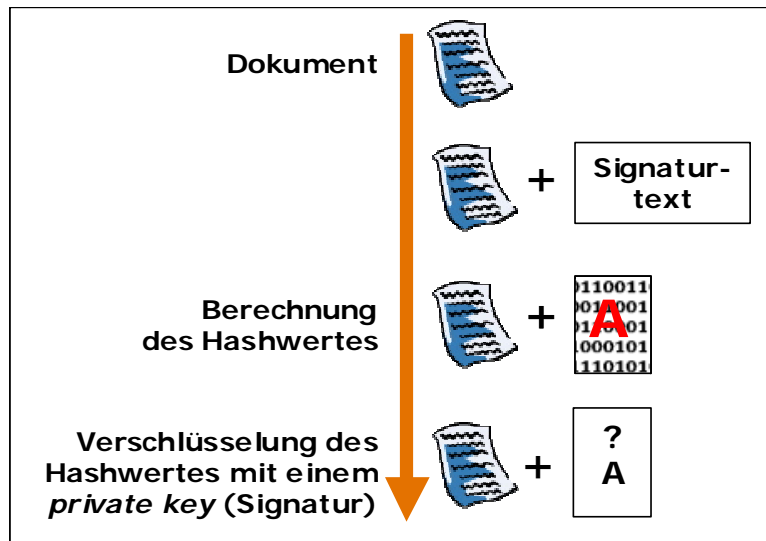


Abbildung 29: Erstellung der Elektronischen Signatur

Der Hash-Wert wird über das Dokument und den Signaturtext berechnet. Dieser Hash-Wert, d.h. eine Verknüpfung von Dokument und Signaturtext, wird dann signiert.

Eine Signatur wird durch das Dokument und den Signaturtyp bestimmt. Also wird nicht nur das Dokument signiert, sondern darüber hinaus auch noch der Signaturtext. Das zu signierende Paket besteht daher aus einem Signaturheader, einem Signaturtext und dem eigentlichen Dokument.

§ Im Signaturheader sind GUID, Kurzbezeichnung und Länge des Signaturtextes, ID des Anwenders/des Unterschreibenden, der Public Key des Anwenders, die zertifizierende Stelle des Systems (wer hat die Chipkarte ausgestellt), Datum, die Uhrzeit, die Stationsnummer, vollständiger Name des Anwenders/Unterschreibenden und optionale Felder wie Standort, erzeugendes System etc. enthalten.

§ Der Signaturtext enthält den in der Datenbank hinterlegten Text.

§ Danach wird das Dokument angehängt.

Über dieses Paket wird ein Hash gebildet, der dann mittels des Private Keys unterschrieben wird. Das ist dann die Signatur des Dokumentes.

Die Elektronische Signatur kann sowohl direkt im enaio® client, als auch eingebettet in den Workflow zur Autorisierung von Workflow-Vorgängen verwendet werden. Dabei wird automatisch am entsprechenden Schritt ein Dokument erzeugt, dass dann signiert und in der Workflowakte abgelegt wird. Der enaio® client erzeugt aus dem gewählten Dokument eine PDF-Datei. Diese PDF-Datei wird im Signaturmodul signiert und verifiziert.

Datenbanktabellen

Signaturtexte und deren Bezeichnung werden im enaio® administrator konfiguriert und in der Datenbank abgelegt. Signaturtexte und Signaturbezeichnungen werden in der Tabelle `ossigtext` abgelegt. Damit wird erreicht, dass die Elektronische Signatur schon mit einem vordefinierten Text aufgerufen werden kann. Das ist insbesondere in Workflowszenarien sinnvoll, da dem User hier vorgegeben werden kann, an welchen Schritten er welche Arten von Signaturen hinterlegen kann. Die Signatur zu einem Dokument wird mit allen informativen Daten in der Tabelle `osdigsig` der Datenbank gespeichert. Der Hash-Wert über ein Dokument bzw. über Dokument und Signatur wird in der Tabelle `osdochash` gespeichert.

Externes Speichern und Versenden der Elektronischen Signatur

Zum externen Speichern und Versenden von Signaturen werden diese mit dem Header und Signaturtext in einer Datei gespeichert.

Nun kann der jeweilige Benutzer anhand des Dokumentes und der Signaturdatei prüfen, ob die Signatur auch mit dem Benutzer usw. übereinstimmt. Dazu muss er wieder ein Signaturpaket zusammenbauen und dann anhand des öffentlichen Schlüssels des Benutzers prüfen, ob das Dokument und der Signaturtext mit der Signatur übereinstimmen.

Detaillierte Informationen zur Verwendung der Elektronischen Signatur können Sie den Handbüchern zu enaio® client und enaio® administrator entnehmen.

Sicherung der App OS|mobileDMS unter iOS

Mobile Anwendungen werden überwiegend unterwegs verwendet. Sie laufen auf kleinen, portablen Smartphones oder Tablet-Computern. Aufgrund der Größe und Portabilität sind sie jedoch leicht durch Dritte in einem unaufmerksamen Moment zu entwenden. Eine Entwendung hat oftmals nicht nur die Folge, dass das mobile Endgerät verloren ist, sondern auch die auf dem Gerät gespeicherten Daten. Der Wert sensibler persönlicher oder Unternehmensdaten ist oftmals beträchtlich größer als der Wert des eigentlichen mobilen Endgeräts. Aus diesem Grund müssen mobile Anwendungen, die mit sensiblen Daten arbeiten, auch bei Diebstahl des mobilen Endgeräts sicherstellen, dass diese nicht in die Hände Dritter fallen.

Neben dem materiellen Diebstahl des mobilen Endgeräts existiert auch der virtuelle Diebstahl von Daten durch böartige Apps, die aus Unkenntnis oder durch Dritte auf dem mobilen Endgerät installiert wurden. Solche Apps lesen oft sensible Daten, die auf dem mobilen Endgerät verarbeitet werden, aus und senden sie an den Hersteller der böartigen App.

Sicherheitskonzept von Apple iOS

Im Betriebssystem iOS aus dem Hause Apple werden alle Apps in separaten Sandboxes installiert und ausgeführt. Das Sandboxkonzept schirmt die Apps auf dem Betriebssystem voneinander ab. Einer App ist es somit nicht möglich auf Daten anderer Apps zuzugreifen. Dieses Sicherheitskonzept seitens des Betriebssystems beugt bereits dem virtuellen Datendiebstahl von Apps vor, ohne dass in den installierten Apps spezielle Vorkehrungen getroffen werden müssen.

Zusätzlich dazu sichert iOS den Zugriff auf Daten, indem es die Eingabe eines Passworts fordert. Der Anwender kann im Betriebssystem ein vierstelliges, numerisches Passwort hinterlegen, das bei Reaktivierung des mobilen Endgeräts abgefragt wird. Ohne das Passwort kann das Endgerät nicht verwendet werden. Darüber hinaus bietet das Betriebssystem eine Option an, dass alle Daten auf dem mobilen Endgerät gelöscht werden, wenn das Passwort zehnmal falsch eingegeben wurde. Diese beiden Sicherheitsfunktionen schützen bereits rudimentär vor der Entwendung sensibler Daten durch Dritte im Falle eines Diebstahls des mobilen Endgeräts. Sie setzen jedoch voraus, dass der Anwender die Sicherheitsfunktionen aktiviert hat und das Passwort einen hohen Komplexitätsgrad hat.

Viele Daten werden über Funknetze auf das mobile Endgerät übertragen. Hier unterstützt das Betriebssystem die gängigen Verschlüsselungsverfahren für Funktechnologien WEP, WPA (Enterprise), WPA2 (Enterprise) sowie VPN. Hierdurch können die Datenverbindungen zwischen dem mobilen Endgerät und den Accesspoints gesichert erfolgen. Bei einer Verbindung über das Mobilfunknetz werden die Daten mittels A5/3 verschlüsselt übertragen.

Sicherheitskonzept von OS|mobileDMS

Zusätzlich zum betriebssystemseitigen Sicherheitskonzept verfügt die App OS|mobileDMS von OPTIMAL SYSTEMS über weitere Sicherheitsfunktionen.

Um Daten über die App vom enaio®-Server abrufen zu können, muss sich die App an der REST-Schnittstelle enaio® appconnector anmelden. Die Authentifizierung erfolgt hierbei über Basic Authentication (RFC 2617), die den Benutzernamen und das Passwort als Base64-kodierte Zeichenketten an enaio® appconnector sendet.

Da bei dieser Art der Authentifizierung der Benutzername und das Passwort unverschlüsselt über das HTTP-Protokoll versendet werden, unterstützt OS|mobileDMS parallel zum HTTP-Protokoll auch das gesicherte HTTPS-Protokoll. Wenn das HTTPS-Protokoll für die Übertragung eingesetzt wird, so erfolgt die gesamte Datenkommunikation zwischen OS|mobileDMS und der REST-Schnittstelle enaio® appconnector SSL-verschlüsselt. Der Einsatz von SSL wird dringend empfohlen, da die Verschlüsselung der Datenverbindung über WIFI oder UMTS nur bis zum jeweiligen Accesspoint erfolgt und danach die Daten unverschlüsselt weitergeleitet werden. SSL gewährleistet eine vollständige Verschlüsselung... Infrastrukturelemente in dieser Verbindung können die Daten nur schwer entschlüsseln.

Die Zugangsdaten zur Authentifizierung des Anwenders am enaio®-Server werden innerhalb von OS|mobileDMS unverschlüsselt gespeichert, jedoch verhindert die Sandbox des Betriebssystems, dass andere Apps darauf zugreifen können. Weil das Passwort bei der Eingabe in OS|mobileDMS mit Punkten maskiert wird, kann es auch später nicht von Dritten in Klartext auf dem mobilen Endgerät eingesehen werden.

Bei erfolgreicher Authentifizierung von OS|mobileDMS an enaio® appconnector werden von der laufenden App fortan alle empfangenen Daten in einem HTTP-Cache zwischengespeichert. Die Daten werden ebenfalls innerhalb der Sandbox der App abgelegt und sind durch diese vor einem Zugriff Dritter geschützt. Hierbei wird innerhalb von OS|mobileDMS der AFCache verwendet. Der AFCache verwirft alle zwischengespeicherten Daten, sobald die App beendet wird.

Neben dem HTTP-Caching können Favoritendokumente auch offline zugänglich gemacht werden. Dies kann beispielsweise notwendig sein, wenn Dokumente auch ohne bestehende Internetverbindung bei einem Kundentermin verfügbar sein sollen. Offline verfügbare Favoritendokumente werden im App-Dokumentverzeichnis, das durch die Sandbox des Betriebssystems vor dem Zugriff anderer Apps gesichert ist, unverschlüsselt gespeichert. Die Indexdaten zu dem Dokument werden zurzeit nicht offline verfügbar gemacht, sofern sie sich nicht im AFCache befinden. Das Herunterladen und Speichern eines Favoritendokuments im Dateisystem erfolgt automatisch, sobald ein Dokument innerhalb der App als Favorit gekennzeichnet wird.

Das Dokument wird ebenfalls automatisch aus dem Dateisystem der App entfernt, sobald der Anwender das Dokument aus den Favoriten entfernt. Um neben den Favoritendokumenten auch die Liste der Favoriten offline verfügbar zu haben, wird sie ebenfalls im AFCache und im Dateisystem zwischengespeichert. Die Liste wird bei jedem HTTP-Abruf aktualisiert und ist unverschlüsselt im Dateisystem abgelegt.

Parallel zu den Favoriten werden als weiterer Datensatz sogenannte Droptargets im Dateisystem von OS|mobileDMS zwischengespeichert. Droptargets bilden die Grundlage für das Erfassen von Informationen über die App. In Droptargets sind Abläufe wie z. B. das Erfassen eines Kontakts oder eines Rückrufs hinterlegt. Aus den Ablaufinformationen werden dann dynamisch Formulare erstellt, die den Anwender durch den jeweiligen Ablauf führen. Wenn das mobile Endgerät mit einer Kamera ausgestattet ist, können über eine dann verfügbare Option Fotos

aufgenommen und direkt im ECM-System gespeichert werden. Über diese Option aufgenommen, werden Fotos nach der Aufnahme nicht in der Fotogalerie des Betriebssystems abgelegt, sondern, sofern im Droptarget entsprechend definiert, als Dokument in den Favoriten von OS|mobileDMS aufgenommen.

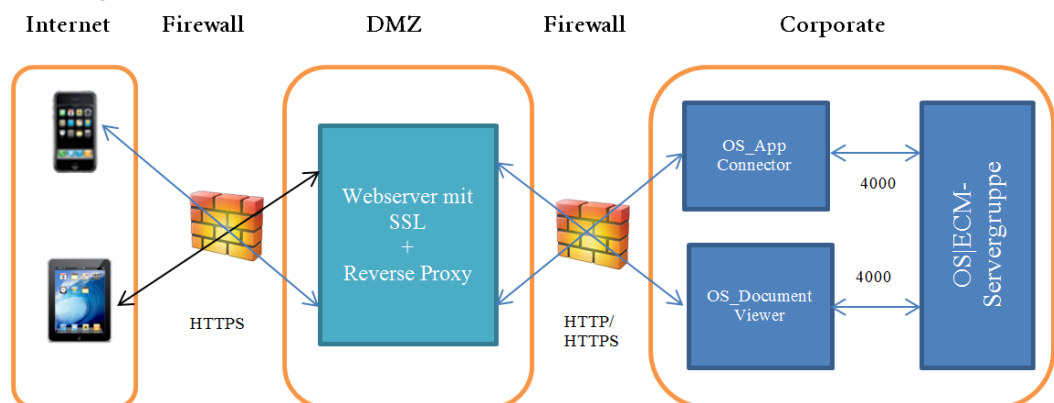
Damit die Daten, die Anwender in OS|mobileDMS verwalten, auch vor dem Zugriff Dritter geschützt sind, wenn das Betriebssystem nicht gesperrt ist, kann in der App eine weitere PIN eingerichtet werden. Diese vierstellige PIN wird bei jedem Start und jeder Aktivierung der App erneut abgefragt. Darüber hinaus kann in den App-Einstellungen konfiguriert werden, dass nach dreimaliger falscher Eingabe der PIN alle in OS|mobileDMS gespeicherten Serverprofile, also Verbindungsdaten zu enaio®-Servern inklusive Benutzername und Passwort, gelöscht und das Demoprofil (demo.ecm.mobi) der OPTIMAL SYSTEMS GmbH wieder aktiviert wird.

IT-Sicherheit für enaio® appconnector und enaio® documentviewer

OS|mobileDMS kommuniziert per HTTP-Anfrage mit enaio® appconnector und enaio® documentviewer. enaio® appconnector liefert alle Daten bis auf die Dokumente sowie Indexdaten der Dokumente zu OS|mobileDMS. enaio® documentviewer wird von der App zur Dokument- und Indexdatenanzeige aufgerufen und als WebView in OS|mobileDMS zur Anzeige eingebettet.

Um die Sicherheit der Daten durch enaio® appconnector sowie enaio® documentviewer zu gewährleisten, sollten System und Netzwerk gemäß den IT-Grundschutzkatalogen des Bundesamts für Sicherheit in der Informationstechnik aufgebaut und konfiguriert sein.

Die nachfolgende Grafik veranschaulicht den Zugriff von OS|mobileDMS auf die beiden genannten Kerndienste.



Die Grafik zeigt, wie OS|mobileDMS auf den mobilen Endgeräten mittels einer SSL-gesicherten HTTPS-Verbindung auf einen Webserver in der DMZ zugreift. Dieser verfügt über einen Reverse Proxy und übersetzt die Anfragen von OS|mobileDMS auf interne HTTPS-Anfragen an die Kerndienste enaio® appconnector und enaio® documentviewer. Die Kerndienste wiederum fragen die Rohdaten vom enaio®-Server ab, der in der Grafik als Servergruppe dargestellt ist.

Wenn das Corporate-Netzwerk aus mehreren verteilten Knoten besteht (z. B. Rechenzentrum Stuttgart, München, Hamburg usw.), sollten alle in der Grafik gezeigten Elemente des Corporate-Netzwerks in jedem Knoten existieren. Der Webserver mit Reverse Proxy würde dann je nach Anfrage-URL die Anfrage an einen speziellen Knoten übersetzen. Ähnlich kann in einem Mehrmandantensystem verfahren werden. Pro enaio®-Server-Mandant könnte ein Corporate-Knoten erstellt werden. Der DMZ-Webserver kann dann Anfrage-URL auf die einzelnen Mandanten übersetzen und an den richtigen Corporate-Knoten senden.

Archivierung

Backup

Die wichtigste enaio®-externe Maßnahme zur Vermeidung von Datenverlust ist die Dateisicherung. Die Dateisicherung ist keine Administrationstätigkeit im enaio®-System. Zu Sicherungsstrategien wird die einschlägige Literatur empfohlen. Folglich sollen nur einige wichtige Hinweise zur Sicherung gelistet werden.

Folgendes sollte systematisch gesichert werden:

§ Daten

Dafür sind die Dateien auf dem Datenbankserver zu sichern, die die enaio®-Datenbanktabellen enthalten. Die Sicherung der entsprechenden Dateien wird dringend empfohlen. Ohne eine regelmäßige Sicherung der entsprechenden Dateien, ist die Wiederherstellung eines beschädigten enaio®-Systems nicht möglich.

§ Dokumentdateien

enaio®-Dokumentdateien sind einerseits im WORK-Verzeichnis des Anwendungsservers bzw. der Anwendungsservergruppe, andererseits sind archivierte Dokumente auf Archivmedien gespeichert. Es wird dringend empfohlen, sowohl das oder die WORK-Verzeichnisse zu sichern, als auch die Medien zu spiegeln oder zu kopieren. Ohne Sicherung des WORK-Verzeichnisses und der Archivmedien ist die Wiederherstellung eines enaio®-Systems nicht möglich.

§ Programmverzeichnisse:

§ Server

Die Anwendungsverzeichnisse können gesichert werden. Dies ist aber nicht erforderlich. Die Neuinstallation von Anwendungs- und Fileservern ist mit der Setup-DVD und den Original-Lizenzdateien schnell durchführbar. Dafür sind aber dringend die Lizenzdateien zu sichern. Neben den Programmverzeichnissen sollte auch die Systemregistrierung des Servers gesichert werden.

§ Client

Die Sicherung von Clientinstallationen kann sinnvoll sein, sie ist aber nicht erforderlich, da Netzwerkinstallation an einzelnen Arbeitsstationen schnell durchgeführt sind. Wenn eine große Anzahl an Clients vorhanden ist, ist die Sicherung einzelner Client-Stationen sehr aufwändig.

Bei der Sicherung von Daten und Dokumentdateien ist darauf zu achten, dass die Sicherung zu einem Zeitpunkt und in einem Systemzustand durchgeführt wird. Damit stellen Sie sicher, dass keine Inkonsistenzen auftreten, z.B. vorhandene Indexdatensätze mit fehlenden Dokumentdateien. Führen Sie die Sicherung deshalb dann durch, wenn keine Produktivtätigkeit erfolgt.

Sind Daten, Dokumentdateien und Server-Programmverzeichnisse gesichert, so lässt sich durch Zurückspielen der Datenbank, des WORK-Verzeichnisses und der Programmverzeichnisse inklusive Systemregistrierung des Anwendungsservers sehr schnell eine Wiederherstellung des enaio®-Systems erreichen.

Ist ein enaio®-System z.B. aufgrund eines Festplattendefekts nicht mehr arbeitsfähig, gehen mit der Festplatte auch die Dokumentdateien verloren, die nach der letzten Sicherung erfasst wurden. Ist auf derselben Festplatte auch die Datenbank installiert, dann sind auch die Indexdaten seit der letzten Sicherung verloren. Bei einer systematischen Ablage der erfassten Papierbelege sollte aber eine Nacherfassung der verlorenen Dokumente leicht zu organisieren sein.

Grundlagen

Einleitung

Eine zentrale Funktion eines DMS ist die digitale Archivierung elektronischer Dokumente. Damit die digitalisierten Belege rechtlich als dokumentenechte Repräsentation der physischen Dokumente anerkannt werden, muss das Archivierungsverfahren einige gesetzliche Anforderungen erfüllen. Zu diesen Anforderungen gehört u.a. die unveränderliche Speicherung der Dokumente auf geeigneten Medien, z.B. WORM-Medien (Write Once Read Many).

Archivierung bedeutet in enaio® die Übertragung erfasster digitaler Dokumente vom Festplattenspeicher beim Anwendungsserver auf ein Archivmedium, z.B. eine WORM. Dokumente werden nur dann archiviert, wenn der zuständige Sachbearbeiter dem Dokument explizit den Status 'Archivierbar' gibt. Die Archivierung wird in enaio® als Automatische Aktion konfiguriert und als eigenständiger Batch-Prozess aus dem enaio® administrator oder enaio® start durchgeführt. Die Archivierung besteht darin, dass die Dokumente auf ein anderes Medium überspielt und aus dem WORK-Bereich entfernt werden.

Für eine dokumentenechte Archivierung können ebenfalls externe Systeme angeschlossen werden. Zur Verfügung stehen beispielsweise EMC-Centera, iTernity und iXOS. Diesen Systemen werden dann die Dokumente zur eigenständigen Verwaltung übergeben.

Eine Archivierung ist auch möglich, wenn die Archivmedien nicht den oben besprochenen Anforderungen entsprechen. Dann werden die Dokumente zwar nicht manipulationssicher abgelegt, alle anderen Prozesse sind aber gleich. Ist der Anwender nicht durch gesetzliche Bestimmungen oder andere Motive gezwungen, eine dokumentenechte Archivierung durchzuführen, ist es trotzdem sinnvoll, regelmäßig bestimmte Dokumente aus dem WORK-Bereich auf externe Medien auszulagern, also eine Archivierung durchzuführen. Diese kann im einfachsten Fall in Verzeichnisse auf speziell dafür bereitgestellte Festplatten erfolgen.

Um eine den rechtlichen Anforderungen entsprechende Archivierung zu gewährleisten, ist das Zusammenspiel von allen relevanten Faktoren unverzichtbar. Dazu gehören die anzuwendenden Verfahren ebenso wie die eingesetzte Hardware oder diverse Softwarekomponenten.

Alle in Frage kommenden Aspekte sollten empfehlenswerter Weise in einer vom haftenden Betreiber zu erstellenden Verfahrensdokumentation geregelt und dokumentiert werden. Mitarbeiter von OPTIMAL SYSTEMS können diesbezüglich gerne auf Anfrage im Projekt behilflich sein.

Bei allen von OPTIMAL SYSTEMS zertifizierten Compliance-Speicherlösungen unterstützt enaio® eine den rechtlichen Anforderungen entsprechende

Archivierung. Das ordnungsgemäße Archivieren von Dokumenten, insbesondere auch das Setzen der physikalischen Retentionszeiten sind dabei naturgemäß vom jeweiligen Speichersystem abhängig. Kann beispielsweise ein solcher Compliance-Speicher Retentionszeiten nur bis zum Jahr 2038 verwalten, so wird diese Limitierung auch durch die Archivierungssoftware, z. B. enaio®, nicht aufgehoben werden können.

enaio® stellt Werkzeuge und Mittel für verschiedene zertifizierte Archivspeichersysteme zur Verfügung, um eine einfache Konfiguration und einen sicheren Betrieb zu gewährleisten. Es sind jedoch stets die in den Dokumentationen zu den jeweiligen Schnittstellen genannten Einstellungen vorzunehmen.

Es wird deshalb empfohlen, unter anderem die Planung der Aufbewahrungsfristen, die korrekte Auswahl des Archivspeichersystems und dessen Konfiguration, insbesondere die Konfiguration der Retentionszeiten, die entsprechenden Einstellungen für die Archivspeichersysteme in enaio® sowie die richtige Konfiguration und Ausführung der Archivierungsläufe in enaio® im Projekt mit den Beratern abzustimmen, durchzuführen, zu dokumentieren und zu testen.

Ablaufschema der Arbeit mit archivierten und nichtarchivierten Dokumenten

enaio® ist ein LAN-orientiertes DMS mit Client-Server-Architektur. Die folgende Grafik stellt die Architektur von enaio® ganz allgemein dar. Ein Anwendungsserver bedient mehrere Client-Anwendungen (enaio® client, enaio® editor, enaio® administrator und enaio® capture) und steuert die Archivierung.

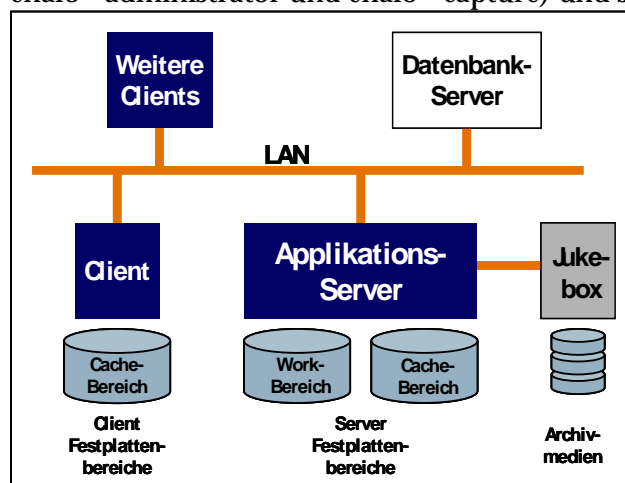


Abbildung 30: enaio Server-/Clientarchitektur

Vier Vorgänge sollen grafisch erläutert werden: Ein neues Dokument anlegen, ein bestehendes Dokument öffnen, das Archivieren von Dokumenten und das Öffnen archivierter Dokumente. Es werden nur Dateioperationen schematisch gezeigt, Datenbankvorgänge werden nicht berücksichtigt:

Anlegen eines neuen Dokuments:

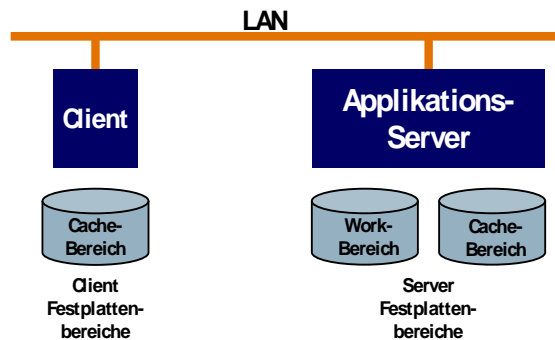


Abbildung 31: Anlegen eines neuen Dokuments

1. Im Client wird ein neues Dokument angelegt. Die dazugehörige Datei wird im CACHE-Bereich des Clients gespeichert.
2. Wenn der Sachbearbeiter das Dokument schließt, wird die Datei zum WORK-Bereich des Anwendungsservers übertragen und im CACHE-Bereich des Clients gelöscht. Dieser Vorgang wird 'Einchecken' genannt.

Dokument zum Bearbeiten öffnen:

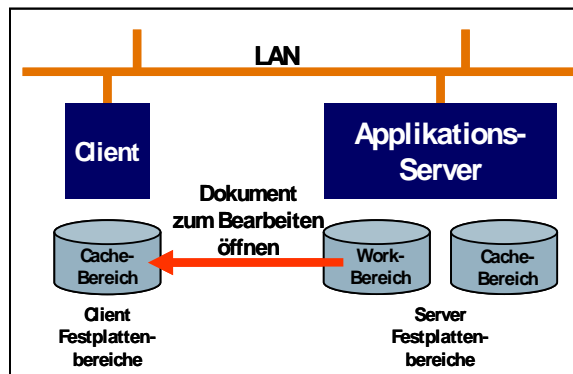


Abbildung 32: Ein Dokument zum Bearbeiten öffnen

Hier wird die erforderliche Datei vom WORK-Bereich des Anwendungsservers zum CACHE-Bereich des Clients übertragen. Die Datei wird 'ausgecheckt' und steht anderen Bearbeitern nur schreibgeschützt zur Verfügung.

Archivierung eines Dokuments:

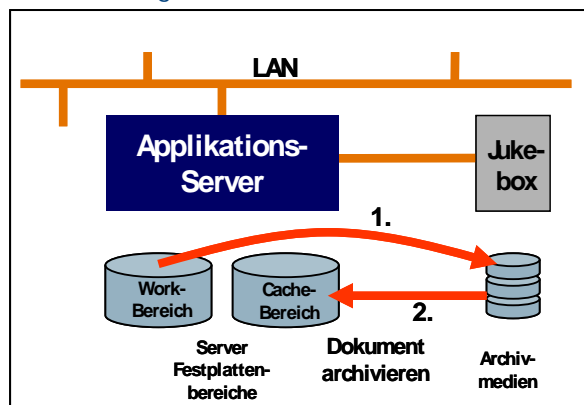


Abbildung 33: Archivierung eines Dokuments

1. Ein bestehendes Dokument wird vom WORK-Bereich des Anwendungsservers auf ein Medium kopiert und im WORK-Bereich gelöscht.
2. Danach wird eine Dateikopie im CACHE-Bereich des Anwendungsservers angelegt.

Archiviertes Dokument öffnen:

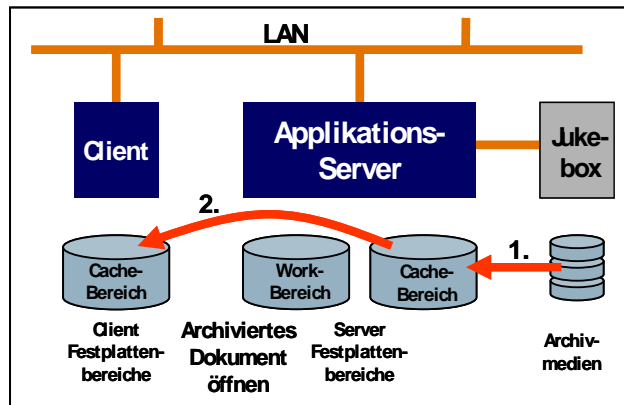


Abbildung 34: Öffnen eines Dokuments

1. Ist das Dokument im CACHE-Bereich des Anwendungsservers vorhanden, wird es zum CACHE-Bereich des Clients schreibgeschützt kopiert.
2. Wenn keine Kopie der Datei im CACHE-Bereich des Anwendungsservers liegt, wird sie zuvor vom Archivmedium in den CACHE-Bereich des Anwendungsservers kopiert und dann schreibgeschützt zum Client-CACHE-Bereich kopiert.

Medien, Medienset und Spiegelmedien

Die dokumentenechte Archivierung erfolgt auf WORM-Medien in sogenannten Jukeboxen. Ähnlich den Jukeboxen der Unterhaltungsindustrie, enthält eine Jukebox für die Archivierung ein oder mehrere Abspielgeräte (Laufwerke) und viele Slots, in denen sich die eigentlichen Speichermedien befinden. enaio® unterstützt nur Jukeboxen, die über die Software InveStore der Firma Pegasus angesprochen und über das Dateisystem des Betriebssystems eingegliedert werden können. Bei der Einrichtung der Jukebox unter InveStore und der Beschickung mit den Speichermedien, wird die Jukebox unabhängig von der Anzahl physisch vorhandener Laufwerke nur über einen Laufwerksbuchstaben im Dateisystem angezeigt.

Die Speichermedien in der Jukebox erscheinen als Verzeichnisse dieses Laufwerkes. Daraus ergeben sich die bei der Archivierung gebräuchlichen Begriffe:

- § Medien sind die in einer Jukebox enthaltenen Datenträger (WORM-Medien), wobei bei doppelseitigen Datenträgern jede Seite als Medium betrachtet wird. Jedes Medium wird unter seinem Formatierungsnamen als Verzeichnis angezeigt.
- § Spiegelmedien sind Medien, die zu Erhöhung der Datensicherung zur Dopplung der Archivierung den Originalmedien zugeordnet werden. Ein Spiegelmedium ist eine genaue Kopie des Originalmediums. Man erhält somit zwei Sicherungskopien seiner archivierten Dokumente.
- § Mediensets sind Zusammenfassungen von mehreren Verzeichnissen (Medien). Sie dienen der Strukturierung der Archivierung und erleichtern die Verwaltung der Archive. Sie enthalten beliebige, vom Benutzer anzugebende Medien und können zur Datensicherheit auch Spiegelmedien umfassen.

In der folgenden Abbildung wird ein Beispiel für die Struktur eines Archivs mit Medien (medium1 bis medium4) und Spiegelmedien (smedium1 bis smedium4) dargestellt:

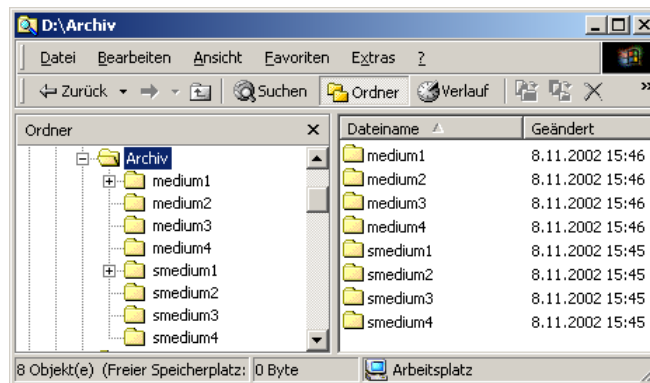


Abbildung 35: Beispiel einer Archivstruktur mit Medien und Spiegelmedien

Mit den Konfigurationstools, die enaio® für die Einrichtung der Archivierung anbietet, können nun beliebig Mediensets (Zusammenstellungen von Medien) gebildet werden. Die Einrichtung von Mediensets, Medien und Spiegelmedien wird im folgenden Kapitel beschrieben.

Die Einrichtung der Archivierung wird über enaio® enterprise-manager vorgenommen und detailliert in der Dokumentation 'Automatische Aktion Archivieren' beschrieben.

Archivierung und Zugriff auf archivierte Dokumente

Archivierungskonzept

Bei der Archivierung wird davon ausgegangen, dass die Archivmedien nur einmal beschreibbar (ähnlich wie CD in Multisessions) sind. Dateien können weder gelöscht noch für eine Änderung geöffnet werden. Es muss bei der Archivierung sichergestellt werden, dass nicht nur das Dokument selbst, sondern auch alle Informationen zum Dokument auf diesem Medium Platz finden. Deshalb ist der Algorithmus für die Speicherung von Daten recht kompliziert. enaio® bietet zwei Methoden für die Bestimmung des Speicherplatzes auf einem Archivmedium an, wenn das Medium sich auf einer Jukebox befindet und per Pegasus-Software angesteuert wird. In der Registrierung kann unter `HKLM\SOFTWARE\OPTIMAL SYSTEMS\ 'AppServer'\Schemata\4.0\Archive` der Eintrag `MediaSizeMethod` erfolgen. Dieser ist nur sinnvoll für Jukeboxen mit einer Ansteuerung über Pegasus InveStore und kann drei Werte besitzen:

- 0 Bestimmung über das Dateisystem (default)
- 1 Bestimmung über Pegasus-Software (Inhalt von '!\FSFREE.###')
- 2 Bestimmung über Pegasus-Software (Größe von 'FSFREE___.###')

Die Methode über das Dateisystem ist schneller, während die Bestimmung über die Pegasusmethode genauer ist. Die Verwendung der ersten Methode ist kein größeres Risiko für einen Datenverlust, sondern hat zur Folge, dass möglicherweise nicht so viele Daten auf dem Medium Platz finden. Für Medien auf Festplatten ist dieser Parameter gegenstandslos.

Bei der Konfigurierung der Automatischen Aktion 'Archivierung' werden in einem Konfigurationsdialog alle Dokumententypen angezeigt, für die eine Verknüpfung

mit Mediensets in der Datenbanktabelle `medrel` (vgl. Abbildung 40) erfolgte. Dem Anwender wird außerdem gezeigt, wie viele Dokumente je Objekttyp als archivierbar gekennzeichnet sind. Bei der Konfiguration können nun die Dokumententypen ausgewählt werden, die archiviert werden können.

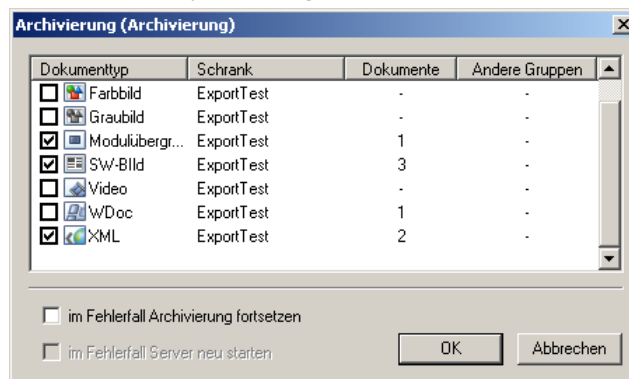


Abbildung 36: Auswahl der zu archivierenden Dokumententypen

Für jedes Medium ist die Mediengröße, die Clustergröße und die Speichergröße, die auf dem Medium mindestens freigehalten werden soll, festgelegt. Aus diesen Werten und dem schon verbrauchten Speicherplatz kann der noch zum Archivieren freie Speicherplatz ermittelt werden.

Danach wird die Anzahl zu archivierender Dateien eines Objekttyps, deren Größe und die Größe der Recovery-Dateien `stamm#.xxx`, `regis#.xxx`, `object#.xxx`, die die Indexinformationen enthalten, ermittelt. Finden diese Dateien alle Platz auf dem Medium, so werden sie dorthin kopiert. Es werden immer nur komplette Dokumente auf einem Medium archiviert. Besteht ein Dokument aus mehreren Dateien (Seiten), so müssen alle Dateien (Seiten) des Dokumentes auf dem Medium Platz finden. Ansonsten wird dieses Dokument auf dem nächsten zur Verfügung stehenden Medium archiviert. Die Archivierung erfolgt inkrementell. Es werden alle Dokumente des aktuellen Dokumententyps, die auf dem Medium Platz finden (bzw. die zu archivieren waren) auf das Archivmedium kopiert. Dabei wird die folgende Dateistruktur erzeugt:

```
Pfad\\Archivname\\Mediumname\\Haupttyp\\Untertyp\\Dokumententyp\\DocID(mod 0x100)\\DocID.Seitenzahl
```

oder

```
Pfad\\Archivname\\Mediumname\\Haupttyp\\Untertyp\\Dokumententyp\\DocID(mod 0x100)\\DocID.ext
```

Das Unterverzeichnis 'Dokumententyp' kann nur zwei Werte annehmen:

§ '02' für das Dia des Dokuments und

§ '03' für das eigentliche Dokument.

Für W-, M-, E- und XML-Dokumente ist 'ext' durch die konkrete Endung des Dokuments zu ersetzen.

Auf jedem Medium, das für die Archivierung verwendet wird, legt das Archivierungsprogramm eine Datei `as_label.dat` ab, in die der Name des Mediums eingetragen wird, in unserem Beispiel also `MEDIUM1`.

Außerdem wird auf dem Medium das Verzeichnis `sys` angelegt, in dem bei jedem Archivierungsvorgang die Recovery-Dateien mit den Indexdaten sowie die zum Zeitpunkt der Archivierung gültige Objektdefinition abgelegt werden. Die Recovery-Dateien sind reine ASCII-Textdateien, in denen die Verschlagwortungen aller Ordner, Register und Dokumente jedes archivierten Dokumentes eingetragen werden. Es werden also die Dateien `stamm#.xxx`, `regis#.xxx`, `object#.xxx` und

`sdrel.xxx` erzeugt (Die dreistellige Zahl `xxx` hängt von dem inkrementellem Status ab.). Diese Dateien haben einen Miniheader:

- § 1. Zeile: Versionsinformationen
- § 2. Zeile: Vorhandene Felder (Felder durch Tabs getrennt)
- § 3. Zeile bis n. Zeile: Daten (Felder durch Tabs getrennt)

Beispiel für eine Datei mit Recovery-Daten (hier für zwei Ordner im Schrank 1)

```
stamm1,Dokumente, optimal_AS Version 3.x * optimal systems GmbH #id,Ordner
id          feld1
19          Dokumente in der Hauptgruppe
45          Dokumente in der Nebenstelle 1
```

Datei `stamm1.006`

Beim Schreiben der Indexdaten ist darauf zu achten, dass Schrank- und Registerdaten nicht redundant vorliegen.

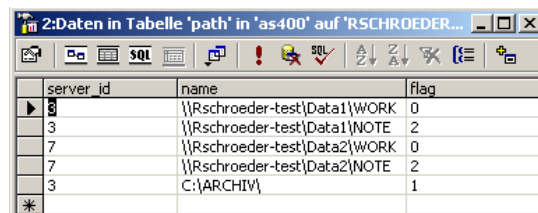
Im Verzeichnis `ARCHIVE`, das parallel zum Verzeichnis `WORK` im Datenverzeichnis erzeugt wird, wird für jedes Medium ein Verzeichnis mit dem Namen des Mediums angelegt. Im Unterverzeichnis `_SYS` werden die Recovery-Dateien des letzten Archivierungsdurchlaufes abgelegt. Außerdem wird hier im Medienverzeichnis eine Datei `recover.ini` mit folgender Syntax angelegt, in der die Nummer des letzten Archivdurchlaufes enthalten ist.

```
[SYS]
LNR=6
```

Datei `recover.ini` aus dem Verzeichnis `ARCHIVE`

Folgenden Datenbanktabellen stellen Informationen über die Archivierung zur Verfügung:

1. Datenbanktabelle `path`:



| server_id | name | flag |
|-----------|------------------------------|------|
| 3 | \\Rschroeder-test\Data1\WORK | 0 |
| 3 | \\Rschroeder-test\Data1\NOTE | 2 |
| 7 | \\Rschroeder-test\Data2\WORK | 0 |
| 7 | \\Rschroeder-test\Data2\NOTE | 2 |
| 3 | C:\ARCHIV\ | 1 |

Abbildung 37: Tabelle `path` für die im System bekannten Pfade

In der Tabelle `path` werden die für jede Servergruppe definierten Pfade eingetragen. Das Feld `server_id` bedeutet hier `servergruppen_id`. Die Flags haben folgende Bedeutung:

- § 0 – Pfad zum Server-WORK,
- § 1 – Archivpfad,
- § 2 – Pfad zu Notizen

2. Datenbanktabelle `medien`:

| id | name | server_id | set_id | state | medorder | medsize | written | mirror_id |
|----|----------|-----------|--------|-------|----------|---------|---------|-----------|
| 4 | WORK | 3 | <NULL> | 3 | <NULL> | <NULL> | <NULL> | <NULL> |
| 5 | NOTE | 3 | <NULL> | 4 | <NULL> | <NULL> | <NULL> | <NULL> |
| 8 | WORK | 7 | <NULL> | 3 | <NULL> | <NULL> | <NULL> | <NULL> |
| 9 | NOTE | 7 | <NULL> | 4 | <NULL> | <NULL> | <NULL> | <NULL> |
| 77 | MEDIUM1 | 3 | 76 | 0 | 1 | 500 | 1 | 82 |
| 78 | MEDIUM2 | 3 | 76 | 0 | 2 | 500 | 0 | 84 |
| 79 | MEDIUM3 | 3 | 76 | 0 | 3 | 500 | 0 | 86 |
| 89 | MEDIUM4 | 3 | 76 | 0 | 4 | 500 | 0 | 88 |
| 82 | SMEDIUM1 | 0 | 0 | 1 | 0 | 500 | 1 | <NULL> |
| 84 | SMEDIUM2 | 0 | 0 | 1 | 0 | 500 | 0 | <NULL> |
| 86 | SMEDIUM3 | 0 | 0 | 1 | 0 | 500 | 0 | <NULL> |
| 88 | SMEDIUM4 | 0 | 0 | 1 | 0 | 500 | 0 | <NULL> |

Abbildung 38: Datenbanktabelle *medien* für die Zuordnung von Medien zu Mediensets

Jedem Medium der Tabelle wird eine eindeutige ID aus der Tabelle *osnextindex* zugeordnet. Die Tabelle *medien* erfüllt gleich mehrere Aufgaben:

§ über das Feld *state* wird mitgeteilt, um welche Art eines Mediums es sich handelt

- 0 frei
- 1 gesperrt
- 2 ohne Bedeutung
- 3 WORK-Bereich
- 4 NOTE-Bereich

§ *server_id* bedeutet wieder *servergruppen_id*,

§ über *set_id* werden die eingetragenen Medien den gewünschten Mediensets zugeordnet,

§ über *mirror_id* werden den Archivmedien Spiegelmedien zugeordnet.

3. Datenbanktabelle *sets*:

| id | name | systemid | alias |
|----|------|----------|-------|
| 76 | SET1 | 75 | |

Abbildung 39: Datenbanktabelle *sets* für die Mediensets

In dieser Tabelle werden alle Mediensets eingetragen, eine systemweit eindeutige ID und ein Name für das Medienset eingetragen. Die ID eines Sets wird in der Tabelle *medien* als *set_id* eingetragen und erlaubt somit die Zuordnung eines Archivmediums zu einem Medienset.

4. Datenbanktabelle *medrel*:

| id | type | state | set_id |
|--------|------|--------|--------|
| <NULL> | 48 | <NULL> | 76 |
| <NULL> | 49 | <NULL> | 76 |
| <NULL> | 43 | <NULL> | 76 |
| <NULL> | 44 | <NULL> | 76 |
| <NULL> | 58 | <NULL> | 76 |
| <NULL> | 59 | <NULL> | 76 |
| <NULL> | 23 | <NULL> | 76 |
| <NULL> | 24 | <NULL> | 76 |
| <NULL> | 63 | <NULL> | 76 |
| <NULL> | 64 | <NULL> | 76 |
| <NULL> | 28 | <NULL> | 76 |
| <NULL> | 29 | <NULL> | 76 |
| <NULL> | 34 | <NULL> | 76 |
| <NULL> | 38 | <NULL> | 76 |
| <NULL> | 39 | <NULL> | 76 |
| <NULL> | 18 | <NULL> | 76 |
| <NULL> | 19 | <NULL> | 76 |

Abbildung 40: Datenbanktabelle *medrel* zur Zuordnung Mediensets zu Dokumenttypen

Werden bei der Konfiguration eines Mediensets diesem Objekttypen aus der Objektdefinition zugewiesen, so werden diese Typen hier eingetragen. Das Feld `type` beinhaltet die gewohnten Haupt- und Nebentypen, nur dass mit einer speziellen Konvertierung diese Typen in 2 Byte große Variablen umgewandelt werden.

Die Archivierung erfolgt über die Automatische Aktion 'Archivierung' (`axacarch.dll`), die im enaio® administrator eingebunden, konfiguriert und auch gestartet werden kann. Außerdem ist es möglich, die Archivierung über enaio® start zu bestimmten Zeitpunkten, periodisch usw. ausführen zu lassen. Die Beschreibung der Automatischen Aktion 'Archivierung' kann der entsprechenden Technischen Information zu Automatischen Aktionen entnommen werden.

Bei jedem Archivierungslauf werden automatisch die Objektdefinitionsdaten aus Gründen der Revisionssicherheit mit auf die Archivmedien geschrieben. Werden bei einem Archivierungslauf jeweils nur wenige Dokumente archiviert, zugleich aber umfangreiche Objektdefinitionsdaten, kann es sinnvoll sein, die Objektdefinition nicht mit auf die Archivmedien zu schreiben, sondern durch Verfahrensdokumentationen die Revisionssicherheit sicherzustellen. Deshalb kann über enaio® enterprise-manager das Archivieren der Objektdefinitionsdaten abgeschaltet werden.

Bei erfolgreicher Archivierung wird das entsprechende Dokument vom WORK-Bereich in den CACHE-Bereich (siehe Abbildung 41) des Servers verschoben, d.h. im Cache angelegt und im WORK gelöscht. Das CACHE-Verzeichnis besitzt genau die gleiche Struktur wie das Dateisystem im WORK-Bereich, so dass der Zugriff auf diese Dokumente nach dem gleichen Algorithmus ablaufen kann, wie wenn das Dokument im WORK-Bereich liegen würde. Da nach einer Archivierung die Dateien nicht mehr verändert werden können, ermöglicht die Arbeit mit einem CACHE-Bereich einen schnelleren Datenzugriff. Fordert ein Client ein archiviertes Dokument vom Server, so wird zuerst im CACHE-Bereich nachgesehen. Wird das Dokument dort gefunden, dann wird es aus dem CACHE-Bereich dem Client zur Verfügung gestellt. Da der Zugriff auf eine Festplatte im Allgemeinen sehr viel schneller ist, als der Zugriff auf ein Archivmedium, kann somit das Dokument schneller bereitgestellt werden.

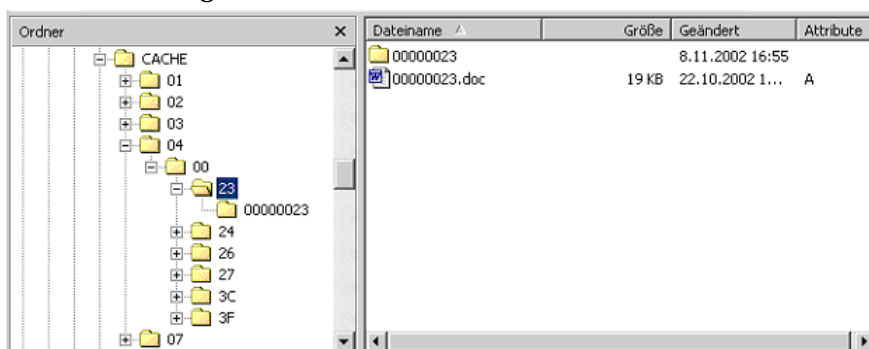


Abbildung 41: Beispiel für ein Server-CACHE- Verzeichnis nach einer Archivierung

Zur Verwaltung der Eigenschaften des CACHE-Bereichs stellt der Anwendungsserver periodische Aktionen zur Verfügung, durch die ein Überlauf bzw. Veralten des Dokumentenbestandes verhindert werden sollen. Dazu kann in der Registrierung für die konkrete Instanz des Anwendungsservers im Unterverzeichnis BATCHES folgendes festgelegt werden:

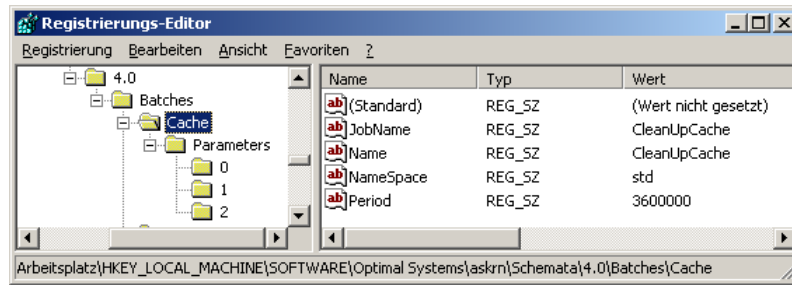


Abbildung 42: Einstellungen für den CACHE-Bereich

Es wird ein Serverjob 'CleanUpCache' des Namensraumes (Exekutors) 'std' alle 60 Minuten (3600000 ms entsprechen 60 Minuten) aufgerufen, um den Cache zu überprüfen. Die Überprüfung wird durch die angegebenen Parameter bestimmt:

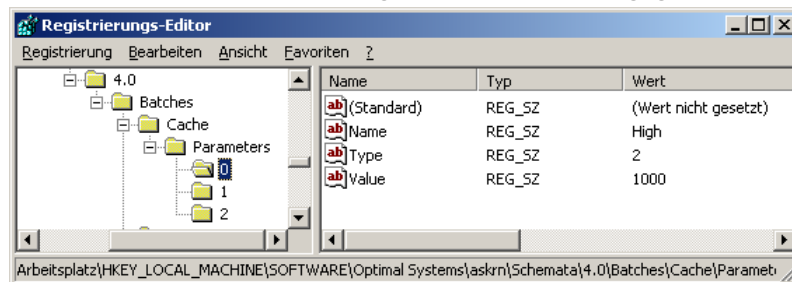


Abbildung 43: High-Marke des Füllstandes des CACHES

Der Wert 'High' bedeutet die maximale Größe (in MB), wobei Typ=2 besagt, dass es sich um einen Integerwert handelt.

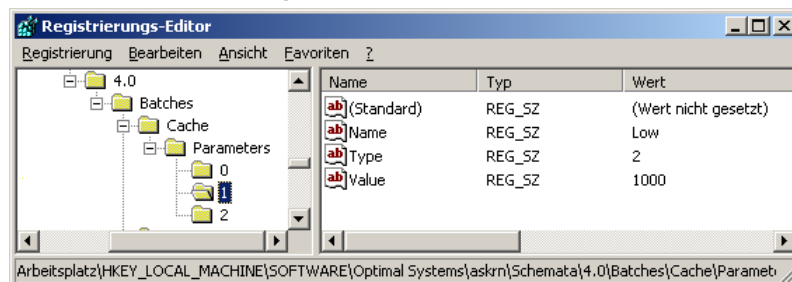


Abbildung 44: Low-Marke des Füllstandes des CACHES

Der Wert 'Low' bedeutet die minimale Größe (in MB), wobei Typ=2 besagt, dass es sich um einen Integerwert handelt.

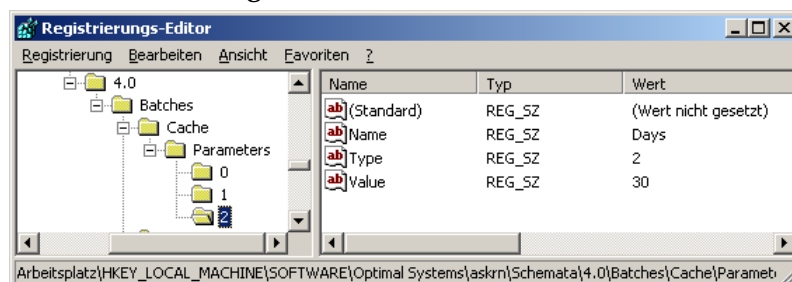


Abbildung 45: Verweildauer der Dokumente im CACHE

Der Wert 'Days' (in Tagen) bedeutet, dass alle Dateien zu entfernen sind, die sich, ohne gelesen zu werden, länger als diese Anzahl von Tagen im CACHE befinden. Entfernt werden sie aber nur dann, wenn diese obere Marke überschritten wurde. Dateien werden solange entfernt, bis entweder die untere Marke erreicht ist oder keine Dateien mehr vorhanden sind, die älter sind als angegeben. Vorrang hat dabei das Alter der Dateien.

Die CACHE-Bereinigung erfolgt nach folgendem Algorithmus. Es wird periodisch überprüft, ob die maximale Größe des CACHES überschritten wird. Wenn ja, so werden die Verzeichnisse im CACHE sukzessive nach Dateien durchsucht, die das Kriterium erfüllen. Wird eine solche Datei gefunden, so wird sie gelöscht. Ist damit die minimale CACHE-Größe erreicht, wird die Prozedur abgebrochen, wenn nicht, wird Verzeichnis für Verzeichnis durchsucht. Werden keine Dateien gefunden, die das Löschkriterium erfüllen, werden keine Dateien gelöscht und die CACHE-Größe wird zeitweise überschritten. Deshalb sollte diese Größe sinnvoll gewählt werden und auf der Festplatte freier Speicherplatz verbleiben, damit das letztgenannte Szenarium nicht zu Problemen mit dem Dateisystem führt.

Zur Verbesserung der Datensicherheit beim Archivieren, kann im enaio® administrator bei der Konfiguration der Archivierung eingestellt werden, dass von allen archivierten Dateien eine Kopie im Verzeichnis `BACKUP` angelegt werden soll (vgl. Abbildung 46). Dieses Verzeichnis besitzt genau die gleiche Struktur, wie das Dateisystem auf dem Archivmedium.

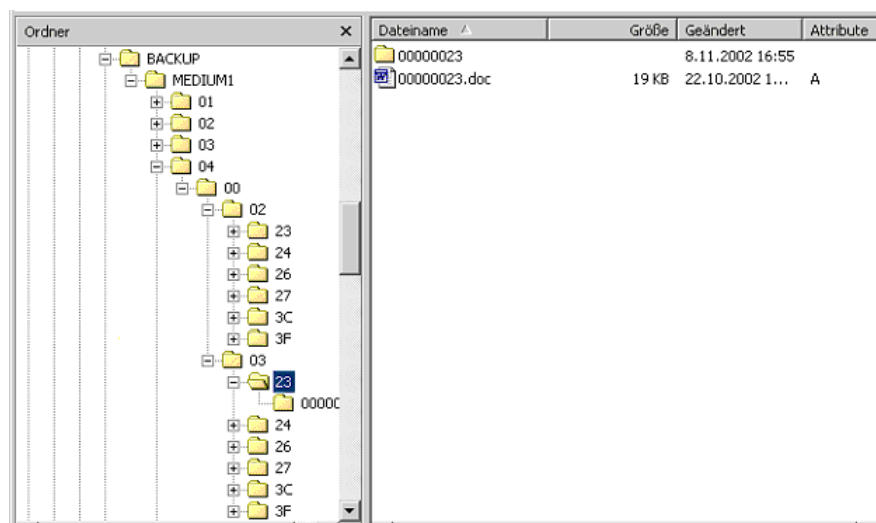


Abbildung 46: Backupverzeichnis einer Archivierung mit erhöhter Datensicherheit

Protokollierung

Während der Archivierung wird ein Protokoll geführt, in dem die Archivierung aller Dokumententypen und Hinweise auf Fehler eingetragen wird. Diese Datei erhält den Namen `osJJMMTT.rep` (JJ- zweistellige Jahreszahl, MM- zweistellige Monatszahl, TT- zweistellige Tageszahl). Bei aufeinanderfolgenden Archivierungen mit selben Datum wird die Datei fortgeschrieben. Der Report enthält folgende Informationen:

- § Datum und Uhrzeit des Reports
- § welcher Dokumenttyp archiviert werden sollte,
- § wann die Aktion gestartet wurde,
- § in welcher Servergruppe die Dokumente gefunden wurden,
- § durch welchen Server der Gruppe die Archivierung durchgeführt wurde,
- § wie viele Dokumente als archivierbar gefunden wurden,
- § ob Registerinformationen gefunden wurden,
- § die ermittelten Medien und Spiegelmedien,
- § Fehler,
- § Bestimmung des freien Speicherplatzes auf den Medien,

- § Resultat, wie viele Dokumente archiviert wurden und
- § eine Ergebniszusammenfassung.

Diese Information wird für jeden archivierten Dokumententyp angegeben. Der Report wird bei jeder Archivierung erstellt. Außerdem ist es möglich, noch weitere detailliertere Informationen zur Archivierung zu erhalten. Dazu ist im enaio® enterprise-manager die Option 'Bestätigte Archivierung' auszuwählen. Dann werden im gleichen Verzeichnis parallel zum Archivreport die Dateien `osTTMMJJ.ext` angelegt. 'ext' steht für Dateierweiterung und wird nach folgendem Schema gebildet:

- § Zeichen 1 = Haupttyp des Dokuments, z.B. 1 für X-Dokumente
- § Zeichen 2,3 = Untertyp des Dokuments, z.B. 00 für das erste Auftreten von X-Dokumenten in der Objektdefinition

Folgende Dokumententypen sind somit möglich:

- § `osTTMMJJ.100` bis `osTTMMJJ.1FF` für X-Dokumente
- § `osTTMMJJ.200` bis `osTTMMJJ.2FF` für D-Dokumente
- § `osTTMMJJ.300` bis `osTTMMJJ.3FF` für P-Dokumente
- § `osTTMMJJ.400` bis `osTTMMJJ.4FF` für W-Dokumente
- § `osTTMMJJ.500` bis `osTTMMJJ.5FF` für Film-Dokumente
- § `osTTMMJJ.600` bis `osTTMMJJ.6FF` für E-Mail-Dokumente
- § `osTTMMJJ.700` bis `osTTMMJJ.7FF` für XML-Dokumente

Unten werden die zur oben dargestellten Archivierung gehörenden ausführlichen Protokolldateien angeführt:

| Ordner | Bezeichnung | Medium | Pfad |
|--------------------------------------------|-------------|--------------------------|------|
| Dokumente in der Hauptgruppe X-Dok 1 an HG | MEDIUM1 | 01\00\03\1D\0000001d.000 | |
| Dokumente in der Hauptgruppe X-Dok 2 an HG | MEDIUM1 | 01\00\03\1E\0000001e.000 | |
| Dokumente in der Hauptgruppe X-Dok 3 an HG | MEDIUM1 | 01\00\03\1F\0000001f.000 | |

Datei `os081102.100` für die Archivierung von X-Dokumenten

| Ordner | Bezeichnung | Medium | Pfad |
|----------------------------------------------|-------------|--------------------------|------|
| Dokumente in der Nebenstelle 1 D-Dok 1 an NS | MEDIUM1 | 02\00\03\33\00000033.000 | |
| Dokumente in der Nebenstelle 1 D-Dok 2 an NS | MEDIUM1 | 02\00\03\34\00000034.000 | |
| Dokumente in der Nebenstelle 1 D-Dok 3 an NS | MEDIUM1 | 02\00\03\35\00000035.000 | |

Datei `os081102.200` für die Archivierung von D-Dokumenten

| Ordner | Bezeichnung | Medium | Pfad |
|--------------------------------------------|-------------|--------------------------|------|
| Dokumente in der Hauptgruppe P-Dok 1 an HG | MEDIUM1 | 03\00\03\20\00000020.000 | |
| Dokumente in der Hauptgruppe P-Dok 2 an HG | MEDIUM1 | 03\00\03\21\00000021.000 | |
| Dokumente in der Hauptgruppe P-Dok 3 an HG | MEDIUM1 | 03\00\03\22\00000022.000 | |

Datei `os081102.300` für die Archivierung von P-Dokumenten

| Ordner | Bezeichnung | Medium | Pfad |
|----------------------------------------------|-------------|--------------------------|------|
| Dokumente in der Hauptgruppe Brief 1 an HG | MEDIUM1 | 04\00\03\23\00000023.doc | |
| Dokumente in der Hauptgruppe Brief 2 an HG | MEDIUM1 | 04\00\03\24\00000024.doc | |
| Dokumente in der Hauptgruppe Info 1 an HG | MEDIUM1 | 04\00\03\26\00000026.doc | |
| Dokumente in der Hauptgruppe Info 2 | MEDIUM1 | 04\00\03\27\00000027.doc | |
| Dokumente in der Nebenstelle 1 Brief 1 an NS | MEDIUM1 | 04\00\03\3C\0000003C.doc | |
| Dokumente in der Nebenstelle 1 Info 2 an NS | MEDIUM1 | 04\00\03\3F\0000003F.doc | |

Datei `os081102.400` für die Archivierung von W-Dokumenten

| Ordner | Bezeichnung | Medium | Pfad |
|---------------------------------------------|-------------|--------------------------|------|
| Dokumente in der Hauptgruppe simple 1 an HG | MEDIUM1 | 07\00\03\28\00000028.xml | |

| | | |
|-----------------------------------------------|---------|--------------------------|
| Dokumente in der Hauptgruppe invoice 1 an HG | MEDIUM1 | 07\00\03\29\00000029.xml |
| Dokumente in der Hauptgruppe mbox 1 an HG | MEDIUM1 | 07\00\03\2A\0000002a.xml |
| Dokumente in der Hauptgruppe mbox 2 an HG | MEDIUM1 | 07\00\03\2B\0000002b.xml |
| Dokumente in der Hauptgruppe test 1 an HG | MEDIUM1 | 07\00\03\2C\0000002c.xml |
| Dokumente in der Nebenstelle 1 simple 1 an NS | MEDIUM1 | 07\00\03\40\00000040.xml |

Dateios081102.700 für die Archivierung von XML-Dokumenten

In jeder Datei werden Ordner-, (wenn vorhanden) Register- und Dokumentenverschlagnwortung, Mediennamen und kompletter Dateiname des archivierten Dokumentes eingetragen.

Arbeit mit archivierten Dokumenten

Folgendes Szenario soll hier als Beispiel dargestellt werden. Ein Benutzer sucht ein bestimmtes Dokument und will es zum Lesen öffnen. Dazu ist als Erstes eine Suchanfrage mit der Anfragemaske zu stellen, die zum gesuchten Dokumententyp gehört:

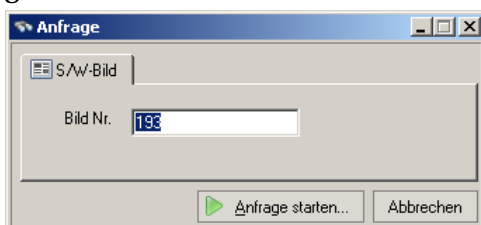


Abbildung 47: Anfragemaske nach einem D-Dokument mit der Bild Nr. 193*

Durch die Maske wird dem enaio® client mitgeteilt, um welchen Haupt- und Untertyp es sich handelt und in welchem Feld der Verschlagnwortung nach dem Suchbegriff gesucht werden soll.

In unserem Beispiel sind das:

- § Haupttyp: 2 (D-Dokument)
- § Untertyp: 0 (als Information aus der Objektdefinition)
- § Suchbegriff: 193* (in den Einstellungen, Registerkarte 'Auto' wurde Erweiterung am Ende ausgewählt)

Die Suche wird über folgendes SQL-Statement realisiert:

```
SELECT d.feld1, d.id, d.zeitstempel, d.version, d.links, d.flags,
       d.lockuser, d.haupttyp, d.anzahl, d.medium_doc
FROM object2 d, stamm1 o, sdrel x
WHERE x.stamm_id = o.id and x.object_id = d.id and
      ( {fn UCASE(d.feld1)} like '193%')
```

Die hier einbezogenen Datenbanktabellen `object2`, `stamm1`, und `sdrel` haben folgenden Inhalt:

| id | zeitstempel | haupttyp | untertyp | anzahl | flags | medium_doc | name_doc | feld1 |
|-----|-------------|----------|----------|--------|-------|------------|-----------------------|-------|
| 433 | 1124799483 | 2 | 21 | 1 | 2 | 4 | 02\15\B1\000001B1.000 | 19310 |
| 434 | 1124799503 | 2 | 21 | 1 | 2 | 4 | 02\15\B2\000001B2.000 | 19311 |
| 435 | 1124799515 | 2 | 21 | 1 | 2 | 4 | 02\15\B3\000001B3.000 | 19312 |
| 436 | 1124799529 | 2 | 21 | 1 | 2 | 4 | 02\15\B4\000001B4.000 | 19299 |
| * | | | | | | | | |

Abbildung 48: Datenbanktabelle `object2` mit 6 D-Dokumenten

| id | feld1 |
|-----|--------------------------------|
| 432 | Dokumente in der Hauptgruppe |
| 437 | Dokumente in der Nebenstelle 1 |
| * | |

Abbildung 49: Datenbanktabelle `stamm1` mit zwei Ordnern

| loeschen | zeitstempel | stamm_id | object_id | objekttyp | register | regtype | deleted |
|----------|-------------|----------|-----------|-----------|----------|---------|------------|
| | 1124289178 | 397 | 398 | 131072 | 0 | 0 | 0 |
| <NULL> | 1124289561 | 397 | 400 | 131072 | 0 | 0 | 0 |
| <NULL> | 1124290476 | 397 | 401 | 131072 | 0 | 0 | 1124290749 |
| <NULL> | 1124290752 | 397 | 402 | 196608 | 0 | 0 | 0 |
| <NULL> | 1124290823 | 397 | 403 | 196621 | 0 | 0 | 1124291194 |
| <NULL> | 1124291199 | 397 | 404 | 196621 | 0 | 0 | 1124292383 |
| <NULL> | 1124292386 | 397 | 405 | 196608 | 0 | 0 | 1124293464 |
| <NULL> | 1124292540 | 397 | 406 | 131072 | 0 | 0 | 0 |
| <NULL> | 1124293467 | 397 | 407 | 196608 | 0 | 0 | 0 |
| <NULL> | 1124293613 | 397 | 408 | 131072 | 0 | 0 | 0 |
| <NULL> | 1124792641 | 422 | 424 | 262151 | 423 | 6488066 | 0 |
| <NULL> | 1124792760 | 426 | 427 | 131089 | 0 | 0 | 0 |
| <NULL> | 1124799483 | 432 | 433 | 131093 | 0 | 0 | 0 |
| <NULL> | 1124799503 | 432 | 434 | 131093 | 0 | 0 | 0 |

Abbildung 50: Datenbanktabelle `sdr1` für die Dokument-Ordner-Register-Relationen

Die Ausführung dieser Suche führt zu folgender Trefferliste:

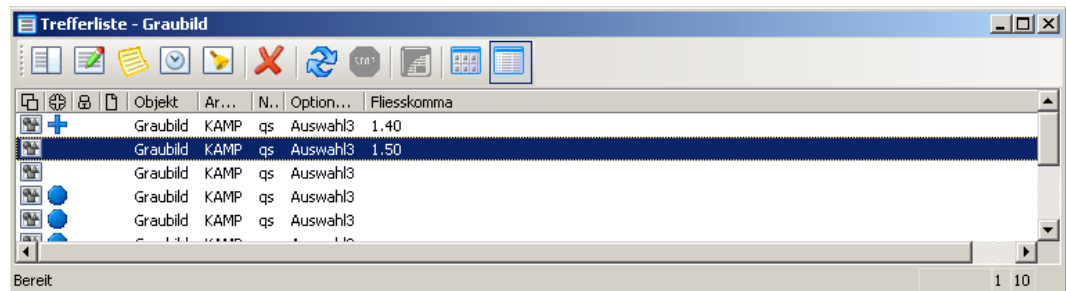


Abbildung 51: Trefferliste einer Suche nach D-Dokumenten

Gefunden wurden wie erwartet (vgl. Abbildung 48) etliche Dokumente, von denen drei als archiviert (`object2.flag=0`) und drei als archivierbar (`object2.flag=1`) markiert sind. Für das markierte archivierte Dokument können über F8 folgende Informationen ermittelt werden:

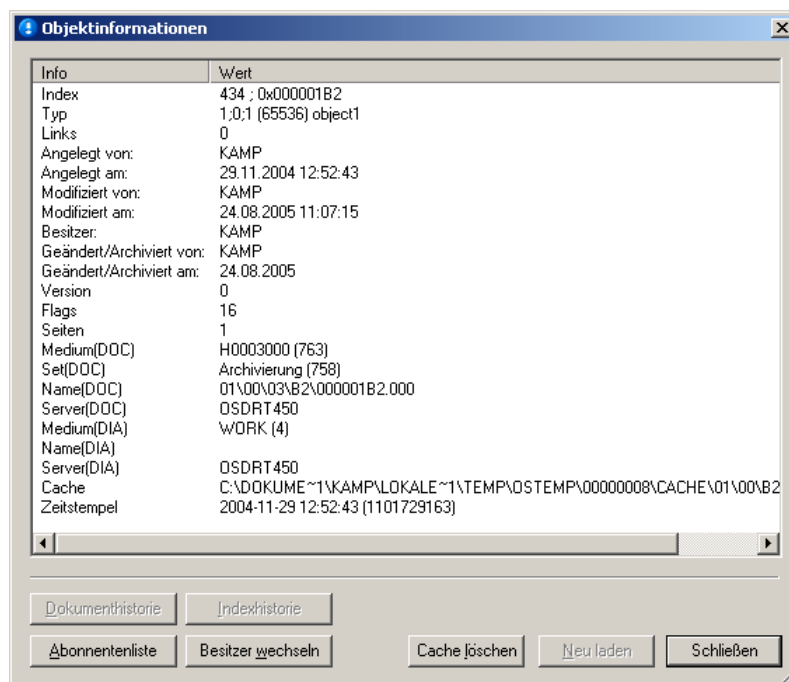


Abbildung 52: Informationen zum Dokument mit der id=434 (0x1B2)

Dieses Dokument soll nun beim Anwendungsserver angefordert werden. Es wird angenommen, dass sich das Dokument nicht im CACHE-Bereich befindet, sondern vom Archivmedium geholt werden muss. Dazu ist der vollständige Dateiname (Archivpfad, Medium und Name im Dateisystem) zu bestimmen.

Über das folgende SQL-Statement wird ermittelt, ob das Dokument ordnungsgemäß in den Datenbanktabellen eingetragen ist.

```
SELECT d.feld1, d.id, d.zeitstempel, d.version, d.links, d.flags,
```



```
d.lockuser, d.haupttyp, d.anzahl, d.medium_doc
FROM object2 d, stamm1 o, sdrel x
WHERE d.id = 433 and x.object_id=d.id and x.stamm_id=o.id
```

Anschließend werden die Verschlagwortung des Dokuments und die Systemeigenschaften eingelesen.

Verschlagwortung

```
select feld1 from object2 where id = 433
```

Systemeigenschaften

```
SELECT zeitstempel, haupttyp, untertyp, anzahl, flags, medium_doc,
medium_dia, name_doc, name_dia, angelegt, anleger,
archiviert, archivar, lockuser, foreignid, systemid
FROM object2
WHERE id=433
```

Damit stehen für weitere SQL-Anfragen bzw. für die Dokumentenanzeige die folgenden Informationen zur Verfügung:

- § flag=0 bedeutet archiviert
- § medium_doc ist die id des Archivmediums (4)
- § medium_name ist der Pfad im Dateisystem des Archivmediums (02\15\B1\000001B1.000)

Vor allem object2.medium_name='02\00\03\33\00000033.000' ist schon ein Teil des vollständigen Dateinamens des anzuzeigenden Dokumentes.

Nun kann der Name des Archivmediums ermittelt werden.

```
SELECT name, mirror_id, set_id, server_id
FROM medien
WHERE id=4
```

Damit sind die Werte für name='Medium1' (medien.name='Medium1'), mirror_id=82, set_id=76, server_id=3 bekannt und es kann der Name des Spiegelmediums ermittelt werden.

```
SELECT m2.name
FROM medien m1, medien m2
WHERE m1.name = 'MEDIUM1' AND m1.mirror_id = m2.id
```

Zum Schluss muss noch der Pfad zum Archiv für die Servergruppe server_id=3 ermittelt werden.

```
SELECT name
FROM path
where flag=1 and server_id=3
```

Als Resultat wird 'path.name= C:\Archiv' geliefert. Nun sind alle Elemente für den Dateistandort bekannt und es kann der vollständige Dateiname gebildet werden:

path.name\medien.name\object2.medium_name

Für das hier genannte Beispiel entsteht der Dateiname:

C:\Archiv\MEDIUM1\02\15\B1\000001B1.000

Externe Archivierungssysteme

Neben der oben beschriebenen Archivierung können ebenfalls externe Archivsysteme zur Archivierung eingebunden werden. Zur Verfügung stehen beispielsweise EMC-Centera, iTernity, iXOS und Tivoli Storage.

Für diese Lösung wird ein virtuelles Archiv für die Verbindung eingebunden, ein Medienset eingerichtet und diesem Medienset die gewünschten Dokumenttypen zugeordnet.

Die Archivierung selbst erfolgt, wie bei der Archivierung über einen enaio® server, als automatische Aktion.

Diese externen Archivierungssysteme unterstützen in der Regel Retentionszeiten und das gezielte Löschen von Dokumenten.

Beide Aspekte können besonders aus datenschutzrechtlichen Gründen wichtig sein. Detaillierte Informationen zu externen Archivierungssystemen stellen wir über den 'Storage Guide' zur Verfügung.

Administration

Überblick

Gegenstand dieses Kapitels sind die Tätigkeiten zur Administration eines enaio®-Systems. Da die Administration eine Vielzahl von verschiedenen Tätigkeiten umfasst, sollen zunächst die administrativen Tätigkeiten von anderen Tätigkeiten in und um das enaio®-System herum und anschließend die administrativen Tätigkeiten von den Produktiv-Tätigkeiten in einem enaio®-System abgegrenzt werden. Zuletzt werden die administrativen Tätigkeiten innerhalb eines enaio®-Systems differenziert und zusammenfassend beschrieben.

Abgrenzung: Administrations-Tätigkeiten außerhalb des enaio®-Systems

Die administrativen Tätigkeiten im Rahmen eines enaio®-Systems sind von allgemeinen administrativen Tätigkeiten im EDV-Umfeld des enaio®-Systems abzugrenzen. Diese zuletzt genannten Tätigkeiten dienen ebenfalls zur Vorbereitung und zum Betrieb des enaio®-Systems, sind aber nicht Gegenstand einer Beschreibung im Rahmen dieses Systemhandbuchs.

Dazu zählen folgende Tätigkeiten:

- § Hardware-Installation und Einrichtung:
 - § Netzwerk
 - § PCs
 - § Jukebox
 - § Scanner
 - § Backup-Hardware
 - § Sonstige Hardware (z.B. OCR-Dongles)
- § Software-Installation und Einrichtung:
 - § Betriebssystem
 - § Netzwerk, Domänenverwaltung
 - § Datenbank
 - § Email
 - § Treiber
 - § Office
 - § System-Backup
 - § DB-Sicherung und -Pflege
- § Behebung von Störungen außerhalb des enaio®-Systems:

- § Hardware-Fehler
- § Betriebssystemfehler
- § Datenbankfehler
- § Sonstige Softwarefehler, z.B. in MS Office, Treiberfehler

All diese Tätigkeiten sind nicht Gegenstand der Beschreibung.

Abgrenzung: Produktive und administrative Tätigkeiten innerhalb von enaio®

Innerhalb eines enaio®-Systems sind administrative Tätigkeiten von produktiven Tätigkeiten abzugrenzen. Produktive Tätigkeiten sind Tätigkeiten, die im enaio® client, in enaio® capture und in anderen Client-Anwendungen durchgeführt werden, wozu auch die Verwendung z.B. des Archivdruckers oder der Übernahmemaakros in MS Office Produkten zählt. Im enaio® client und enaio® webclient sind für alle Benutzer hauptsächlich produktive Tätigkeiten möglich.

Als produktive Tätigkeiten seien alle Tätigkeiten bezeichnet, bei denen durch enaio®-Benutzer im enaio® client, in enaio® capture und anderen Clientanwendungen eine der folgenden Aktionen ausgeführt wird:

- § DMS-Objekte werden recherchiert.
- § DMS-Objekte werden erzeugt.
- § DMS-Objekte werden gelöscht.
- § Die Verschlagwortung der DMS-Objekte wird geprüft.
- § Die Verschlagwortung der DMS-Objekte wird geändert.
- § Dokumentdateien der DMS-Objekte werden geprüft.
- § Dokumentdateien der DMS-Objekte werden validiert.
- § Dokumentdateien werden gelöscht.

Außer diesen produktiven Tätigkeiten nehmen Benutzer im enaio® client, in enaio® capture und in anderen Clientanwendungen selbst verschiedene vorbereitende Einstellungen vor, wie z. B. die Einrichtung der grafischen Oberfläche, Auswahl der anzuzeigenden Felder, Anfrageverhalten usw. Diese Tätigkeiten werden ebenfalls nicht zu den administrativen Systemtätigkeiten gezählt, die hier Gegenstand der Beschreibung sind.

Administrative Tätigkeiten sind als Tätigkeiten aufgefasst, die von besonders autorisierten Personen zur Vorbereitung der produktiven Tätigkeiten im enaio®-System, zur Erhaltung und zur Erweiterung der Systemfunktionen ausgeführt werden. Diese Arten der administrativen Tätigkeit kann man kurz als System-Vorbereitung, -Pflege und -Erweiterung bezeichnen.

Eine besondere Klasse von administrativen Tätigkeiten sind automatische Importaktionen, z.B. der dBase-III-Import oder der HL7-Server, die von Administratoren konfiguriert und ausgeführt werden. Diese Systemkomponenten können fast alle Aktionen ausführen, die oben als produktive Tätigkeiten definiert sind. Deren Ausführung erfolgt aber nicht in einer der o. g. Clientanwendung, sondern innerhalb eines Administrations-Werkzeugs und – nach erfolgter Konfiguration – ohne Benutzereingriff.

enaio®-Administration

Die administrativen Tätigkeiten können nach ihrer Funktion für das enaio®-System differenziert werden und lassen sich innerhalb des enaio®-Systems in Tätigkeiten zur Vorbereitung des Produktiveinsatzes von enaio® und Tätigkeiten zur Systemerhaltung und -pflege einteilen. Dazu kommen Tätigkeiten, die über die produktiven Tätigkeiten hinausgehende Funktionen bereitstellen. Dieses sind z.B. die Kommunikation mit anderen Systemen und die Konfiguration und Durchführung von Datenimporten. Zusammengefasst erfolgt die Einteilung der administrativen Tätigkeiten in:

- § Vorbereitung des Produktivbetriebs,
- § Systemerhaltung und -Pflege,
- § Erweiternde Tätigkeiten.

Diese drei Tätigkeitsbereiche sollen im Folgenden zusammengefasst werden.

Vorbereitung des enaio®-Produktivbetriebs

Damit im enaio® client überhaupt ein Objekt erstellt werden kann, sind eine Reihe von administrativen Tätigkeiten auszuführen. Als Minimum sind folgende Tätigkeiten erforderlich:

- § Installation des enaio®-Systems (siehe dazu Kapitel Installation dieses Systemhandbuchs),
- § Definition folgender Objekte im enaio® editor:
 - § Ein Schrank,
 - § Ein Dokumententyp, z.B. D-Dokument.
- § Einrichtung des Sicherheitssystems im enaio® administrator:
 - § Erstellung einer Benutzergruppe,
 - § Erstellung eines Benutzers für diese Benutzergruppe,
 - § Zuordnung der erstellten Objekttypen zur Benutzergruppe,
 - § Vergabe von Zugriffsberechtigungen für dieses Objekt an die Benutzergruppe.
- § Einrichtung des Lizenzsystems im enaio® enterprise-manager:
 - § Hinzufügen einer neuen Station,
 - § Zuordnung der erforderlichen Lizenzmodule zu der neuen Station.

Nach diesen Aktionen und dem erforderlichen Neustart des Anwendungsservers ist ein Benutzer an einer Arbeitsstation in der Lage, den enaio® client zu starten, Ordner in einem Schrank anzulegen und in diesen Ordnern S/W-Dokumente anzulegen. Ob er bei der Erfassung der Dokumente einen Scanner einsetzen kann, hängt davon ab, ob ein Scanner an der Station angeschlossen und richtig eingerichtet ist.

Damit ist erst ein kleiner Teil der möglichen Systemeigenschaften eingerichtet. Wenn man die volle Leistungsfähigkeit des Systems nutzen will, dann sind weitere Konfigurationstätigkeiten erforderlich.

- § Definition weiterer Objekte:
 - § Schränke,
 - § Register,
 - § Dokumenttypen.
- § Einrichtung des Sicherheitssystems im enaio® administrator:

- § Erstellung mehrerer Benutzergruppen,
- § Erstellung mehrerer Benutzer und Zuordnung zu Benutzergruppen,
- § Differenzierte Zuordnung der erstellten Objekttypen zu den Benutzergruppen,
- § Differenzierte Vergabe der Zugriffsberechtigungen für die Objekte an die Benutzergruppen.
- § W-Modul-Einrichtung für W-Dokumenttypen:
 - § Vorbereitung der Dokumentvorlagen (keine enaio®-spezifische Administrationstätigkeit),
 - § Einrichtung von Word- bzw. Excel-Vorlagen für die Datenübernahme.
- § Einrichtung des Lizenzsystems im enaio® enterprise-manager:
 - § Hinzufügen aller erforderlichen Stationen,
 - § Zuordnung der erforderlichen Lizenzmodule zu den neuen Stationen.
- § D-Modul-Einrichtung für die Verwendung des Archivdruckers,
- § Einrichtung von Events und sonstiger Scriptings (enaio® events, AddOn, enaio® capture-Skripte),
- § Manuelle Eingriffe in Konfigurationsdateien (as.cfg, ...) und Registrierung,
- § enaio® capture-Administration (siehe dazu Kapitel 'enaio® capture'.),
- § Workflow-Einrichtung (siehe dazu das Systemhandbuch 'Workflow Management System').

Systemerhaltung und -Pflege

Die zweite Gruppe von Tätigkeiten dient der Erhaltung und Optimierung der eingerichteten Systemfunktionalitäten. Folgende Tätigkeiten fallen darunter.

- § Einrichtung der Protokollierung,
- § Reparaturinstallationen,
- § Fehlerbehebungen,
- § Einrichtung und Ausführung von automatischen Aktionen zur Systempflege.

Erweiternde Administrationstätigkeiten

Als letzte Gruppe werden administrative Tätigkeiten zur Erweiterung der Systemfunktionalitäten über die produktiven Tätigkeiten beschrieben. Dies sind folgende Tätigkeiten:

- § Erweiterungsinstallationen,
- § Konfiguration des Archivsystems,
- § Einrichtung und Ausführung von automatischen Aktionen für Datenimporte,
- § Einrichtung und Ausführung von automatischen Aktionen für Dokumentenbearbeitung,
- § Inbetriebnahme von HL7-Servern.
- § Inbetriebnahme von Dicom-Servern.

Vorbereitung des Produktivbetriebs

Installation

Installationen sind in diesem Handbuch detaillierter im eigenen Kapitel 'Installationen' beschrieben. Eine schrittweise Beschreibung des Ablaufs einzelner Installationsvorgänge erhalten Sie im Handbuch 'Installationsbeschreibung'. Im Folgenden soll nur eine kurze Zusammenfassung der Systeminstallation erfolgen.

Installationen sind – nach den logistischen Vorbereitungen im DV-Umfeld – der erste Schritt zur Inbetriebnahme eines enaio®-Systems. Die Installationsverläufe sind vielfältig, da das zu erstellende Zielsystem sehr unterschiedliche Formen haben kann. enaio® ist so flexibel, dass Systemkonfigurationen von einem Einplatzsystem bis zu einem komplexen System im Netzwerk möglich sind.

Bei Einplatzsystemen wird zunächst eine Anwendungsserver-Installation, danach eine lokale Clientinstallation durchgeführt. Anschließend erfolgen die weiteren Administrationstätigkeiten.

Einfache Systeme bestehen aus einem Anwendungsserver, einem Fileserver und einer Anzahl Clients. Der erste Schritt ist die Installation eines Anwendungsservers. Nach dem Serverstart können Fileserver und Clients installiert werden. Parallel zu den weiteren Administrationstätigkeiten wie die Erstellung der Objektdefinition kann mit der Installation von weiteren Clients begonnen werden.

Größere Systeme haben mehrere Anwendungsserver in mehreren Servergruppen. Die Anzahl der Clients ist erheblich größer. Hier ist der Installationsprozess komplexer, aber auch dabei gilt, dass als erste Aktion ein Anwendungsserver installiert werden muss. Die anderen Anwendungsserver können gleich anschließend oder später installiert werden. Der einer Servergruppe zugeordnete Fileserver kann erst installiert werden, wenn mindestens ein Server in der Gruppe installiert und gestartet ist. Nachdem ein Administrations-Client eingerichtet worden ist, kann mit den weiteren administrativen Tätigkeiten fortgefahren werden. Anschließend oder parallel dazu werden dann die Netzwerkinstallation der Clients durchgeführt.

Die Installation erfolgt dialoggesteuert über einen Installations-Assistenten. Der Installationsverlauf verzweigt in Abhängigkeit der Entscheidungen des Administrators. Zur Erleichterung der Aufgaben werden dem Administrator zwei wichtige Hilfsmittel bei der Installation bereitgestellt. Das eine ist der Wartungsmodus, der Reparatur und Aktualisierung von Installationen beschleunigt, das andere ist das Silent Setup, dass die automatisierte Installation von Clients erlaubt.

Erstellung der Objektdefinition

Die Definition von DMS-Objekten ist unerlässliche Voraussetzung für den Betrieb eines enaio®-Systems. Ohne definierte Objekte ist keine sinnvolle Systemfunktionalität verfügbar. Erst mit einer adäquaten Objektdefinition ist die volle Leistungsfähigkeit des Systems abrufbar.

Zur Erstellung einer Objektdefinition ist der enaio® editor zu starten. Dieser muss installiert sein, der Anwendungsserver muss gestartet sein und die Station, an der der Editor ausgeführt wird, muss über eine ausreichende Lizenzierung verfügen. Der Benutzer, der sich anmeldet, muss die Berechtigung haben, den grafischen Editor zu starten, die Objektdefinition zu bearbeiten und eine Datenbankanpassung durchzuführen, damit er alle beschriebenen Aktionen durchführen kann.

Beim ersten Start des grafischen Editors ist noch keine Objektdefinition vorhanden. In dieser Situation gibt es zwei Handlungsmöglichkeiten:

- § Sie erstellen eine neue Objektdefinition.
- § Sie importieren eine vorhandene Objektdefinitions-Datei.

Im zweiten Fall ist eine Objektdefinition eines anderen enaio®-System vorhanden oder die Consulting-Abteilung von OPTIMAL SYSTEMS hat eine Objektdefinition vorbereitet. Dieser Fall soll im Weiteren nicht behandelt werden, da dabei einige Bearbeitungsschritte entfallen. Daher soll nur die Erstellung einer neuen Objektdefinition vollständig beschrieben werden.

Bei der Erstellung einer neuen Objektdefinition führen Sie folgende Schritte durch:

- § Öffnen Sie die aktuelle Objektdefinition. Da diese leer ist, ist der Strukturbaum, der alle Objekte anzeigt, ebenfalls leer.
- § Legen Sie einen neuen Schrank an.
- § Erstellen Sie die Felder zur Verschlagwortung des Schrankes.
- § Stellen Sie die Feldeigenschaften des Schrankes ein.
- § Legen Sie, falls erforderlich, ein neues Register im Schrank an.
- § Erstellen Sie die Felder zur Verschlagwortung des Registers.
- § Stellen Sie die Feldeigenschaften des Registers ein.
- § Wiederholen Sie das Anlegen von Register so oft wie erforderlich.
- § Legen Sie einen neuen Dokumenttyp an.
- § Erstellen Sie die Felder zur Verschlagwortung des Dokumenttyps.
- § Stellen Sie die Feldeigenschaften des Dokumenttyps ein.
- § Wiederholen Sie das Anlegen von Dokumenttypen so oft wie erforderlich.
- § Wiederholen Sie das Anlegen von Schränken, darin Register# und Dokumenttypen so oft wie erforderlich.
- § Führen Sie eine Datenbank Anpassung durch.

Danach stehen die Objekte zur Verfügung. In der Datenbank werden einerseits alle erforderlichen Stamm-, Register- und Objekt-Tabellen angelegt, andererseits sind in der Objektdefinitionsdatei die Informationen über die Datenblätter, Felder und Feldeigenschaften abgelegt. Um die Objekte verwenden zu können, sind der Anwendungsserver oder mindestens die betroffenen Exekutoren neu zu starten. (Dies kann automatisch durch den enaio® editor ausgelöst werden.) Außerdem sind noch weitere Einstellungen im enaio® administrator vorzunehmen.

Objekttypen

In einer Objektdefinition können verschiedene Objekttypen angelegt werden. Dabei sind Schrank-, Register und Dokumenttyp-Objekte zu unterscheiden. In der folgenden Tabelle sind alle zurzeit verfügbaren Objekttypen aufgeführt.

| Objekttyp | Objekttyp-Name |
|-----------|-------------------------------|
| 0 | Schrank |
| 99 | Register |
| 1 | X-Dokument (Graustufenbild) |
| 2 | D-Dokument (S/W-Bild) |
| 3 | P-Dokument (Farbbild) |
| 4 | W-Dokument (Windows-Dokument) |
| 5 | M-Dokument (Film-Dokument) |

| | |
|---|---------------------|
| 6 | E-Dokument (E-Mail) |
| 7 | XML-Dokument (XML) |

Tabelle 37: Objekttypen

Außerdem gibt es noch die modulübergreifenden Dokumenttypen, deren Objekttyp erst beim Anlegen eines konkreten Dokuments festgelegt. Modulübergreifenden Dokumenttypen können als Objekttypen 1 bis 5 angelegt werden.

Dialogformulare

Für jeden Objekttyp, der entworfen werden soll, wird ein Dialogformular definiert. Diese Maske wird im enaio® client bei drei verschiedenen Benutzeraktionen verwendet:

- § Beim Neuanlegen von Objekten.
- § Beim Recherchieren von Objekten. Der Dialog wird in diesem Fall 'Anfrage' genannt.
- § Beim Anzeigen der Objektverschlagwortung. Der Dialog wird in diesem Fall 'Datenblatt' genannt.

Für Dialogformulare lassen sich verschiedene Eigenschaften einstellen. Dies sind Einstellungen zur Position und Größe, Flags zur Volltextindexierung u.a., sowie sonstige Einstellungen wie Hintergrundbild und Objekt-Icon.

Felder

Dialogformulare enthalten verschiedene Typen von Dialogelementen. Diese sind neben optischen Elementen wie Hintergrundbild und sonstige Grafiken vor allem Felder, die dem Benutzer die Eingabe von Daten ermöglichen.

Folgende Dialogelemente können in ein Dialogformular eingebunden werden:

- § Textfeld
- § Kontrollkästchen
- § Optionsschaltflächen
- § Gruppenfeld
- § Schaltfläche
- § Statischer Text
- § Grafik
- § Tabelle
- § PageControl

Diese Felder können frei positioniert und in der Größe eingestellt werden. Feldern kann eine Anzahl von Eigenschaften zugewiesen werden, die zur Flexibilität der Objektdefinition beitragen.

Die Bedeutung der einzelnen Eigenschaften können Sie der Dokumentation zum enaio® editor entnehmen.

Weitere Dialogelemente

Über die Eigenschaften der Felder lassen sich weitere Dialogelemente anzeigen, die Kataloge. Folgende Kataloge sind verfügbar:

- § Liste
- § Baum
- § Hierarchie

- § Datenbank
- § Struktur
- § AddOns:
 - § Recherche-AddOn
 - § AddOn zum Einfügen des Datenbankindex
 - § Datums-AddOn
 - § Web-AddOn
 - § VBScript-AddOn
 - § Euro-AddOn
 - § Benutzer-AddOn
 - § Benutzergruppen-AddOn

Kataloge sind Feldern fest zugeordnet und haben Funktionen in Bezug auf die Wertemengen bzw. Werte der Felder. Kataloge haben folgende Funktionen:

- § Eingabe von Werten in Felder:
 - § Recherche-AddOn
 - § AddOn zum Einfügen des Datenbankindex
 - § Datums-AddOn
 - § VBScript-AddOn
 - § Benutzer-AddOn
 - § Benutzergruppen-AddOn
- § Änderung von Daten, z.B. Währungskonvertierung:
 - § Euro-AddOn
 - § VBScript-AddOn
- § Filterung der in einem Feld angezeigten Daten:
 - § Liste
 - § Baum
 - § Hierarchie
 - § Datenbank
 - § Struktur
 - § VBScript-AddOn
- § Sonstige Funktionen:
 - § Steuerung von Programmen (VBScript-AddOn)
 - § Öffnen einer Website, Senden von E-Mails (Web-AddOn)

Das VBScript-AddOn kann aufgrund seiner Flexibilität für alle Katalogfunktionen eingesetzt werden. Da über VB Script Zugriff auf das komplette Objektmodell von enaio® möglich ist, sind noch weitere Funktionen denkbar.

Dem Handbuch zum enaio® editor können Sie die einzelnen Schritte zur Erstellung von Objektdefinition entnehmen.

Einrichtung des Sicherheitssystems

Nachdem die Objektdefinition fertig gestellt und die Datenbank angepasst wurde, ist im nächsten Schritt das Sicherheitssystem im enaio® administrator einzurichten. Die Reihenfolge kann im Prinzip auch umgekehrt werden, d.h. zunächst

Einrichtung des Sicherheitssystems und danach Erstellung der Objektdefinition. Diese Reihenfolge ist umständlicher, da die Erteilung von Rechten an den enaio®-Objekten erst nach ihrer Erstellung möglich ist und somit die erneute Bearbeitung des Sicherheitssystems erforderlich wäre.

Die Einrichtung des Sicherheitssystems ist erforderlich, um festzulegen, welche Benutzer die Berechtigung bekommen, erstellte Objekte zu bearbeiten und installierte enaio®-Anwendungen zu verwenden. Zur Vereinfachung der Erstellung vieler Benutzer ist es möglich, Profilbenutzer anzulegen. Die Einstellungen des Profilbenutzers können mit einfachen Mitteln neuen Benutzern zugewiesen werden.

Folgende Reihenfolge bei der Einrichtung des Sicherheitssystems wird empfohlen:

1. Erstellung einer Benutzergruppe.
2. Zuweisung von Objektrechten zur Benutzergruppe.
3. Erstellung von Klauseln zu den Objektrechten.
4. Erstellung aller erforderlichen Gruppen in den Schritten 1. bis 3.
5. Erstellung eines Benutzers.
6. Erteilung von Systemrollen zum Benutzer.
7. Zuordnung von Benutzern zu einer oder mehreren Systemgruppen.
8. Erstellung aller erforderlichen Benutzer in den Schritten 5. bis 7.

Diese Reihenfolge ist ein Anhaltspunkt. Es ist auch möglich, zunächst eine Benutzergruppe anzulegen, danach alle Benutzer, die in dieser Gruppe aufgenommen werden sollen. Danach wird für jede weitere Gruppe auf dieselbe Weise verfahren. Bei diesem Verfahren müssen diejenigen Benutzer, die mehreren Gruppen zugehören sollen, entweder mehrmals oder erst nach Erstellung aller erforderlichen Gruppen bearbeiten werden. Das Sicherheitssystem kann jederzeit erweitert oder geändert werden.

Mit Installation von enaio® wird ein Benutzer 'Root' und eine Benutzergruppe 'Standard' angelegt. Root hat administrative Rechte und gehört zur Gruppe Standard. Es wird empfohlen, einen neuen Administrator zu erstellen, und den Benutzer Root zu entfernen. Zumindest sollte man sofort das Passwort dieses Benutzers ändern. Beachten Sie bitte, dass der letzte Benutzer mit Administrationsrechten nicht entfernt werden kann.

Wenn Sie nach Installation von enaio® keine administrativen Anwendungen starten können, da Sie das Passwort des Benutzers Root nicht kennen, dann kontaktieren Sie bitte den Support von OPTIMAL SYSTEMS.

Benutzergruppen und Objektrechte

Die Berechtigung, DMS-Objekte zu bearbeiten, wird Benutzergruppen zugewiesen. Benutzergruppen umfassen somit eine Anzahl von Benutzern mit identischen Rechten auf eine Anzahl von DMS-Objekten. Jeder Benutzer kann mehreren Benutzergruppen zugeordnet werden. Wenn ein Benutzer zu mehreren Gruppen gehört, gelten bei konkurrierenden Berechtigungen stets die weiterreichenden Berechtigungen, komplementäre Berechtigungen sind additiv. Benutzer bekommen bei mehrfacher Gruppenzugehörigkeit immer die Vereinigungsmenge der Rechte beider Gruppen. Diese Regel gilt auch für Klauseln, die für Objektrechte erstellt werden können.

Gruppenzugehörigkeit wird beim Anlegen von neuen Benutzern festgelegt und kann nachträglich geändert werden. Falls die neuen Benutzer produktiv tätig werden sollen, d.h. DMS-Objekte bearbeiten sollen, dann sollten zuerst

Benutzergruppen angelegt und Objektrechte vergeben werden. Für reine Administrationstätigkeit ist keine Erteilung von Objektrechten erforderlich.

Es bietet sich an, mit den Benutzergruppen die Struktur der Organisationseinheit abzubilden, in der das enaio®-System eingesetzt wird. Erfolgt z.B. der Einsatz des Systems in einem Krankenhaus, dann kann die Erstellung von stationsbezogenen Benutzergruppen sinnvoll sein. Gleichzeitig ist vielleicht die Abbildung des Ranges einer Person von Bedeutung, so dass für hervorgehobene Personen, z.B. Oberärzte und Chefärzte, eine oder mehrere eigene Benutzergruppen mit erweiterten Rechten angelegt werden. Dies sind aber nur grobe Anhaltspunkte für den Entwurf von Benutzergruppen. Detaillierte Informationen zum Design von Benutzergruppen können Sie bei der Consulting-Abteilung von OPTIMAL SYSTEMS erfragen.

Das Sicherheitssystem bietet die Möglichkeit, die enaio®-Benutzergruppen mit Benutzergruppen von NT-Domänen abzugleichen. Dabei können die Namen der Benutzergruppen aus der NT-Domäne, an der der Administrator angemeldet ist, übernommen werden. Diese Möglichkeit finden Sie in der Benutzergruppen-Einrichtung von enaio®.

Bei diesem Abgleich stehen nur die Gruppennamen zur Verfügung, da die enaio®-Berechtigungen nicht aus den Domänen-Berechtigungen abgeleitet werden können. Diese Gruppen müssen nach Abgleich wie beschrieben eingerichtet werden.

Objektrechte

Der Umfang dieser Zugriffsrechte unterscheidet sich nach den Objekttypen Ordner, Register und Dokumenttypen.

Ordner-Zugriffsrechte:

- § Index anzeigen (R). Die Verschlagwortung des Ordners darf eingesehen werden, wenn dieses Recht erteilt wurde. Dieses Recht ist Voraussetzung für alle anderen Rechte. Hat ein Benutzer nicht das Ordner-Zugriffsrecht 'Indexdaten anzeigen', dann wird in der Objektsuche die Recherchemaske nicht angezeigt. Ebenfalls nicht angezeigt werden die Recherchemasken der zugeordneten Register und Dokumenttypen. Für den Benutzer ist also der gesamte Inhalt aller Ordner dieses Typs nicht recherchierbar.
- § Indexdaten schreiben (W). Die Verschlagwortung der erzeugten Ordner darf erzeugt oder geändert werden, wenn dieses Recht erteilt wurde. Dieses Recht ist Voraussetzung für die Rechte D, X und U. Der Benutzer kann auf Register und Dokumente aus diesen Ordnern zugreifen, wenn er entsprechende Rechte auf Register und Dokumenttypen besitzt und er über die Wiedervorlage, den Workflow, gespeicherte Anfragen, Mappen oder E-Mails auf den Inhalt des Ordners zugreifen kann.
- § Objekt löschen (D). Ordner dürfen gelöscht werden, wenn dieses Recht erteilt wurde. Benutzer mit dem Ordner-Zugriffsrecht 'Objekt löschen' können Ordner nur löschen, wenn sie für alle enthaltenen Objekte das Löschrecht haben.
- § Objekt ausgeben (X). Der Ordner darf geöffnet werden, der Inhalt darf angezeigt werden, wenn dieses Recht erteilt wurde.
- § Objekt schreiben (U). Ohne Funktion für Ordner und Register.

Register-Zugriffsrechte:

- § Index anzeigen (R). Die Verschlagwortung des Ordners bzw. Registers darf eingesehen werden, wenn dieses Recht erteilt wurde. Dieses Recht ist

Voraussetzung für alle anderen Rechte. Hat ein Benutzer nicht das Register-Zugriffsrecht 'Indexdaten anzeigen', wird in der Objektsuche die Recherchemaske nicht angezeigt. Der Inhalt von Register# kann allerdings recherchiert werden, wenn der Benutzer entsprechende Rechte auf die Register oder Dokumente im Register besitzt.

- § Indexdaten schreiben (W). Die Verschlagwortung der erzeugten Register darf erzeugt oder geändert werden, wenn dieses Recht erteilt wurde. Dieses Recht ist Voraussetzung für die Rechte D, X und U.
- § Objekt löschen (D). Register dürfen gelöscht werden, wenn dieses Recht erteilt wurde. Benutzer mit dem Zugriffsrecht 'Objekt löschen' können Register nur löschen, wenn sie für alle enthaltenen Objekte das Löschrecht haben.
- § Objekt ausgeben (X). Das Register darf geöffnet werden, der Inhalt darf angezeigt werden, wenn dieses Recht erteilt wurde.
- § Objekt schreiben (U). Ohne Funktion für Ordner und Register.

Dokumenttyp-Zugriffsrechte:

- § Index anzeigen (R). Die Verschlagwortung der erzeugten Dokumente darf eingesehen werden, wenn dieses Recht erteilt wurde. Dieses Recht ist Voraussetzung für alle anderen Rechte. Hat ein Benutzer nicht das Dokumenten-Zugriffsrecht 'Indexdaten anzeigen', wird in der Objektsuche die Recherchemaske nicht angezeigt. Ein Zugriff auf diese Dokumente ist ebenfalls weder über die Wiedervorlagen, noch über E-Mails, gespeicherte Anfragen, Mappen oder Workflow möglich.
- § Indexdaten schreiben (W). Die Verschlagwortung des erzeugten Dokuments darf erzeugt oder geändert werden, wenn dieses Recht erteilt wurde. Dieses Recht ist Voraussetzung für die Rechte D, X und U. Um Dokumente mit Seiten anzulegen, wird zusätzlich das Recht 'Objekt schreiben' benötigt.
- § Objekt löschen (D). Das Dokument darf gelöscht werden, wenn dieses Recht erteilt wurde.
- § Objekt ausgeben (X). Das Dokument darf geöffnet, ausgedruckt und exportiert werden, wenn dieses Recht erteilt wurde.
- § Objekt schreiben (U). Das Dokument darf angelegt werden, Dokumentdateien dürfen angelegt und überschrieben werden.
- § Gruppenanmerkungen (G). Es dürfen Anmerkungen zu Dokumenttypen erstellt werden, die nur für die Benutzergruppe des Benutzers sichtbar sind. Dieses Recht ist nur für Bilddokumente, also für D-, P- und X-Dokumenttypen relevant.
- § Öffentliche Anmerkungen (P). Es dürfen Anmerkungen zu Dokumenttypen erstellt werden, die nur alle Benutzer sichtbar sind, die das Dokument sehen dürfen. Dieses Recht ist nur für Bilddokumente, also für D-, P- und X-Dokumenttypen relevant.

Wenn Benutzer das Recht 'Objekt schreiben (U)' haben, nicht aber die Rechte 'Gruppenanmerkungen (G)' oder 'Öffentliche Anmerkungen (P)', dann ist ein Problem mit statischen Folien vorhanden. Statische Folien dürfen vom Benutzer nicht bearbeitet werden. Beim Öffnen des Dokuments durch diesen Benutzer werden daher die Folien in die Dokumentdatei fest 'eingebrannt'. Legt nun der Benutzer dieses Dokument ab, wird das Original verändert. Am besten kann das Problem umgangen werden, indem alle Benutzer, die die Dokumente bearbeiten dürfen, auch die jeweiligen Anmerkungsrechte erhalten.

Klauseln auf Objektrechte

Jedes der im Abschnitt 'Objektrechte' beschriebenen Rechte kann mit Klauseln beschränkt werden. Das Objektrecht wird für alle Objekte des gewählten Objekttyps eingestellt. Klauseln sind so gestaltet, dass sie mit den Systemvariablen der Benutzer-Session und der Verschlagwortung eines konkreten DMS-Objekts einen Booleschen Wert ergeben. Ist dieser Wert Wahr, dann tritt das Objektrecht in Kraft, ist das Ergebnis Falsch, dann wird das Objektrecht an diesem konkreten Objekt verweigert. Dies soll zunächst an einem einfachen Beispiel erläutert werden.

Beispiel: Benutzer 'B' gehört zur Gruppe 'G', die am Objekttyp 'O' nur das Recht 'Index anzeigen (R)' hat. Dieses Recht ist mit einer Klausel belegt. Diese Klausel ist einfach, sie lautet

`Ersteller = #BENUTZER#`

'Ersteller' ist ein Feld in der Datenmaske des Objekts, #BENUTZER# ist eine Systemvariable, die den Namen des angemeldeten Benutzers enthält. Die Klausel besagt also, dass Benutzer der Gruppe 'G' die Verschlagwortung von Objekte des Typs 'O' nur dann sehen dürfen, wenn das Feld 'Ersteller' den Namen des angemeldeten Benutzers enthält.

Einfache Klauseln haben folgende Syntax:

`Feldname Operator Wert`

Für Feldnamen können alle Felder der Objektdatenmaske gewählt. Außer diesen Feldnamen können Sie folgende Werte verwenden:

| Wert | Bedeutung |
|-----------------|------------------------------------------------------------------------|
| #BESITZER# | Besitzer des aktuellen Objektes. |
| #COMPUTER-IP# | IP-Adresse des Computers, an dem der aktuelle Benutzer angemeldet ist. |
| #COMPUTER-GUID# | GUID des Computers, an dem der aktuelle Benutzer angemeldet ist. |
| #COMPUTER-NAME# | Name des Computers, an dem der aktuelle Benutzer angemeldet ist. |

Tabelle 38: Felder in Klauseln

Bei Dokumenttypen können auch Basisparameterdaten verwendet werden:

| Wert | Bedeutung |
|----------------------|---------------------------------------------------------------------------------|
| #ANLAGEDATUM# | Datum der Erstellung des Objekts. |
| #ANLEGER# | Anleger des Objekts. |
| #ARCHIVAR# | Der letzte Bearbeiter des Objekts. Nach einer Archivierung nicht mehr änderbar. |
| #ARCHIVIERUNGSDATUM# | Das letzte Bearbeitungsdatum. Nach einer Archivierung nicht mehr änderbar. |

Tabelle 39 Zusätzliche Felder in Klauseln bei Dokumenten

Folgende Operatoren stehen zur Verfügung:

| Operator | Bedeutung |
|----------|-----------|
| = | Gleich |
| <> | Ungleich |

| | |
|-----|---------------------|
| < | Kleiner |
| > | Größer |
| <= | Kleiner oder gleich |
| >= | Größer oder gleich |
| In | Enthalten in |
| !in | Nicht enthalten in |

Tabelle 40: Verknüpfungen zwischen Feldern und Werten

Als Wert können Sie manuell einen konstanten Wert eintragen, oder einer der folgenden Variablen benutzen:

| Wert | Bedeutung |
|-----------------|-----------------------------------------------------------------------------------------------|
| #BENUTZER# | Name des aktuellen Benutzers. |
| #DATUM# | Das aktuelle Datum. |
| #GRUPPEN# | Die Gruppen, in denen der Benutzer Mitglied ist. Verwenden Sie hier Operator 'in' oder '!in'. |
| #COMPUTER-IP# | IP-Adresse des Computers, an dem der aktuelle Benutzer angemeldet ist. |
| #COMPUTER-GUID# | GUID des Computers, an dem der aktuelle Benutzer angemeldet ist. |
| #COMPUTER-NAME# | Name des Computers, an dem der aktuelle Benutzer angemeldet ist. |
| #NULL# | Entsprechendes Indexdatenfeld ist leer. |

Tabelle 41: Werte in Klauseln

Bei Dokumenttypen können auch Basisparameterdaten verwendet werden:

| Wert | Bedeutung |
|----------------------|---------------------------------------------------------------------------------|
| #ANLAGEDATUM# | Datum der Erstellung des Objekts. |
| #ANLEGER# | Anleger des Objekts. |
| #ARCHIVAR# | Der letzte Bearbeiter des Objekts. Nach einer Archivierung nicht mehr änderbar. |
| #ARCHIVIERUNGSDATUM# | Das letzte Bearbeitungsdatum. Nach einer Archivierung nicht mehr änderbar. |

Tabelle 42: Zusätzliche Werte in Klauseln bei Dokumenten

Für die manuelle Eingabe von Werten können Sie Platzhalter verwenden: '*' für beliebige und beliebig viele Zeichen und '?' für ein beliebiges Zeichen.

Einfache Klauseln können mit logischen Operatoren verbunden werden.

| Wert | Bedeutung |
|-------|----------------|
| UND | Logisches UND |
| ODER | Logisches ODER |
| NICHT | Negation |

Tabelle 43: Logische Operatoren in komplexen Klauseln

Der NICHT-Operator steht vor einer Klausel und hat Vorrang. UND und ODER werden von links nach rechts sequentiell abgearbeitet. Um den Vorrang der Operatoren zu ändern, können die Klauseln durch Klammern gruppiert werden. Mit diesen Mitteln erstellen Sie komplexe zusammengesetzte Klauseln, wie z.B. die folgende:

Ausdruck1 UND (Ausdruck2 ODER NICHT Ausdruck3)

Benutzer und Systemrollen

Die Berechtigung, enaio®-Anwendungen auszuführen und bestimmte Tätigkeiten in diesen Anwendungen auszuführen, werden nicht Benutzergruppen, sondern einzelnen Benutzern zugewiesen. Diese Berechtigungen heißen in enaio® Systemrollen. Verschiedene Systemrollen sind für unterschiedliche Clientanwendungen gültig. Eine Übersicht über die im System vorhandenen Systemrollen finden Sie im Anhang dieses Dokuments.

LDAP

Das Sicherheitssystem bietet die Möglichkeit, die Authentifizierung von Benutzern über einen externen LDAP-Verzeichnisdienst durchzuführen. LDAP steht für 'Lightweight Directory Access Protocol'. Wenn diese Form der Authentifizierung eingestellt ist, erfolgt die Passwort-Verwaltung von Benutzern nur noch über den LDAP-Dienst, in der DMS-Datenbank werden keine Passworte mehr gespeichert.

Die im LDAP-Dienst registrierten Benutzer können in das enaio®-Sicherheitssystem übernommen werden. Dazu muss zunächst in der Konfiguration des enaio®-Gesamtsystems die Verbindung zum LDAP-Dienst eingerichtet werden. Folgende Einstellungen sind dort vorzunehmen:

- § LDAP-Server: hier ist der Name des Rechners einzutragen, auf dem der LDAP-Verzeichnisdienst läuft.
- § Port: Dies ist der IP-Port, über den die Kommunikation mit dem LDAP-Server erfolgt.
- § Bindingstring: spezielle Zeichenfolge, die eine bestimmte Position innerhalb der Struktur des LDAP-Verzeichnisses beschreibt und den Startpunkt für Suchanfragen an den Dienst bildet.
- § Mapping: dabei werden die oft kryptisch anmutenden Variablennamen des LDAP-Dienstes in enaio®-Variablennamen umgesetzt, die leichter zu verstehen sind. Diese Namen werden für Anfragen an den Verzeichnisdienst verwendet.

Nachdem diese Einstellungen vorgenommen wurden, kann in der enaio®-Benutzerverwaltung der LDAP-Dienst für die Übernahme von Benutzern verwendet werden. In der LDAP-Benutzerverwaltung von enaio® können folgende Aktionen durchgeführt werden:

- § Anfragen an den LDAP-Dienst. Damit ist die gefilterte Recherche von LDAP-Benutzern möglich.
- § Übernahme von LDAP-Benutzern. Recherchierte Benutzer können in das Sicherheitssystem übernommen werden.
- § Abgleich mit dem LDAP-Dienst. Das Entfernen von Benutzern im LDAP-Dienst wird vom enaio®-Sicherheitssystem nicht automatisch erkannt. Daher muss der Abgleich mit dem LDAP-Dienst explizit ausgeführt werden.

Im Sicherheitssystem müssen LDAP-Benutzer weiter wie beschrieben konfiguriert werden, da die enaio®-Berechtigungen nicht aus den LDAP-Berechtigungen abgeleitet werden können. Das umfasst die Zuordnung zu Benutzergruppen und

die Erteilung von Systemrollen. Wenn Sie die LDAP-Authentifizierung aktiviert haben und der LDAP-Verzeichnisdienst nicht zur Verfügung steht, kann sich kein Benutzer anmelden. Verwalten Sie deshalb mindestens einen Benutzer, der über die Systemrolle 'Supervisor' verfügt, mit einem OS-Passwort. Dieser Benutzer kann den enaio® administrator oder den enaio® enterprise-manager starten und Einstellungen in der Benutzerverwaltung ändern. Für den Start müssen Sie zuerst die LDAP-Authentifizierung ausschalten. Dazu ändern Sie in der Registrierung über den Schlüssel ... \Schemata\4.0\Login den Wert der Zeichenkette LoginMode von '1' auf '0'. Danach können alle Benutzer, die über ein OS-Passwort verfügen, entsprechend ihrer Systemrollen Programme starten.

Profile

Zur Vereinfachung der Erstellung einer großen Anzahl von Benutzern können Profilbenutzer angelegt werden. Dafür wird ein Benutzer auf die übliche Weise angelegt, mit dem Unterschied, dass er die Eigenschaft 'Profilbenutzer' erhält. Dieser Benutzer wird Benutzergruppen zugeordnet und erhält Systemrollen. Danach werden mit der Anmeldung dieses Benutzers in den betreffenden Anwendungen alle Einstellungen, die an andere Benutzer verteilt werden sollen, vorgenommen.

Wurden alle erforderlichen Einstellungen vorgenommen, dann kann das Profil verteilt werden. Das bedeutet, dass dieses Profil neuen oder bestehenden Benutzern zugewiesen wird. Dies geschieht durch explizite Aktion des Administrators, d.h. das Profil muss jedem einzelnen Zielbenutzer aktiv zugewiesen werden.

Folgende Einstellungen des Profilbenutzers können neuen Benutzern zugewiesen werden:

- § Systemrollen,
- § Gruppenmitgliedschaft,
- § Einstellungen, die im enaio® client vorgenommen werden,
- § Gespeicherten Anfragen,
- § Erweiterten Anfragen,
- § Verknüpfungen zu externen Programmen,
- § Einstellungen für den Arbeitsbereich,
- § Einstellungen für die Navigation,
- § Scaneinstellungen.

Dezentrale Benutzeradministration

Innerhalb komplexer Umgebungen kann es sinnvoll sein, Bereiche für eine dezentrale Benutzeradministration durch einen lokalen Administrator einzurichten.

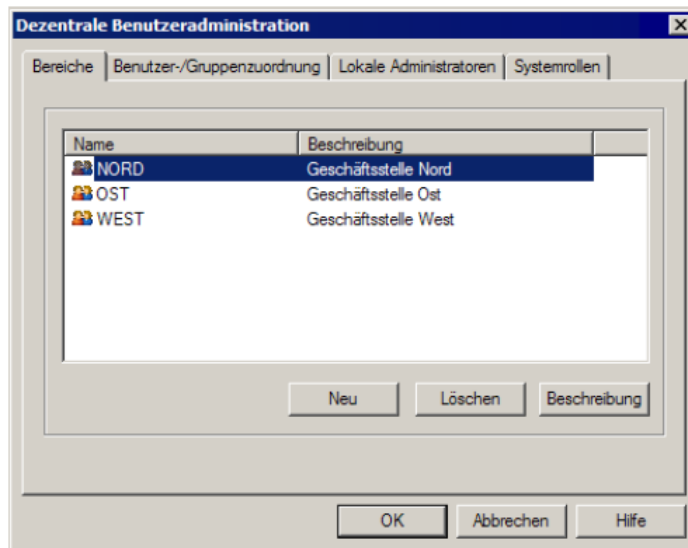


Abbildung 53: Dezentrale Benutzeradministration

Im enaio® administrator richtet der Systemadministrator lokale Bereiche ein und legt fest, welche Zugriffsrechte die Gruppen dieser Bereiche haben, wer lokale Bereiche administrieren kann und welche Systemrollen lokale Administratoren an die Benutzer ihrer Bereiche vergeben dürfen. Ein lokaler Administrator kann also in seinem lokalen Bereich Benutzer aufnehmen, diesen Benutzern Gruppen mit vorgegebenen Zugriffsrechten zuordnen und ihnen vorgegebene Systemrollen zuordnen. Der lokale Administrator arbeitet wie der Systemadministrator mit den Dialogen des Sicherheitssystems, der Funktionsumfang wird aber entsprechend eingeschränkt.

Active Directory Service

Eine Authentifizierung über einen Active Directory Service ist jetzt ebenfalls möglich. Über den enaio® enterprise-manager wird 'Active Directory' als Login-Modus angegeben. Zusätzlich wird die Domäne angegeben, an die enaio® server den Benutzer anmeldet.

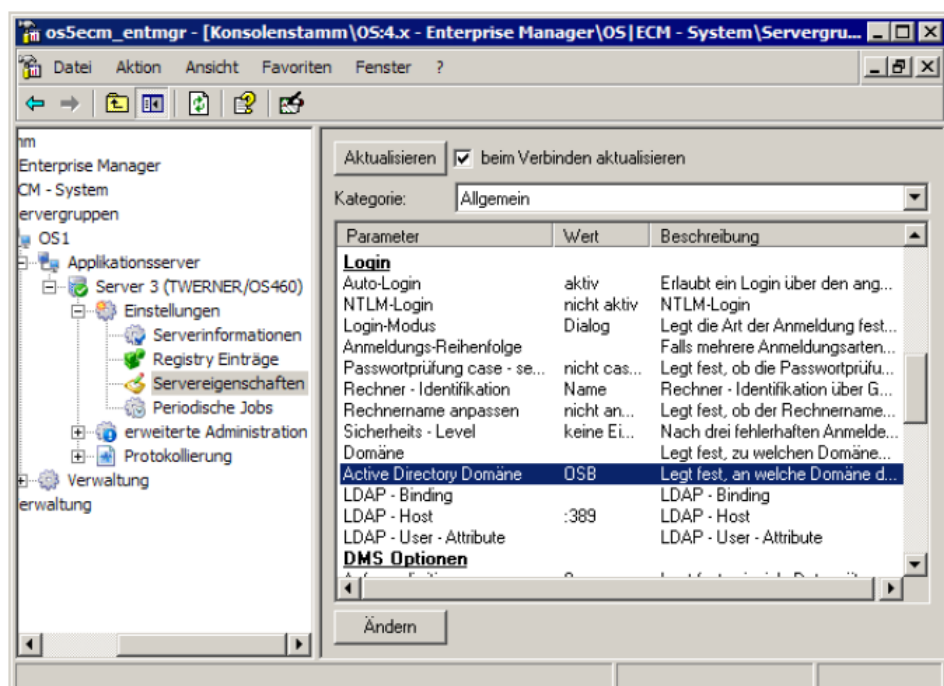


Abbildung 54: Active Directory Service

Authentifizierungs-Reihenfolge

In komplexen Umfeldern kann es notwendig oder sinnvoll sein, mehrere Authentifizierungsmechanismen einzusetzen. In diesem Fall geben Sie eine Authentifizierungs-Reihenfolge an. Scheitert eine Authentifizierung an einem System, wird in der angegebenen Reihenfolge jeweils eine weitere Authentifizierung über das folgende System versucht. Die Authentifizierungs-Reihenfolge geben Sie über den enaio® enterprise-manager an.

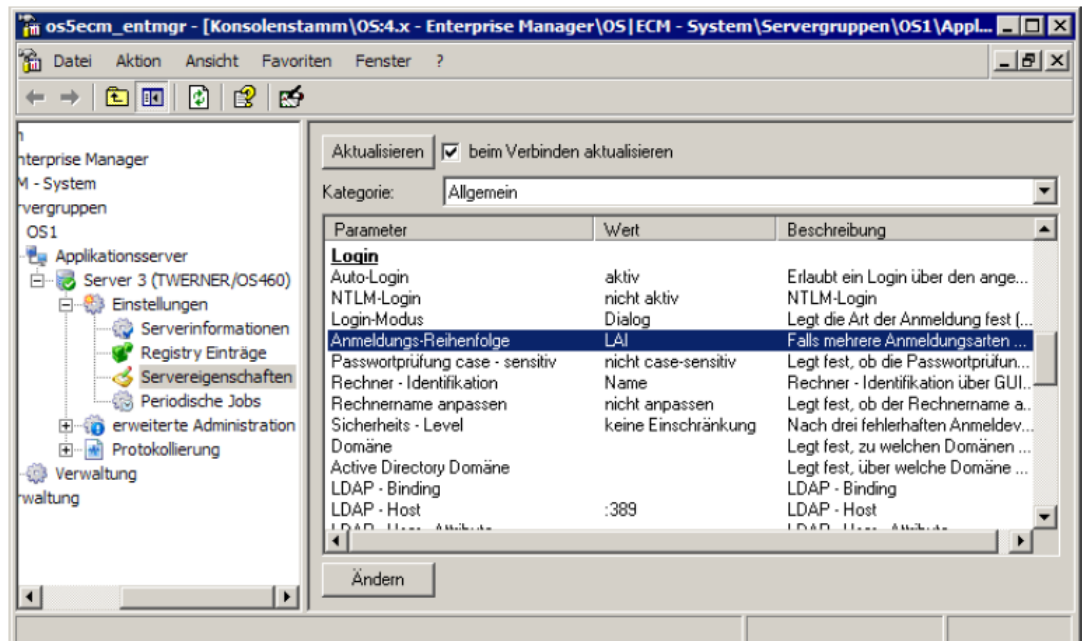


Abbildung 55: Authentifizierungs-Reihenfolge

Active Directory Synchronisation

Mit der Automatischen Aktion 'Active Directory Synchronisation' können Daten der OS-Benutzerverwaltung mit Daten einer Active Directory Benutzerverwaltung abgeglichen werden.

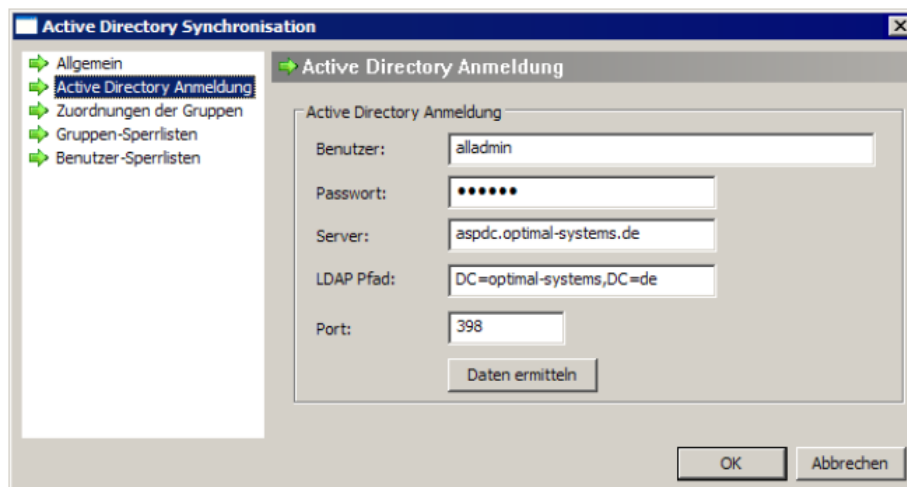


Abbildung 56: Active Directory Synchronisation

Die Aktion übernimmt Active Directory Gruppen und erstellt Zuordnungen zwischen Active Directory Gruppen und bereits bestehenden OS-Gruppen.

Änderungen innerhalb der Active Directory Benutzerverwaltung werden durch die Aktion ebenfalls für die OS-Benutzerverwaltung übernommen.

Zugleich können einzelne Gruppen und Benutzer über Sperrlisten von Änderungen ausgenommen werden. Für die Aktion binden Sie die Bibliothek 'axacuimp.dll' ein. Diese Bibliothek enthält ebenfalls die Funktionen für die Aktion 'Benutzer - und Gruppenimport'.

Einrichtung des Lizenzsystems

Bei der Einrichtung des Lizenzsystems wird festgelegt, an welchen Arbeitsplätzen welche Anwendungen und Module eingesetzt werden können. Die Berechtigung zum Ausführen von enaio®-Anwendungen wird über Lizenzen gesteuert.

Lizenzen werden bei OPTIMAL SYSTEMS erworben. Die Lizenzen werden in der verschlüsselten Datei `aslic.dat` übergeben. Diese Datei wird mit einem Import-Tool in die DMS-Datenbank importiert und dort in einem BLOB-Feld abgelegt. 'BLOB' bedeutet 'Binary Large Object'. In solchen Feldern werden keine einfachen Werte abgelegt, sondern komplexe Objekte, z.B. Dateien.

Mit dem Import der Lizenzdatei `aslic.dat` in die Datenbanktabelle `oslicense` werden dem System also alle zur Verfügung stehenden Lizenzen bekannt gemacht. Diese Lizenzen, die hier Module genannt werden, sind nun auf die vorgesehenen Arbeitsstationen zu verteilen. Dafür steht in der Lizenzverwaltung des enaio® enterprise-managers ein Tool zur Verfügung. Darin werden folgende Schritte zur Konfiguration durchgeführt:

§ Zunächst werden Stationen hinzugefügt. Stationen können aus der vorhandenen NT-Domäne übernommen oder manuell erstellt werden.

§ Den vorhandenen Stationen werden dann Module hinzugefügt.

Beim Beenden der Konfiguration des Lizenzsystems werden die Informationen in die Datenbank geschrieben. Wenn diese Schritte erfolgt und die Änderungen am Lizenzsystem gespeichert wurden, stehen den konfigurierten Stationen die zugeordneten Module zur Verfügung. Dies gilt unabhängig von Benutzern, da die Lizenzen stationsbezogen sind.

Die Identifikation von Arbeitsstationen erfolgt über die TCP/IP-Adresse, die GUID-Nummer oder den Rechnernamen. Der Rechnername wird über den Domänencontroller oder über die Datei 'hosts' ermittelt, intern wird aber immer die TCP/IP-Adresse oder die GUID-Nummer verwendet. In der Tabelle `osstations` werden die konfigurierten logischen enaio®-Arbeitsplätze gespeichert. In der Tabelle `osstationnc` werden die physikalischen GUID- und IP-Adressen der einzelnen Stationen gespeichert und logischen Stationen zugeordnet. Diese Adressen werden aus den vorhandenen Netzwerkkarten des Arbeitsplatzes ermittelt. Auf diese Weise ist es möglich, Stationen mit mehreren Netzwerkkarten zu verwalten. Die Tabelle `oslicresources` bindet die beiden Tabellen `osresources` und `osstations`. Hier werden die verfügbaren Ressourcen den einzelnen Stationen zugeordnet.

Während des Betriebs wertet der Anwendungsserver die vorhandenen und die angeforderten Lizenzen aus. Ist ein Maximum erreicht, dann wird die Erteilung der Lizenz verweigert und die Komponente startet nicht. In der Tabelle `ossession` werden die Sessions, d.h. die Verbindungen von Clientanwendungen mit enaio®-Servern, gespeichert. Jede erfolgreiche Client-Anmeldung bei einem Server bekommt einen eindeutigen Eintrag in dieser Tabelle. Die Session ist eindeutig mit einem Server und mit einer Station verbunden. In der Tabelle `oslockedres` werden die Ressourcen, die von einer Session angefordert und belegt wurden, gespeichert. Die Tabelle ist mit den Tabelle `ossession` und `osresources` verbunden.

Nur Clientverbindungen im Rahmen einer gültigen Session können Ressourcen wie z.B. Lizenzen belegen. Wenn ein enaio®-Server irregulär beendet wird, dann bleiben sowohl die Sessioneinträge als auch die Einträge der belegten Ressourcen erhalten, so dass ein anderer Server derselben Servergruppe die Session übernehmen kann.

Die Möglichkeit der Verteilung von Lizenzen auf Stationen hängt von der Art der Lizenz ab. Ein Modul kann auf zwei verschiedene Arten lizenziert sein:

- § Named: Ein Modul mit solch einer Lizenz kann nur auf einer genau bezeichneten Station ausgeführt werden. 10 Named Lizenzen können auf genau 10 Stationen ausgeführt werden, mehr nicht.
- § Concurrent: Concurrent Lizenzen ermöglichen die Ausführung von Modulen auf beliebig vielen Rechnern, solange die maximale Anzahl der Lizenzen nicht überschritten wird. 10 Concurrent Lizenzen können auf z.B. 100 Stationen ausgeführt werden, aber gleichzeitig können nur 10 Stationen arbeiten.

Für einige Module ist der Eintrag von Konfigurationen in den Lizenzen erforderlich, z.B. für automatische Importaktionen. Die eingetragene Anzahl von Konfigurationen kann nicht überschritten werden. Dieser Eintrag wird bei anderen Modulen verwendet, um die automatische Lizenzfreigabe zu steuern. Der Eintrag wird als Anzahl von Minuten interpretiert. Die automatische Lizenzfreigabe erfolgt dann, wenn die Anwendung während der eingetragenen Anzahl von Minuten untätig war.

Lizenzimport

Der erste Lizenzimport erfolgt bei der Installation des ersten Anwendungsservers. Ohne gültige Lizenz kann der Anwendungsserver nicht starten. Dabei kann es sich auch um eine temporäre Testlizenz handeln. Während der Installation wird die angegebene Lizenzdatei `aslic.dat` komplett in die Datenbank importiert, die beiliegende Datei `aslic.cfg` wird analysiert, der Inhalt wird in verschiedene Tabellen der Datenbank geschrieben. In der Datei `aslic.cfg` ist eine Default-Verteilung der Module eingetragen, die im Wesentlichen die Zuordnung der Anwendungsserver-Lizenz zur angegebenen Server-Station enthält. Der Anwendungsserver kann nur auf einer mit der GUID benannten Station ausgeführt werden. Diese GUID muss an OPTIMAL SYSTEMS vor Erstellung der Lizenz übermittelt werden.

Neu erworbene Lizenzen werden von OPTIMAL SYSTEMS ebenfalls in Form der Datei `aslic.dat` übermittelt. Diese Datei muss mit dem enaio® enterprise-manager in die Datenbank importiert werden.

Mit dem enaio® enterprise-manager stellen Sie eine ODBC-Verbindung her, indem Sie den Namen einer ODBC-Quelle, den Benutzernamen und das Kennwort eines autorisierten Datenbankbenutzers und das Verzeichnis, in dem die Lizenzdateien liegen, angeben. Danach kann der Import der Lizenzdateien ausgeführt werden.

Einrichtung von Dokumenttypen

Für einige der verfügbaren Objekttypen müssen nach der Erstellung der Objekte im enaio® editor zusätzlichen Konfigurationseinstellungen vorgenommen werden, z. B. das Drag & Drop-Verhalten von Emails oder die Einrichtung von Viewern.

Von den schon vorgestellten Objekttypen haben einige eingebaute Viewer im enaio® client. Zur Erläuterung soll hier die Tabelle mit den verfügbaren Objekttypen wiederholt dargestellt werden.

| Objekttyp | Objekttyp-Name |
|-----------|----------------|
|-----------|----------------|

| | |
|----|-------------------------------|
| 0 | Schrank |
| 99 | Register |
| 1 | X-Dokument (Graustufenbild) |
| 2 | D-Dokument (S/W-Bild) |
| 3 | P-Dokument (Farbbild) |
| 4 | W-Dokument (Windows-Dokument) |
| 5 | M-Dokument (Film-Dokument) |
| 6 | E-Dokument (E-Mail) |
| 7 | XML-Dokument (XML) |

Tabelle 44: Objekttypen

Die Objekttypen 'Schrank' und 'Register' bedürfen keines Viewers, da nur deren Verschlagwortung oder deren Inhalt dargestellt wird. Diese Funktion wird vom enaio® client erfüllt. Auch sonst sind keine weiteren administrativen Einstellungen erforderlich. Für alle Dokumenttypen, die als reine Fundstelle definiert sind und somit ohne Dokumentdatei gespeichert werden, gilt gleiches. X-, D- und P-Dokumenttypen werden durch im Client implementierte Viewer angezeigt. Daher ist die Konfiguration von Viewern nicht erforderlich.

W-Dokumente haben keine eingebauten Viewer, sie werden gerade dadurch definiert, dass eine Verknüpfung zu Windows-Anwendungen erforderlich ist. Für W-Dokumente müssen daher Viewer und Vorlagen konfiguriert werden. Es ist außerdem sicher zu stellen, dass an den betreffenden Stationen die entsprechende Anwendung zum Betrachten bzw. Bearbeiten der W-Dokumente installiert ist.

M-Dokumente werden vom enaio® client unter Verwendung von Funktionen des Windows Media Players angezeigt, auch hier ist keine Viewer-Konfiguration erforderlich. Es ist jedoch darauf zu achten, dass eine korrekte Installation des Windows Media Players vorhanden ist.

E-Dokumente können intern angezeigt werden oder über MS Outlook. MS Outlook Express wird nicht unterstützt. Auf welche Weise die Anzeige erfolgt, hängt vom Format der angezeigten E-Mail ab. Wenn E-Mails aus dem internen Posteingang von enaio® client als DMS-Objekte eingefügt werden, dann werden die E-Mails im MIME-Format abgelegt. In diesem Fall erfolgt die Anzeige über den internen Viewer. Erfolgt der Import per Drag & Drop aus MS Outlook heraus, dann erfolgt die Anzeige über MS Outlook. In beiden Fällen ist eine Viewer-Konfiguration nicht erforderlich. Für den zweiten Fall ist lediglich dafür zu sorgen, dass auf den Stationen MS Outlook installiert ist, an denen solche E-Mails betrachtet werden sollen. Das Drag & Drop von E-Mails muss manuell konfiguriert werden.

XML-Dokumente werden durch die Anzeige-Engine des Internet Explorers angezeigt. Empfohlen wird die Installation des Internet Explorers 6. Für die korrekte Anzeige der XML-Dokumente ist die Vorbereitung von Stylesheet-Vorlagen erforderlich. Außerdem ist die Bekanntgabe der Vorlagen durch manuelle Einträge in der Datei `as.cfg` erforderlich.

Dem Handbuch zum enaio® administrator können Sie die einzelnen Schritte zur Konfiguration von Dokumenttypen entnehmen.

Events und Scripting (Eventeditor, AddOn, Index-Skripte)

Alle Kernkomponenten von enaio® sind in der Programmiersprache Visual C++ programmiert, die eine hohe Performance bei großer Flexibilität gewährleistet. Einige Zusatzkomponenten sind in Visual Basic (VB) geschrieben. Visual Basic

erlaubt das effiziente Erstellen von leistungsfähigen Anwendungen mit besonderer Orientierung auf GUI-Funktionalität.

An mehreren Stellen im enaio®-System kommt eine Skript-Sprache zum Einsatz. Dabei handelt es sich um Visual Basic Script (VB Script). VB-Skript hat fast denselben Leistungsumfang wie VB. Der wesentliche Unterschied ist, dass VB Script keiner Kompilierung bedarf, sondern in der Laufzeitumgebung interpretiert wird, sofern der Windows Scripting Host installiert ist.

Mit VB Script können Sie an verschiedenen Stellen im enaio®-System eigene kleine Programmfunktionen einbauen und damit die jeweiligen Anwendungsfunktionen nach eigenen Wünschen anpassen. Außerdem können Sie eine große Anzahl von Funktionen realisieren, deren Implementierung in die Kernkomponente nicht wirtschaftlich ist.

Auf folgenden Stellen können Sie Skripte erstellen und ausführen lassen:

- § Events in enaio® client
- § Events in enaio® server
- § VBScript-AddOn im enaio® client
- § Precheck- und Aftercheck-Skripte in enaio® capture
- § Workflow-Events
- § Automatische Aktion Daten-/Dokumentenimport
- § Automatische Aktion Daten-/Dokumentenexport

Die Verwendung von VB Script in diesen Umgebungen soll im Folgenden zusammenfassend beschrieben werden. Eine ausführliche Beschreibung finden Sie in den jeweiligen Handbüchern zum enaio® client, zu enaio® capture, zum Workflow und zum enaio® administrator.

Events

Events sind Punkte im Ablauf des Client bzw. des Servers, die bei bestimmten Benutzeraktionen ausgelöst werden. Hier sollen Client-Skripte kurz eingeführt werden. Zum Einsatz und Implementierung von Server-Skripten lesen Sie bitte die entsprechenden Abschnitte in der Dokumentation zu enaio® server. Client-Skripte können z.B. beim Starten des enaio® client, beim Öffnen eines Datenblattes oder beim Verschieben von Objekten ausgeführt werden. Bei Eintreten eines solchen Events prüft enaio® client, ob dem betreffenden Objekt ein Skript für den eingetretenen Event zugeordnet ist. Wenn das der Fall ist, wird das entsprechende Skript ausgeführt.

Mit diesen Events können Daten validiert und geändert werden. Außerdem können externe Programme gestartet werden usw. In dem ausgeführten Skript können Sie den vollen VB-Script-Befehlssatz verwenden.

Events werden im Eventeditor-Fenster von enaio® client den vorhandenen DMS-Objekten zugeordnet und bearbeitet. Zum Ausführen des Eventeditors im enaio® client ist eine EVE-Lizenz erforderlich.

Events sind im Handbuch 'OS_Eventeditor_de.pdf' dokumentiert.

Eventskripte werden verschlüsselt in der Datenbank gespeichert. Die Eventklassen, die die möglichen Eventtypen darstellen, werden in der Tabelle `oseventcode` gespeichert. In der Tabelle `osevents` werden alle Events gespeichert. Sie können aber auch als Datei importiert oder exportiert werden.

Für die meisten Ereignisse erstellt der enaio® client eine Übergabedatei mit kontextbezogenen Daten, auf die Sie mit VB Script zugreifen und die Sie ändern können. Einige Ereignisse benötigen einen Rückgabewert, den Sie in die Übergabedatei eintragen. Einige Ereignisse aktualisieren Daten anhand der

geänderten Einträge in der Übergabedatei. Abhängig vom Rückgabewert führt der Client dann die Aktionen weiter aus oder führt sie nicht aus.

Der Client benötigt einen Rückgabewert für das Event. Dieser Wert muss im Code enthalten sein. `resultcode=1` speichert die Daten aus der Übergabedatei in der Datenmaske, `resultcode=0` ändert die Daten nicht, `resultcode=-2` überträgt die Daten aus der Übergabedatei in die Datenmaske, die Datenmaske bleibt im Client mit den geänderten Daten offen. `WriteToFile` speichert die Änderungen in der Übergabedatei ab.

Eventskripte importieren und exportieren

Events, die OPTIMAL SYSTEMS kundenspezifisch erstellt, werden als verschlüsselte Dateien ausgeliefert, die importiert, einem Archivobjekt zugeordnet und in der Datenbank gespeichert werden kann.

Sie können ebenfalls Events im enaio® client einzeln als verschlüsselte Datei speichern und importieren. Unverschlüsselte Dateien können nicht importiert werden, sie haben die Endung `.eve` und können mit Texteditoren betrachtet werden. Verschlüsselte Event-Dateien führen die Endung `.evc` und zeigen daher im Texteditor unsinnig anmutende Zeichenfolgen an. Ein Export/Import aller Skripte im System ist ebenfalls aus dem enaio® client möglich.

Beispiel für ein Eventskript

In diesem Skript wird der Name und Feldinhalt eines AddOn-Felds ausgegeben, und anschließend der AddOn-Feldinhalt modifiziert:

```
MsgBox (AddonFields.Item(0).Name + ' = ' + AddonFields.Item(0).Value)
AddonFields.Item(0).Value = 'neuer Wert'
WriteToFile()
```

Weitere Informationen können Sie dem Handbuch 'Events' entnehmen.

VBScript-AddOn

Ein weiterer Einsatz von VB Script im Client findet sich im VBScript-AddOn. Dieses AddOn ist ein Dialogelement, das einem Feld zugeordnet ist. Es wird im enaio® editor in den Eigenschaften des Feldes konfiguriert. Zusätzlich muss in der entsprechenden Sektion der Datei `as.cfg` für das Feld, dem das VBScript-AddOn zugeordnet ist, die folgende Zeile eingefügt werden:

```
EXTRA00=e:\os\vbscripts\scriptbezeichnung.vbs
```

Ist einem Verschlagwortungsfeld das VBScript-AddOn zugeordnet, dann erhält das Feld eine Katalogschaltfläche. Mit einem Klick der linken Maustaste wird das verbundene Script gestartet, mit einem Klick der rechten Maustaste wird der VB-Editor gestartet.

Das VBScript-AddOn ermöglicht, VB-Skripte auszuführen, um beispielsweise die Verschlagwortung von Feldern mit VB-Skripten zu bearbeiten. Die Ausführung eines Skripts mit dem VBScript-AddOn erfolgt automatisch beim Schließen eines Datenblattes, wenn dies so im enaio® editor konfiguriert ist, oder manuell durch Klicken auf die Katalogschaltfläche des AddOns. Das Editieren der Skripte erfolgt über dieselbe Schaltfläche, sofern eine Lizenz für den VB-Skript-Editor vorhanden ist.

VB-Skripte werden auf Anfrage von OPTIMAL SYSTEMS GmbH geliefert oder können mit dem VB-Editor erstellt werden. Das AddOn muss über das Modul VBX, der VB-Editor über das Modul VBE lizenziert werden. Für den VB-Editor benötigt ein Benutzer das in enaio® administrator erteilte Recht 'Client: Events erstellen'.

Weitere Informationen können Sie dem Handbuch zum enaio® editor entnehmen.

Skripte in enaio® capture

VB Script wird auch in der Validierungskomponente `axvalid.exe` von enaio® capture eingesetzt. In diesem Umfeld haben Skripte eine auf die Überprüfung, Filterung und automatische Änderung von Daten abzielende Funktion. Diese Funktionen werden hier Precheck und Aftercheck genannt. Daten werden vor dem Öffnen eines neuen Datensatzes und nach dem Schließen eines bearbeiteten Datensatzes getestet. Falls der Test negativ ausfällt, können die jeweiligen Daten automatisch geändert oder der Abschluss des Datensatzes abgebrochen werden, damit der Benutzer die Daten erneut bearbeiten kann. Fehlerhafte Daten können farblich hervorgehoben werden.

Die Skripte werden in den Batchoptionen in AXVALID konfiguriert. Die verwendeten Skriptdateien sind wie in den oben genannten Fällen verschlüsselt abgelegt. Eine Speicherung dieser Skripte erfolgt nicht in der DMS-Datenbank, sondern im Dateisystem.

Weitere Informationen können Sie dem Handbuch zu enaio® capture entnehmen.

Automatische Aktion Daten-/Dokumentenimport

Die automatische Aktion 'Daten-/Dokumentenimport' führt, wenn konfiguriert, für jeden Importdatensatz ein kleines VB Script aus. Mit diesem Skript können einerseits Prüfungen durchgeführt werden, die über die normale Validierung innerhalb des Daten-/Dokumentenimportes hinausgehen.

Ohne Skripte ist praktisch nur die Korrektheit der Datentypen validierbar. Mit den Skripten sind inhaltliche Prüfungen, z.B. die Plausibilität von Datumsangaben, und Wertersetzen, z.B. Ersetzung von Ausdrücken durch Katalognummern, möglich.

Automatische Aktion Daten-/Dokumentenexport

Die automatische Aktion 'Daten-/Dokumentenexport' führt, wenn konfiguriert, für jeden Exportdatensatz ein kleines VB Script aus. Mit diesen Skripten sind inhaltliche Prüfungen, z.B. die Plausibilität von Datumsangaben, und Wertersetzen, z.B. Ersetzung von Ausdrücken durch Katalognummern, möglich. Weiterhin kann am Ende eines Exports ein Skript gestartet werden. Darüber ist es z.B. möglich Aktionen direkt nach dem erfolgreichen Gesamt-Export durchzuführen.

Anmerkungen zum Scripting

VB Script ist eine mächtige Skriptsprache, die einen geringeren Umfang als Visual Basic hat. VB Script ist prozedurorientiert und wird interpretiert, d.h. im Gegensatz zu VB ist die Erstellung eigenständiger ausführbarer Dateien mit VB Script nicht möglich. Mit VB Script ist die Erstellung von COM-Objekten möglich, so dass Anwendungen wie MS Word, MS Excel und dem enaio® client gesteuert werden können. Die Erstellung eines COM-Objekts erfolgt über den Befehl:

```
MyAS = CreateObject('optimal_AS.application')
```

Das COM-Objekt der Clients heißt aus Kompatibilitätsgründen `optimal_AS`. Wenn ein Objekt schon vorhanden ist, d.h. wenn genau ein enaio® client in Betrieb ist, dann wird dessen COM-Schnittstelle verwendet. Ist kein Client-Objekt vorhanden, dann wird der zuletzt registrierte, enaio® client gestartet und seine COM-Schnittstelle verwendet. Sind mehrere Clients in Betrieb, dann werden das Objekt und die COM-Schnittstelle des zuletzt gestarteten Clients verwendet.

Das auf diese Weise erstellte Objekt bietet nun eine Reihe von Methoden an, mit denen innerhalb des Clients folgende Funktionen erfüllt werden können:

- § Objektbezogene Funktionen
- § Objekte recherchieren,

- § Objekte einfügen,
- § Objekte aktualisieren,
- § Informationen über Objekte einholen,
- § Objekte im Client öffnen,
- § Objekte ausdrucken
- § usw.
- § Steuerung des Clients
 - § Client aktivieren,
 - § Neuen Benutzer anmelden,
 - § Alle Fenster schließen,
 - § Lizenzen anfordern und freigeben,
 - § E-Mail versenden,
 - § usw.
- § Diverse Funktionen
 - § Informationen über Client-Environment einholen,
 - § XML-Dateien mit Stylesheets in ein Zielformat konvertieren,
 - § Bilddateien in anderes Zielformat umwandeln,
 - § Fehlerbehandlung innerhalb der COM-Schnittstelle,
 - § usw.

Nicht alle vorhandenen COM-Funktionen können in VB Script eingesetzt werden, da in VB Script die Dimensionierung von Variablen auf den Typ VARIANT beschränkt ist. Für die COM-Funktionen sind aber häufig Daten vom Typ LONG erforderlich.

Weiterhin ist auch die Realisierung mancher wichtiger API-Funktionen problematisch, wenn nicht unmöglich. Dazu gehören z.B. Funktionen zum Schreiben von INI-Dateien.

Schließlich sind viele Befehle umständlich zu realisieren. Aus diesem Grund werden mit der Installation von enaio® einige von OPTIMAL SYSTEMS entworfene Bibliotheken installiert und registriert. Diese sind in folgenden Bibliotheksdateien realisiert:

- § `oxactive.dll` stellt eine Reihe von Funktionen für die Ausführung von Events zur Verfügung.
- § `oxvbba.dll` ist eine Bibliothek, die umfangreiche Hilfsfunktionen für die Programmierung in VB und VB Script anbietet.
- § `oxvbscript.dll` stellt eine Reihe von COM-Funktionen zur Verfügung, die in VB Script nicht realisierbar sind.

OXACTIVE.DLL

Über das eingebundene ActiveX-Steuerelement `oxactive.dll` stehen Methoden und Objekte zur Verfügung, mit denen Daten aus der Übergabedatei leicht ausgelesen und geändert werden können. Folgende Objekte stellt das ActiveX-Steuerelement `oxactive.dll` zur Verfügung:

- § `ASFile` bietet Unterstützung im Umgang mit der AddOn- bzw. Eventübergabedatei.
- § `RequestPages` bietet eine Auswahl an Datenmasken an.

- § RequestPage repräsentiert eine Datenmaske.
- § ASFields stellt alle Felder eines Datenblatts zur Verfügung.
- § ASField repräsentiert ein Feld eines Datenblatts.

Darüber hinaus können Sie über VB Script unabhängig vom enaio® client Aktionen ausführen oder über die COM-Schnittstelle unabhängig von der Übergabedatei mit dem enaio® client kommunizieren.

OXVBAS.DLL

Diese Bibliothek bietet eine Reihe von Funktionen zur Unterstützung der Programmierung von VB und VB Script. Das oxvbbs-Objekt bietet folgende Schnittstellen:

- § Functions sind allgemeine Funktionen. Hier erhält man Informationen zum Windows-Betriebssystem, zu Dateiversionen, Dateitypen usw.
- § AS_Functions sind Funktionen der enaio® client-Schnittstelle. Hier sind unterstützende Funktionen zur Ermittlung von Objekt-, Haupt- und Untertypen von enaio®-Objekten zu finden.
- § INI_Functions sind Funktionen zur Bearbeitung von INI-Dateien. Es können einzelne oder alle Schlüssel einer Sektion ermittelt, die Werte ausgelesen und geschrieben werden.
- § Reg_Functions sind Funktionen zur Bearbeitung der Windows-Registrierung. Es können Schlüssel und deren Werte ermittelt, geschrieben und gelöscht werden.

OXVBSCRIPT.DLL

Die mit enaio® gelieferte Bibliothek `oxvbscript.dll` stellt für einige gängige enaio®-COM-Funktionen, die aus VB Script nicht zur Verfügung stehen, Äquivalente bereit:

- § InsertFileList
- § UpdateArchiveData
- § InsertIntoDocument
- § InsertIntoRegister
- § InsertIntoArchive
- § GetFilesFromIDEx

Die Funktionen binden Sie über den Aufruf `CreateObject('oxvbscript.Function')` ein. Parameter und Rückgabewerte entsprechen genau denen der COM-Funktionen. Details zur COM-Schnittstelle finden Sie in der Dokumentation zur COM-Schnittstelle.

Systemerhaltung und -Pflege

Überwachung des Anwendungsservers

Zur Überwachung des Anwendungsserver-Betriebs gibt es neben den Serverprotokollen auch die Möglichkeit, eine Server-Konsole zu öffnen. Diese Server-Konsole steht zur Verfügung, wenn der Anwendungsserver-Dienst mit lokaler Anmeldung gestartet ist, oder wenn der Anwendungsserver nicht als Dienst, sondern als reguläre Anwendung ausgeführt wird. Die Serverkonsole bietet eine Reihe von Informations- und Konfigurationsmöglichkeiten.

Im normalen Betrieb ist die Benutzung der Serverkonsole nicht erforderlich. Ihr Einsatz ist sinnvoll zur Ermittlung eventueller Störungsursachen und sollte nur durch Personal der Support- oder der Entwicklungsabteilung von OPTIMAL SYSTEMS GMBH durchgeführt werden.

Kommunikation zwischen Server und Client

Die Kommunikation zwischen Server und Client erfolgt über ein RPC ('Remote Procedure Call') Protokoll. Standardmäßig ist hier ein kompakter, performanter BIN RPC Mechanismus eingestellt. Eine Umschaltung in XML RPC ist nicht empfohlen. Zur Erhöhung der Datendurchsatzes können die Daten innerhalb des BIN oder XML RPC Mechanismus komprimiert werden. Per Default ist die Kompression abgeschaltet, um Rechnerressourcen zu sparen. Gerade bei der Anbindung von Nebenstellen über langsame Netze (ISDN), sollte die Komprimierung aktiviert werden.

Für das Erkennen von Netzwerkproblemen kann es hilfreich sein, an dieser Stelle eine Protokollierung zu aktivieren. Damit kann dann der gesamte Netzwerkverkehr zwischen Client und Server zu Analysezwecken in eine Protokolldatei geschrieben werden. Die Logdatei 'net_cln.log' liegt dann in Temp-Verzeichnis des Clients. Diese Einstellung führt zu einem erheblichen Performanceverlust auf Clientseite.

Zur Konfiguration der Kommunikation muss in das entsprechende Installationsverzeichnis eines Clients die Datei OXMLJSC.CFG bearbeitet werden.

Folgende Parameter können dort konfiguriert werden (Default-Werte sind unterstrichen):

- lognetwork=0/1 Protokollierung auf Datentransferebene (aus/an)
- binrpc=0/1 – Umschaltung BIN RPC/XML RPC
- hashed=0/1 Über die Daten des Jobaufrufs, nicht der Dateien, kann ein Hashwert gebildet und mitgesendet werden. Die Funktion ist eingeschaltet, '0' schaltet sie aus.
- packed=0/1 – Die Daten des Jobaufrufs können komprimiert werden, wenn das 'packbound'-Größenlimit überschritten ist. Es wird das gesamte RPC Packet an den Server komprimiert. Für die Übertragung von Dateien ist diese Komprimierung irrelevant. Die Funktion ist ausgeschaltet, '1' schaltet sie ein.
Wenn die komprimierte Kommunikation für den Server über enaio® enterprise-manager eingeschaltet wird, dann muss für alle Clients über die Datei OXMLJSC.CFG im jeweiligen Anwendungsverzeichnis die Komprimierung (packed=1) und das Größenlimit (packbound=0) eingestellt werden.
- packbound=300 Größenlimit für Jobgröße in Byte
- rowbase64=0 – nur BIN RPC: Mime-kodierte Parameter in Jobaufrufen (Parametertyp 6) werden direkt, also nicht Mime-kodiert übertragen
- mac=0/1/2/3 – Ermittlung der zur IP-Adresse gehörenden MAC-Adresse.
0 – Ermittlung mit Netbios, falls nicht erfolgreich, Ermittlung mit Ip-Helper
1 – Ermittlung nur mit Netbios
2 – Ermittlung mit Ip-Helper, falls nicht erfolgreich, Ermittlung mit Netbios
3 – Ermittlung nur mit Ip-Helper

In die Verzeichnisse `/client32/` und `/admin/` wird bei der Installation eine Datei `oxmljsc.cfg` mit folgendem Eintrag kopiert:

`mac=3`

Automatische Aktionen

Automatische Aktionen sind Werkzeuge, mit denen wichtige Aufgaben des Systems erfüllt werden. Dazu gehören Import und Export von Daten, Pflege des Systems und Funktionen zur Verarbeitung von Dokumenten. Außerdem gibt es Funktionen zur Steuerung automatischer Aktionen. Eine genauere Beschreibung findet man in den Dokumentationen zu den automatischen Aktionen.

Folgende automatische Aktionen stehen zur Verfügung:

- § Import und Export von Daten und Dokumentdateien
 - § ASCOLD-Import, Bibliothek `axcold.dll`. Diese Aktion dient dem Import von COLD-Daten. COLD bedeutet Computer Output on Laser Disk. Dieses Verfahren ist im Großrechnerbereich verbreitet.
 - § ASFax Import, Bibliothek `axfax.dll`. Diese Aktion erlaubt den Import von Fax-Eingängen.
 - § Daten-/Dokumentenimport, Bibliothek `axacimp.dll`. Diese Aktion ersetzt die Importfunktionalität der Aktion 'Import/Export' und erweitert sie um den Import von XML, Excel und Access. Die Konfiguration ist durch einen Assistenten vereinfacht.
 - § DICOM-Import, Bibliothek `axacdcml.dll`. Diese Aktion importiert DICOM-Dateien, die z.B. vom enaio®-DICOM-Importserver bereitgestellt werden.
 - § IMAGESTREAM Import, Bibliothek `axacimst.dll`. Diese Aktion dient dem Import von IMAGESTREAM-Daten.
 - § Daten-/Dokumentenexport, Bibliothek `axacexp.dll`. Diese Aktion ersetzt die Exportfunktionalität der Aktion 'Import/Export' und erweitert sie um den Export von XML, Excel und Access. Die Konfiguration ist durch einen Assistenten vereinfacht.
- § Systempflege
 - § Abo bereinigen, Bibliothek `axacabo.dll`. Mit dieser Funktion werden nach Kundenvorgaben Abonnement-Einträge gelöscht.
 - § Aktualisierung Datenbank, Bibliothek `axacdbst.dll`. Mit dieser Aktion wird die Datenbankstatistik aktualisiert und damit die Datenbankperformance optimiert.
 - § Bereinigung der Konfigurations- und Protokolldateien. Bibliothek `axaccl.dll`. Diese Aktion löscht nach Ihren Vorgaben Konfigurationen aus der Konfigurationshistorie und Protokolle.
 - § Cachebereinigung, Bibliothek `axaccach.dll`. Mit dieser Aktion werden Dokumentdateien im Server-Cache gelöscht. Im Server-Cache befinden sich Dokumentdateien, deren Originale entweder im WORK-Bereich einer anderen Servergruppe oder auf Archivmedien liegen.
 - § Historienpflege, Bibliothek `axachist.dll`. Mit dieser Funktion werden nach Kundenvorgaben Einträge aus der Dokumentenhistorie gelöscht.
 - § Systemprüfung, Bibliothek `axacsys.dll`. Diese Aktion prüft und korrigiert bestimmte Datenbankzustände.

- § Wiedervorlage bereinigen, Bibliothek `axacwdvl.dll`. Mit dieser Funktion werden nach Ihren Vorgaben Wiedervorlage-Einträge gelöscht.
- § Papierkorb bereinigen, Bibliothek `axaccleantrash.dll`. Mit dieser Funktion wird der Papierkorb des Systems nach Kundenvorgabe aufgeräumt.
- § Benutzerexport/-import, Bibliothek `axacuexp.dll/axacuimp.dll`. Mit dieser Funktion können die im System eingerichteten Benutzer exportiert und importiert werden. Es ist damit eine Unterstützung bei der Erstellung von Testsystemen/Spiegelsystemen möglich.
- § Dokumentverarbeitung
 - § Archivieren, Bibliothek `axacarc.dll`. Diese Aktion überträgt Dokumente auf Archivbereiche und entfernt Sie aus dem WORK Bereich.
 - § Dearchivierung, Bibliothek `axacunac.dll`. Mit dieser Aktion werden Dokumenttypen dearchiviert und in das Server-WORK Verzeichnis übertragen.
 - § Dokumentbereitstellung, Bibliothek `axacpref.dll`. Mit dieser Aktion werden Dokumente von anderen Anwendungsservern bereitgestellt.
 - § Mediendearchivierung, Bibliothek `axacunme.dll`. Mit dieser Aktion werden alle Dokumenttypen auf bestimmten Medien dearchiviert und in das Server-WORK Verzeichnis übertragen.
 - § Objektverschlüsselung, Bibliothek `axaccrypt.dll`. Mit dieser Aktion werden Dokumentdateien von DMS-Objekten verschlüsselt.
 - § Paginierung, Bibliothek `axacpage.dll`. Damit werden Dokumentseiten paginiert, d.h. ein Aufdruck erzeugt, in dem bestimmte Informationen enthalten sind.
 - § Volltextindizierung, Bibliothek `axacidx.dll`. Mit dieser Aktion werden nachträglich DMS-Objekte volltextindiziert.
 - § XML-TagExtraktion, Bibliothek `axacxmle.dll`. Mit dieser Aktion können Sie aus XML-Dateien Dateien im Delimited-ASCII-Format erzeugen.
 - § XML-Transformation, Bibliothek `axacxmlc.dll`. Diese Aktion dient der Konvertierung von XML-Dateien in andere XML-Formate.
- § Steuerung von automatischen Aktionen
 - § Aktionsfolge, in enaio® administrator integriert. In dieser Aktion werden Abfolgen von automatischen Aktionen gebildet.
 - § COM-Aktion Interface, Bibliothek `axacom.dll`. Mit dieser Aktion werden spezielle COM-Bibliotheken gekapselt und als automatische Aktion zugänglich gemacht.
 - § Externes Programm starten, in enaio® administrator integriert. Mit dieser Aktion können Sie externe Windows-Anwendungen starten.
 - § Externes Programm über Kommandozeile starten, Bibliothek `axacexec.dll`. Mit dieser Aktion können Sie externe Windows-Anwendungen und Batchdateien starten und den Returncode auswerten lassen.
 - § Synchronisation, Bibliothek `axacsync.dll`. Diese Aktion erlaubt die Synchronisation von automatischen Aktionen.

Hardware- und Software-Voraussetzungen

Zur Vorbereitung von automatischen Aktionen ist zu prüfen, ob die Voraussetzungen für den Einsatz gegeben sind. Automatische Aktionen werden im enaio® administrator eingerichtet und dort oder durch enaio® start ausgeführt. Daneben gibt es für einige Aktionen gesonderte Voraussetzungen für Hardware und Software. So ist für die dokumentenechte Archivierung ein optisches Laufwerk für nicht wieder beschreibbare Medien erforderlich. Weitere Voraussetzungen werden bei der jeweiligen automatischen Aktion genannt.

Lizenzen

Einige Aktionen können mit der Lizenz des enaio® administrator ausgeführt werden, für andere sind eigene Lizenzen erforderlich. Lizenzen können Sie bei OPTIMAL SYSTEMS GmbH erwerben. Sie werden in enaio® mit Import der Datei `aslic.dat` durch das Programm `aslicimpexp.exe` bekannt gegeben. Die Verteilung der Lizenzen auf Arbeitsstationen nehmen Sie im Lizenzsystem des enaio® administrator vor. Um automatische Aktionen mit dem enaio® administrator konfigurieren und starten zu können, ist der Lizenzstring ADM erforderlich. Außerdem ist für die automatische Ausführung mit enaio® start der Lizenzstring AXA erforderlich.

Nachfolgend eine Liste aller automatischen Aktionen mit den zugehörigen Lizenzstrings:

| Automatische Aktion | Lizenzstring |
|------------------------------------------------------|--------------|
| Abonnementpflege | ADM |
| Aktionsfolge | ADM |
| Aktualisierung Datenbank | ADM |
| Archivieren | ARC |
| ASCOLD-Import | COL |
| ASFax Import | FAX |
| Bereinigung der Konfigurations- und Protokolldateien | ADM |
| Cachebereinigung | ADM |
| COM-Aktion Interface | ADM |
| Daten-/Dokumentenimport | AIE |
| Daten-/Dokumentexport | AIE |
| Dearchivierung | ADM |
| DICOM-Import | ADM |
| Dokumentbereitstellung | ADM |
| Externes Programm starten | ADM |
| Externes Programm über Kommandozeile starten | ADM |
| Historienpflege | ADM |
| IMAGESTREAM Import | LIS |
| Mediendearchivierung | ADM |
| Objektverschlüsselung | KRY |
| Paginierung | PAG |
| Synchronisation | ADM |
| Systemprüfung | ADM |
| Volltextindizierung | ADM |
| Wiedervorlagepflege | ADM |
| XML-TagExtraktion | ADM |
| XML-Transformation | ADM |

Tabelle 45

Detaillierte Informationen zum Lizenzsystem können Sie dem Administrations-Handbuch entnehmen.

Die Installation aller automatischen Aktionen erfolgt mit Installation des enaio® administrator. Die verschiedenen Bibliotheksdateien werden in das Verzeichnis kopiert, in dem der enaio® administrator installiert ist. Eine Registrierung der Bibliotheken ist nicht erforderlich, da sie vom Administrator als Standard-API-DLLs angesprochen werden.

Mehrsprachigkeit

Bei der Herstellung von enaio® wurde streng auf das Konzept von Mehrsprachigkeit geachtet. Damit wird der Einsatz des Produkts in internationalen Betrieben und Organisationen ermöglicht. Prinzipiell wird im Produkt enaio® Mehrsprachigkeit in den Bereichen GUI und Objektdefinition unterstützt. Beide Punkte lassen sich unabhängig voneinander, je nach den Gegebenheiten beim Anwender, einstellen. Im Lieferumfang von enaio® sind sowohl Deutsch als Standardsprache als auch Englisch enthalten.

Mehrsprachigkeit der Objektdefinition

Im enaio® editor können die Indexdatenmasken mehrsprachig angelegt werden. Je nach benutzerspezifischer Einstellung im enaio® client bekommt der angemeldete Benutzer dann die Datenmasken in seiner ausgewählten Sprache präsentiert.

Mehrsprachige Dokumentation und Online-Hilfe

Dokumentation und Online-Hilfen zu enaio® sind in Deutsch und größtenteils auch in Englisch vorhanden. Diese Dokumentationen werden von OPTIMAL SYSTEMS auf Basis von MS Word Dateien erzeugt und im PDF-oder HLP-Format zur Verfügung gestellt.

Mehrsprachigkeit im Server

Je nach den Spracheinstellungen im Server werden die Fehlermeldungen und Warnungen, die vom Server an die Clients geschickt werden übersetzt. Für die Mehrsprachigkeit im Server wird der Microsoft Message Compiler verwendet. Sprachdateien im ASCII-Format können für jede Sprache angelegt und in den Server kompiliert werden.

Auch servernahe Anwendung wie enaio® enterprise-manager, enaio® protocol-viewer werden nach diesem Mechanismus internationalisiert. Die Sprachwahl für die GUIs dieser Module erfolgt über Konfigurationsdateien oder Menüeinträge.

Mehrsprachigkeit im enaio® office-utilities

Die enaio® office-utilities stehen ebenfalls mehrsprachig zur Verfügung. Der Benutzer kann über einen Menüpunkt die Sprache in der GUI für diese Module umschalten. Bei diesen Tools ist die Mehrsprachigkeit ebenfalls über das Kompilieren von Sprachressourcen in die jeweiligen Programme gelöst. Die Sprachdateien liegen ebenfalls als ASCII-Dateien vor.

Mehrsprachigkeit in Clientanwendungen

Die Basis-Sprache von enaio® ist Deutsch. Der Benutzer kann in den Benutzereinstellungen im enaio® client zwischen den Sprachen Deutsch und Englisch wechseln. Das Wechseln der Sprache im enaio® client hat für diesen angemeldeten enaio®-Benutzer Auswirkungen auf die GUI aller enaio®-Clientanwendungen, die internationalisierbar sind. Diese starten sich dann ebenfalls in der im enaio® client ausgewählten Sprache.

Startverhalten des enaio® client

Beim Starten des enaio® client werden umfangreiche Registrierungen durchgeführt, sowie diverse System- und Benutzereinstellungen vom Server neu geladen. Beim Betrieb in Netzwerken mit ungenügender Übertragungsperformance kann der Start des Programms sehr langsam sein.

Zum Ermitteln der zeitlichen Abfolge des Startvorgangs kann der enaio® client mit dem Kommandozeilenparameter '-slog' gestartet werden. In diesem Modus wird der Programmstart ausführlich protokolliert. Das entsprechende Protokoll kann im Menüpunkt geöffnet werden:


Datei > Protokollierung > Startprotokollierung ansehen


Ein Versenden des Protokolls im XML Format per E-Mail ist ebenfalls möglich.

Ein entscheidender Einfluss auf das Startverhalten des enaio® client kann über die Konfiguration des Systems genommen werden. Sehr große und umfangreiche Objektdefinitionen, ein umfassendes Rechtesystem mit vielen Objektklauseln können die Ursache für einen längeren Systemstart verursachen.

Begrenzte Optimierungsmöglichkeiten für die Performance des Programmstarts bieten folgende Möglichkeiten:

- § Abschalten der Registrierung der verwendeten Programmkomponenten beim Systemstart:
 - § Eintrag NOREGISTER=1 in der Sektion [ARCHIV] der *asinit.cfg* im Installationsverzeichnis des enaio® client.
- § Vermeiden von Bildern in der Objektdefinition
- § Verzicht auf ein großes Hintergrundbild im Desktop des enaio® client
- § Verzicht auf Sprachumstellung des enaio® client
- § Optimierung der Wiedervorlagen/Abos des Benutzers
- § Verringerung der Anzahl von gespeicherten Anfragen und externen Programmen
- § Verkleinerung der Objektdefinition
- § Verkleinerung des Rechtesystems


Protokollierung des Applikationsstarts



OS.5|ECM Startup Protokollierung

© OPTIMAL SYSTEMS GmbH

KAMP

Systeminformationen

Anwendung:

OS.5|ECM - Client

Version:

5.20.570

Workstation:

11D3-000C6ED7BC96#11D3-005056C00001#11D3-005056C00008

Server:

kamp#4000

Protokollierung:

| Datum | Zeit | Aktion | Dauer (ms) |
|-------------------|-----------------|-------------------------------------------|-----------------|
| 07.02.2006 | 12:18:11 | Initialisierung der Startprotokollierung | 4.64 |
| 07.02.2006 | 12:18:12 | Initialisierung der Protokollierung | 1052.39 |
| 07.02.2006 | 12:18:14 | Initialisierung des Applikationsfensters | 1835.70 |
| 07.02.2006 | 12:18:15 | Initialisierung weitere Bibliotheken | 476.47 |
| 07.02.2006 | 12:18:21 | Registrierung von COM-Komponenten | 6198.08 |
| 07.02.2006 | 12:18:21 | Vorbereitung der Benutzeranmeldung | 193.31 |
| 07.02.2006 | 12:18:23 | Verbindung zum Server aufgebaut | 1506.49 |
| 07.02.2006 | 12:18:23 | Benutzeranmeldung erfolgt | 626.12 |
| 07.02.2006 | 12:18:25 | Lizenzprüfung erfolgt | 1638.57 |
| 07.02.2006 | 12:18:25 | Datenkanal initialisiert | 541.49 |
| 07.02.2006 | 12:18:28 | Objektdefinition eingelesen | 2291.67 |
| 07.02.2006 | 12:18:31 | Objektdefinition initialisiert | 3581.35 |
| 07.02.2006 | 12:18:31 | Benutzerrechte geladen | 145.91 |
| 07.02.2006 | 12:18:31 | Prüfe auf Softwareupdates | 37.33 |
| 07.02.2006 | 12:18:32 | Plugins und CDL geladen | 1028.92 |
| 07.02.2006 | 12:18:34 | Workflow geladen und initialisiert | 1932.36 |
| 07.02.2006 | 12:18:36 | Sicherheitssystem initialisiert | 1701.56 |
| 07.02.2006 | 12:18:37 | Benutzereinstellungen geladen | 1250.78 |
| 07.02.2006 | 12:18:38 | Benutzereinstellungen initialisiert | 215.01 |
| 07.02.2006 | 12:18:38 | Sprachumstellung der Applikation | 13.78 |
| 07.02.2006 | 12:18:38 | Gespeicherte Anfragen geladen | 423.42 |
| 07.02.2006 | 12:18:38 | Gespeicherte ext. Programme geladen | 289.30 |
| 07.02.2006 | 12:18:38 | Laden der Objektdefinitionsbilder | 111.79 |
| 07.02.2006 | 12:18:38 | Erweiterte Anfragen geladen | 5.03 |
| 07.02.2006 | 12:18:39 | Gespeicherte Objekte geladen | 456.28 |
| 07.02.2006 | 12:18:40 | Active Documents geladen | 908.85 |
| 07.02.2006 | 12:18:41 | Anzeige aktualisiert | 1026.98 |
| 07.02.2006 | 12:18:41 | COM Schnittstelle initialisiert | 57.53 |
| 07.02.2006 | 12:18:41 | Lade Hintergrundbild | 80.90 |
| 07.02.2006 | 12:18:42 | Initialisiere Abo/Wiedervorlage/WF Listen | 1087.51 |
| 07.02.2006 | 12:18:42 | Skriptengine initialisiert | 180.35 |
| 07.02.2006 | 12:18:43 | Weitere Initialisierungen | 759.54 |
| 07.02.2006 | 12:18:43 | Gesamte Startzeit | 31667.59 |

Abbildung 57: Ansicht der Startprotokollierung des enaio® client

Beim Start des enaio® client können ebenfalls Probleme auftreten, wenn zuvor gestartete Clientprozesse nicht richtig beendet wurden. In diesem Fall können Benutzer über die Einstellung Andere Client-Prozesse bei Programmstart beenden sicherstellen, dass diese Prozesse beendet werden. Diese Einstellung kann ebenfalls über einen Eintrag in der Konfigurationsdatei `\etc\as.cfg` systemweit vorgenommen werden:

```
[CLIENT]
KILLPROCESSES=1
```

Kommandozeilenparameter des enaio® client

Normalerweise wird der enaio®-Client ohne Kommandozeilenparameter gestartet. Für besondere Anforderungen stehen folgende Kommandozeilenparameter zur Verfügung:

- § -UID: Hier kann der Name eines Benutzers angegeben werden, der automatisch vom Client angemeldet wird. Sind Name und Passwort des

Benutzers angegeben, erscheint kein Anmeldedialog zur Eingabe von Benutzernamen und Passwort beim Starten der Anwendung.

Falls Benutzernamen Leerzeichen enthalten, dann verwenden Sie -USER. Der Benutzernamen wird dann in Anführungszeichen (") gesetzt.

§ -PWD: Hier kann ein Passwort für den oben angegebenen Benutzer hinterlegt werden. Sind Name und Passwort des Benutzers angegeben, erscheint kein Anmeldedialog zur Eingabe von Benutzernamen und Passwort beim Starten der Anwendung.

Falls Passworte Leerzeichen enthalten, dann verwenden Sie --PASSWORD. Das Passwort wird dann in Anführungszeichen (") gesetzt.

§ -SRV: Hier kann festgelegt werden, mit welchem Server der Client sich verbindet. Die Syntax ist analog zu den Einträgen in der *asinit.cfg*.

§ server#port: Mit diesem Parameter wird der Eintrag eines Servers in der *asinit.cfg* ignoriert.

§ -N: Wenn dieser Kommandozeilenparameter gesetzt ist, wird der Splashscreen während des Startens unterdrückt.

§ -SLOG: Der Client protokolliert seinen Systemstart. Dieses Protokoll kann im Client eingesehen werden.

§ -I: Der Client registriert die von ihm verwendeten Programmmodule. Nach dem Registrieren der Komponenten wird der Client automatisch wieder beendet.

Beispiel: `ax.exe -uid Benutzer -pwd Passwort -srv osserver#4040`

Zusammenfassung der enaio®-Administrationstools

enaio® editor

Die Hauptfunktion von enaio® editor liegt in Ausführung folgender Aktionen:

- § Erstellung von DMS-Objekten.
- § Bearbeitung der Eingabemasken für das Neuanlegen, Ändern und Recherchieren von Objekten. Dabei werden Felder erzeugt, Feldeigenschaften eingestellt, die Tabulatorreihenfolge der Felder festgelegt.
- § Speicherung der Objektdefinition. Die Informationen zu den DMS-Objekten sind einerseits in der Objektdefinitions-Datei und andererseits in der Datenbank gespeichert. Nach Bearbeitung der Objekte ist das explizite Speichern der Änderungen erforderlich.
- § Datenbankanpassung
- § Tabellenprüfung
- § Außerdem können vorbereitete Objektdefinitionsdateien komplett oder teilweise importiert werden.

Informationen zur konkreten Ausführung der o.g. Tätigkeiten erhalten Sie im Handbuch zum enaio® editor.

enaio® administrator

Die Hauptfunktion des enaio® administrator liegt in Ausführung folgender Aktionen:

- § Einrichtung des Gesamtsystems. Hier werden diverse Systemparameter, wie z.B. das Anmeldeverhalten, Hinzufügen von Action-DLLs usw. eingestellt.
- § Einrichtung des Sicherheitssystems. Dabei werden Benutzer und Benutzergruppen angelegt. Einzelnen Benutzern können Systemrollen, d.h. Anwendungsrechte auf enaio®-Anwendungen erhalten, Benutzergruppen werden Objektrechte zugeordnet, die durch Klauseln eingeschränkt werden können. Als Vorlage für neue Benutzer können Benutzerprofile vorbereitet werden.
- § Einrichtung von Dokumenttypen. Für W-Dokumente werden Anwendungen und Vorlagen vorbereitet. Die Vorlagen werden dann den Anwendungen und Benutzern bzw. Benutzergruppen zugeordnet. Für andere Dokumenttypen wie E-Mail und XML sind bestimmte Parameter manuell in Konfigurationsdateien einzurichten.
- § Einrichtung des Archivdruckers. Hier werden Hintergrundbitmaps für Drucken von Dokumenten in das DMS eingerichtet.
- § Automatische Aktionen. Dabei werden automatische Aktionen für unterschiedliche Zwecke eingerichtet. Folgende Systemfunktionen werden von den automatischen Aktionen erfüllt:
 - § Import und Export,
 - § Systempflege,
 - § Dokumentverarbeitung,
 - § Steuerung von automatischen Aktionen.

enaio® enterprise-manager

Die Hauptfunktion von enaio® enterprise-manager liegt in Ausführung folgender Aktionen:

- § Verwaltung und Konfiguration von enaio® server.
- § Einrichtung des Lizenzsystems. Dabei werden die erworbenen Lizenzen auf die vorhandenen Arbeitsstationen verteilt.
- § Einrichtung des Archivsystems. Hier werden die Vorbereitungen getroffen, die erforderlich sind, um die Archivierung, d.h. das Schreiben von Dokumentdateien auf optische Medien, durchzuführen.
- § Ansicht auf die Tabellen der Systemdatenbank

Protokolleinrichtung

Die Einrichtung der Protokolle wird innerhalb des enaio® manager-for-logfiles (axprotocolcfg.exe) vorgenommen. Dazu liegen in den jeweiligen Installationsverzeichnissen von Server und Client-Komponenten Konfigurationsdateien.

Über detaillierte Einstellungsmöglichkeiten kann eine genaue Protokollierung bis hin zu einer arbeitsplatz- und komponentenspezifischen Protokollierung eingestellt werden.

enaio® editor-for-events

Im Eventeditor, der in den enaio® client integriert ist, werden Skripte in VB Script erstellt und aktiviert, die bei bestimmten Aktionen im enaio® client ausgeführt werden. Die Einsprungspunkte für die Skripte sind Anwendungsereignisse wie die

Anmeldung von Benutzern, oder dokumentbezogene Aktionen wie z.B. das Öffnen eines Datenblatts.

enaio® capture im Administrationsmodus

Zur Vorbereitung des Produktivbetriebs von enaio® capture sind im Administrationsmodus folgende Tätigkeiten durchzuführen:

- § Erstellung und Bearbeitung von Konfigurationen. Dabei werden Name, Batchbezeichnung und Zugriffsberechtigungen für die Konfiguration festgelegt.
- § Erstellung von Subprogrammtypen. Einzelne enaio®-Subprogrammdateien werden für die Einbindung in Konfigurationen vorbereitet.
- § Hinzufügen von Subprogrammen zur Konfiguration.
- § Konfiguration der Subprogramme. Dabei werden Import- und Festfelder definiert, die Seitentrennung festgelegt, eine ODBC-Datenquelle für die Batchdatenbank ausgewählt und die Zuordnung der Importfelder zu DMS-Objektfeldern vorgenommen.

Serverkonsole

Wenn ein Anwendungsserver in einem bestimmten Modus gestartet ist, steht zur Überwachung des Anwendungsserver-Betriebs eine Server-Konsole zur Verfügung. Die Serverkonsole bietet eine Reihe von Informations- und Konfigurationsmöglichkeiten.

Transport von enaio® Systemeinstellungen

Hier werden Vorgehensweisen beim Transport von grundlegenden Konfigurationen zwischen einem enaio® Testsystem und einem Produktivsystem erläutert.

Insbesondere in Umgebungen, in denen der Workflow, Events und komplexe Datenimporte eingesetzt werden, empfehlen wir den Aufbau eines Testsystems. Dort können Sie Konfigurationen entwickeln und überprüfen, bevor Sie diese in das Produktivsystem übertragen.

Aufbau eines Testsystems

Für den Aufbau eines Testsystems benötigen Sie eine Testlizenz. Testlizenzen erhalten Sie auf Anfrage von OPTIMAL SYSTEMS oder von den Vertriebspartnern. Die Gültigkeit einer Testlizenz ist in der Regel auf fünf Tage begrenzt, kann aber auch ausgedehnt werden.

Test- und Produktivsystem sollten weitgehend identisch sein, um aussagekräftige Tests zu ermöglichen und den Transport von Konfigurationen zu erleichtern. Die Vorgehensweisen beim Transport von Konfigurationen zwischen einem enaio® Testsystem und einem Produktivsystem einerseits und umgekehrt zwischen einem Produktivsystem und einem Testsystem entsprechen sich. Der Aufbau eines Testsystems ist deshalb im Folgenden nur kurz skizziert.

Installation des Testsystems

Für die Installation eines Testsystems sollten folgende Voraussetzungen erfüllt sein:

- § Betriebssystem

Der enaio® server sollte im Test- und Produktivsystem auf dem gleichen Betriebssystem mit gleicher Spracheinstellung laufen.

§ Datenbank

Test- und Produktivsystem sollten das gleiche Datenbanksystem verwenden.

§ Installationspfade

enaio® und alle Komponenten sollten, relativ zum Rechner, in der gleichen Pfadstruktur installiert werden.

§ enaio® Version

Test- und Produktivsystem sollten die gleiche Version haben.

§ Dienstname

Test- und Produktivsystem sollten den gleichen Dienstnamen haben.
Dadurch wird der Transport von Registrierungseinstellungen erleichtert.
Den Dienstnamen geben Sie an, wenn Sie die Testlizenz anfordern.

Die Installation von enaio® ist im Installationshandbuch dokumentiert.

Konfigurieren des Testsystems

Ist Ihr Produktivsystem bereits in Betrieb, können Sie die folgenden grundlegenden Daten aus dem Produktivsystem exportieren und im Testsystem importieren:

§ Objektdefinition (vgl. Objektdefinition)

Die Objektdefinition exportieren Sie mit dem enaio® editor. Die Exportdatei importieren Sie ebenfalls mit dem enaio® editor in das Testsystem und passen die Datenbanktabellen an.

§ Benutzerdaten und Zugriffsrechte (vgl. Benutzerdaten und Zugriffsrechte)

Sowohl die Benutzerdaten wie auch die Zugriffsrechte auf Archivobjekte exportieren Sie aus dem Produktivsystem über die Automatische Aktion 'Benutzer - und Gruppenexport'. Diese Daten importieren Sie dann über die Aktion 'Benutzer - und Gruppenimport' in das Testsystem.

§ W-Vorlagen (vgl. W-Vorlagen)

Die W-Vorlagen exportieren und importieren Sie über den enaio® administrator.

Weitere grundlegende Konfigurationen eines Systems erfolgen über den enaio® enterprise-manager und den enaio® administrator.

Der enaio® enterprise-manager (vgl. Registrierungseinträge) schreibt die Konfigurationsdaten in die Registrierung und in die Datenbank. Alle Daten, die in der Datenbank verwaltet werden, können nicht transportiert werden. In der Datenbank werden beispielsweise die Daten der Medienverwaltung und die Lizenzeinstellungen gespeichert.

Registrierungsdaten können über den Registrierungs-Editor exportiert und importiert werden. Diese Daten enthalten in der Regel Pfade, die überprüft werden sollten.

Der enaio® administrator (vgl. Konfigurationseinträge in der as.cfg) schreibt Konfigurationsdaten sowohl in die Registrierung wie auch in die Konfigurationsdatei `as.cfg`. Diese Konfigurationsdatei kann transportiert werden, wenn Testsystem und Produktivsystem weitgehend identisch sind.

Objektdefinition

Im enaio® editor werden die ECM-Objekte - Ordner-, Register- und Dokumenttypen - definiert. Die Namen und interne Namen der Objekte und

Objektelemente werden dort angegeben. Automatisch erhalten sowohl die Objekte wie auch die Objektelemente eine eindeutige OS-GUID. Bei der Übertragung der Objektdefinition muss diese OS-GUID mit übertragen werden. Die OS-GUID wird im enaio® editor nicht angezeigt und kann nur der Objektdefinitionsdatei oder der Datenbank entnommen werden.

Öffnen Sie im enaio® editor eine Objektdefinitionsdatei und übernehmen daraus Objekte, wird die OS-GUID aus der Objektdefinitionsdatei nicht übernommen sondern eine neue OS-GUID vergeben. Da einige Funktionen auf die OS-GUID Bezug nehmen, muss sichergestellt werden, dass im Produktiv- und im Testsystem auch die OS-GUID der Objekte und Objektelemente identisch ist.

Das ist nur der Fall, wenn Sie im enaio® editor die Objektdefinition als Datei speichern und die Datei im Zielsystem importieren. Beim Import werden alle Daten aus der Objektdefinitionsdatei übernommen und in die Datenbank übertragen. Dabei werden die bestehenden Objektdefinitionsdaten überschrieben.

Einige Daten werden nicht in der Objektdefinition gespeichert und müssen überprüft werden:

§ Strukturbaumkatalog

Die Objektdefinition enthält Pfad und Bezeichnung der Strukturbaumdatei für den Strukturbaumkatalog. Die Strukturbaumdatei selbst müssen Sie in das Zielsystem kopieren und gegebenenfalls über den enaio® editor im Zielsystem den Pfad anpassen. Für den enaio® webclient müssen Sie über eine Konfigurationsdatei auf eine Kopie der Strukturbaumdatei beim Webserver verweisen.

§ Adress-AddOn

Die Konfiguration für das Adress-AddOn wird in einer Konfigurationsdatei im Ordner `etc` im Datenverzeichnis gespeichert. Die Datei müssen Sie in das entsprechende Verzeichnis des Zielsystems kopieren. Sie ist nach folgendem Schema bezeichnet:

```
axadaddress_'OS-GUID'.xml
```

§ Katalog-AddOn

Die Konfiguration für das Katalog-AddOn wird ebenfalls im Ordner `etc` im Datenverzeichnis gespeichert und muss in das entsprechende Verzeichnis des Zielsystems kopiert werden:

```
axaddxmlltree_'OS-GUID'.xml
```

§ VB-Script-AddOn

Die Objektdefinition enthält Pfad und Bezeichnung des Scripts für das VB-Script-AddOn. Das Script selbst müssen Sie in das Zielsystem kopieren und gegebenenfalls über den enaio® editor im Zielsystem den Pfad anpassen. Auch das Script kann kontextabhängige Bezüge enthalten und muss überprüft werden.

§ Anwendung-AddOn

Die Objektdefinition enthält Pfad und Bezeichnung der Anwendung. Überprüfen Sie, ob im Zielsystem die Anwendung so erreichbar ist.

§ Umrechnungs-AddOn

Das Umrechnungs-AddOn verwendet einen Standard-Umrechnungsfaktor. Wurde der Faktor über einen Eintrag in der Datei `as.cfg` geändert, muss diese Änderung auch im Zielsystem vorgenommen werden.

§ Benutzer-AddOn und Rechtegruppen-AddOn

Zusätzliche Konfigurationen für diese AddOns sind über Einträge in der Datei `as.cfg` möglich.

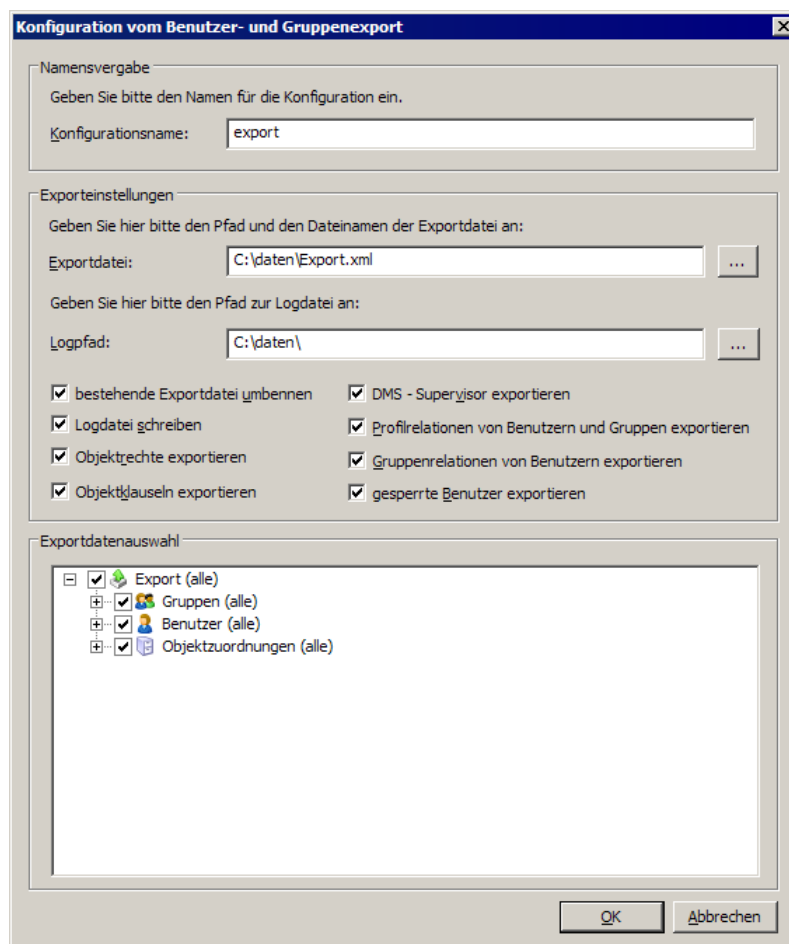
Grafiken, die auf einer Maske eingebunden sind, sowie benutzerspezifische Icons für Objekte oder Listenkataloge werden in der Objektdefinitionsdatei gespeichert und mit übernommen.

Benutzerdaten und Zugriffsrechte

Benutzerdaten, Gruppendaten und gruppenspezifische Zugriffsrechte auf Objekttypen können exportiert und importiert werden.

Ein Export und Import von Benutzer- und Gruppendaten ist nicht möglich, wenn Sie Bereiche für die dezentrale Benutzeradministration eingerichtet haben.

Export und Import führen Sie über automatische Aktionen im enaio® administrator aus.



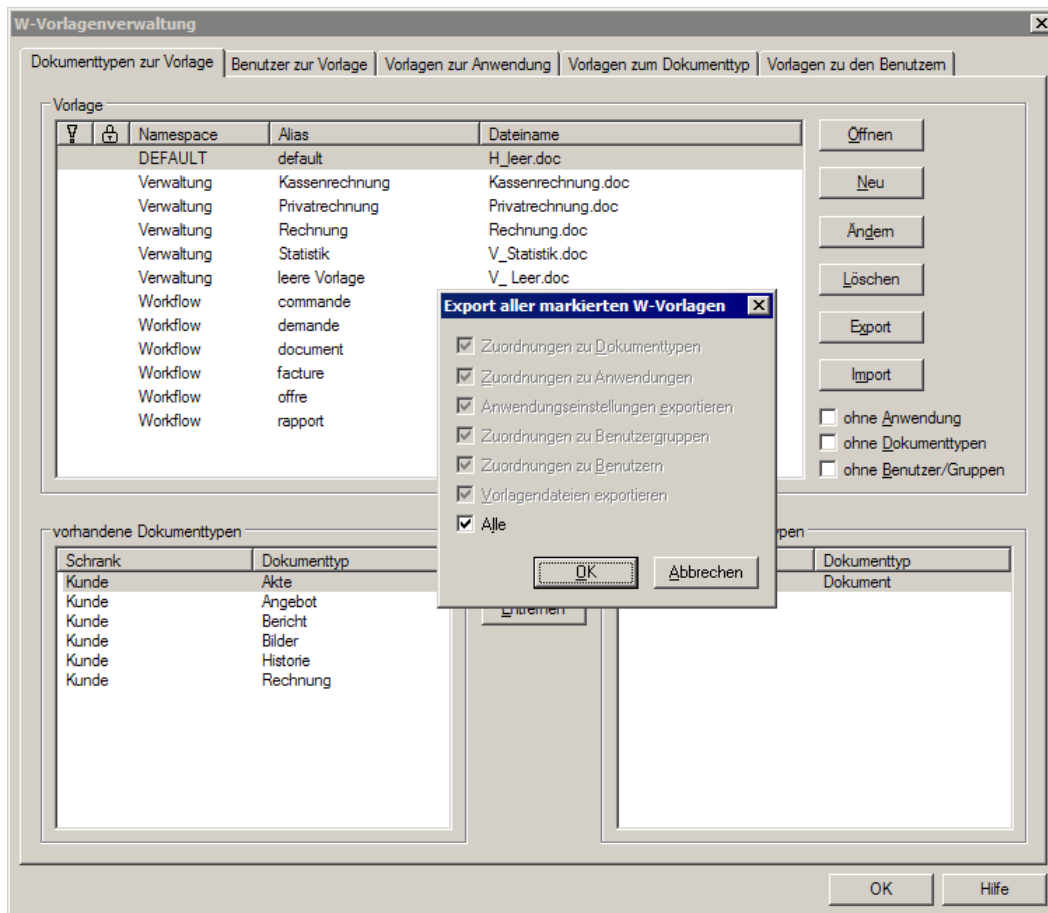
Über die Aktion 'Benutzer - und Gruppenexport' exportieren Sie die Daten. Bei der Konfiguration sollten Sie alle Daten einschließen. Mit der Aktion 'Benutzer - und Gruppenimport' importieren Sie die exportierten Daten in das Zielsystem. Bei der Konfiguration sollten Sie wiederum alle Daten einschließen.

Die Aktionen sind im Handbuch 'enaio® administrator' dokumentiert.

W-Vorlagen

Im enaio® administrator werden W-Dokumenttypen Vorlagendateien zugeordnet. Die Zuordnung ist in ein Rechtssystem eingebunden.

Sowohl die Zuordnungen wie auch die Vorlagedateien können über die W-Vorlagenverwaltung im enaio® administrator exportiert und im Zielsystem importiert werden.

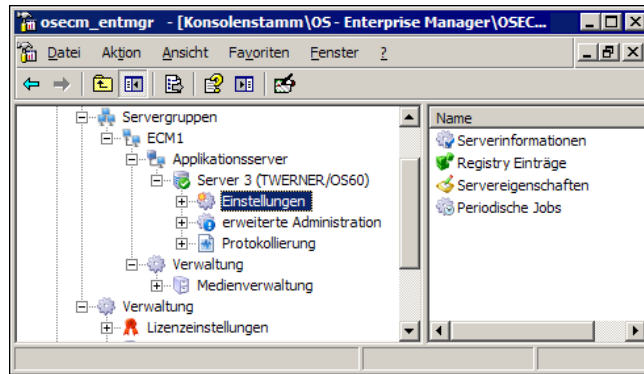


Sowohl für den Export als auch für den Import sollten Sie alle Daten einschließen. Die zugeordneten Vorlagedateien werden beim Export in das angegebene Verzeichnis kopiert, die Zuordnungsdaten in eine XML-Datei geschrieben. Beim Import werden die Vorlagedateien in das Verzeichnis `\etc\Templates` kopiert und die Zuordnungsdaten in die Datenbank übertragen.

Registrierungseinträge

Sowohl der enaio® enterprise-manager wie auch der enaio® administrator speichern Konfigurationsdaten für den enaio® server in der Registrierung des Serverrechners.

Der enaio® enterprise-manager speichert die Daten aus dem Bereich 'Einstellungen' in der Registrierung.



Die Daten der Medienverwaltung und die Daten der Lizenzeinstellungen werden in der Datenbank verwaltet. Diese Daten können nicht transportiert werden.

Die Daten aus dem Bereich 'Einstellungen' können über den Registrierungs-Editor exportiert und auf dem Zielsystem in die Registrierung importiert werden.

Folgende Teilstruktur muss exportiert werden:

```
HKEY_LOCAL_MACHINE\SOFTWARE\OPTIMAL SYSTEMS\<Dienstname>\
```

Haben Produkktivsystem und Testsystem den gleichen Dienstnamen, können die exportierten Registrierungsdaten direkt in das Zielsystem importiert werden. Sind die Dienstnamen nicht gleich, müssen Sie vor dem Import den Dienstnamen in der Registrierungsdatei anpassen.

Die Registrierungsdaten können viele kontextabhängige Daten enthalten, beispielsweise Pfade zu Verzeichnissen und URLs. Diese Pfade sollten Sie nach dem Import mit dem enaio® enterprise-manager im Zielsystem überprüfen.

Alle Registrierungsdaten, die im enaio® administrator bearbeitet werden, können ebenfalls im enaio® enterprise-manager bearbeitet werden und werden damit auch beim Transport der Teilstruktur übertragen.

Konfigurationseinträge in der `as.cfg`

Konfigurationen, die im enaio® administrator vorgenommen werden, werden in der Registrierung, in der Datenbank oder in der Datei `as.cfg` im Datenverzeichnis etc gespeichert.

In die `as.cfg` schreibt der enaio® administrator fast alle Einstellungen für das Gesamtsystem, die auf den Registerkarten Allgemeine enaio®-Konfiguration vorgenommen werden.

Die Daten in der Datei `as.cfg` haben vielfältige Bezüge zu Daten, die an anderer Stelle verwaltet werden. Bevor Sie diese Datei übertragen, sollten Sie folgende Konfigurationsdaten überprüfen:

§ Dienstname und Port

Dem Parameter `comsting` sind Dienstname und Port zugeordnet. Diese Daten müssen angepasst werden, falls sie nicht übereinstimmen.

§ Automatisch Aktionen

Eingebundene Bibliotheken für die Automatischen Aktionen über die Registerkarte Zusätze, Standardbenutzer samt verschlüsseltem Passwort und die Konfigurationsbezeichnungen der Automatischen Aktionen. Die Konfigurationsdaten selbst werden in der Datenbank gespeichert und können nicht transportiert werden.

Enthält die Datei Konfigurationen, deren Daten nicht in der Datenbank vorhanden sind, müssen Sie die Automatische Aktion nachkonfigurieren.

Enthält die Datenbank Daten von Konfigurationen, die nicht in der Datei eingetragen sind, kann auf diese Daten nicht zugegriffen werden.

Die Konfigurationsdaten der Automatischen Aktionen für den Import und Export von Daten können exportiert und importiert werden.

§ Druckkennzeichnung

Konfiguration der Druckkennzeichnung über die Registerkarte Druckkennzeichnung.

§ Archivdruck

Dateiformat und Pfad und Bezeichnung der Hintergrundbilder für den Archivdruck mit dem enaio® printer. Die Bilder müssen gegebenenfalls in das Zielsystem übertragen werden.

§ Webverzeichnis

Konfigurationen über die Registerkarte Webverzeichnis.

§ Notizen

Konfigurationen über die Registerkarte Notizen.

§ Eventverwaltung

Konfigurationen über die Registerkarte Events.

§ Hintergrundbild für den enaio® client

Einbinden eines Hintergrundbildes über die Registerkarte Start. Das Bild wird durch den enaio® administrator in `background.bmp` umbenannt und in den Ordner `etc` im Datenverzeichnis kopiert. Das Bild muss gegebenenfalls in das Zielsystem übertragen werden.

§ E-Mailverwaltung

Konfigurationen für die Übernahme von E-Mails aus einem angebundenen E-Mailsystem (vgl. Kapitel 'E-Mail-Verwaltung' im Handbuch 'enaio® Administration').

§ Überprüfung auf identische Dokumente

Die Überprüfung auf identische Dokumente wird durch einen Eintrag in der Datei `as.cfg` eingeschaltet.

§ Datenbank

Alle Konfigurationen über die Registerkarte Datenbank.

§ Dokumente

Alle Konfigurationen über die Registerkarte Dokumente.

§ automatische Aktualisierung

Über die Registerkarte Start wird die automatische Aktualisierung eingeschaltet, über weitere Einträge in der Datei `as.cfg` kann diese Aktualisierung auf dort angegebene Benutzer eingeschränkt werden.

§ Löschen von Varianten

Das Verhalten beim Löschen von Varianten wird durch einen Eintrag in der Datei `as.cfg` festgelegt.

§ Stylesheets für XML-Dokumenttypen

Zuordnung von Stylesheets zu XML-Dokumenttypen. Die Stylesheets müssen gegebenenfalls in das Zielsystem übertragen werden.

§ Signaturmethode

Die Signaturmethode wird durch einen Eintrag in der Datei `as.cfg` festgelegt.

Die Konfigurationen für die Elektronische Signatur aus dem enaio® administrator werden in der Datenbank gespeichert und müssen im Zielsystem nachkonfiguriert werden.

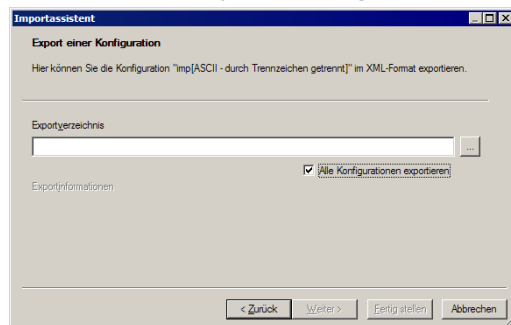
Diese Liste ist nicht vollständig. Weitere Daten für optionale Komponenten werden in der Datei `as.cfg` gespeichert.

Automatische Aktionen zum Datenimport und Datenexport

Import- und Exportkonfigurationen der automatischen Aktionen können exportiert und im Zielsystem importiert werden.

Die Automatischen Aktionen müssen in jedem System über den enaio® administrator, Registerkarte Zusätze, eingebunden werden. Über den Dialog Automatische Aktionen konfigurieren Sie eine bestehende Import- oder Exportkonfiguration oder Sie fügen eine entsprechende Aktion neu hinzu.

Im ersten Konfigurationsschritt wählen Sie dann die Funktion Konfiguration in XML-Datei exportieren und geben folgend ein Verzeichnis an, in die die Konfigurationsdaten exportiert werden sollen und markieren Alle Konfigurationen exportieren.



Diese Dateien importieren Sie dann in das Zielsystem.

Im Zielsystem konfigurieren Sie eine bestehende Import- oder Exportkonfiguration oder fügen eine entsprechende Aktion neu hinzu.

Im ersten Konfigurationsschritt wählen Sie die Funktion Konfiguration aus XML-Datei oder aus `asimpexp.cfg` importieren und geben folgend eine Datei an. Es kann nur jeweils eine Datei importiert werden. Die Konfiguration aus der Datei wird übernommen und kann bearbeitet werden.

Besteht bereits eine Konfiguration mit gleicher Bezeichnung, müssen Sie diese vor dem Import löschen.

enaio® capture und COLD-Importe

Legen Sie im benutzerspezifischen Schlüssel `HKEY_CURRENT_USER\Software\Optimal Systems\Settings` den DWORD-Wert `ConfigExportCaptureCOLD` mit dem Wert 1 an, werden beim Export aller Konfigurationen ebenfalls die Konfigurationsdaten exportiert, die über den Importassistenten für enaio® capture und COLD-Importe erstellt werden. Importieren Sie diese Daten im Zielsystem, stehen sie für die Konfiguration von Importen mit gleicher Bezeichnung in enaio® capture bzw. der Automatischen Aktion 'COLD-Import' zur Verfügung.

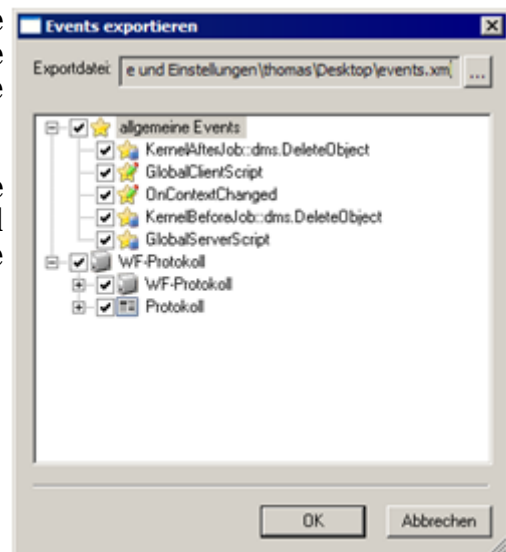
Beachten Sie bitte, dass eine Plausibilitätskontrolle nicht stattfindet.

Events

Events können über den enaio® client aus einem System exportiert und im Zielsystem importiert werden.

Markieren Sie in der Objektsuche Ihren Benutzernamen und wählen Sie aus dem Kontextmenü die Funktionen Eventexport.

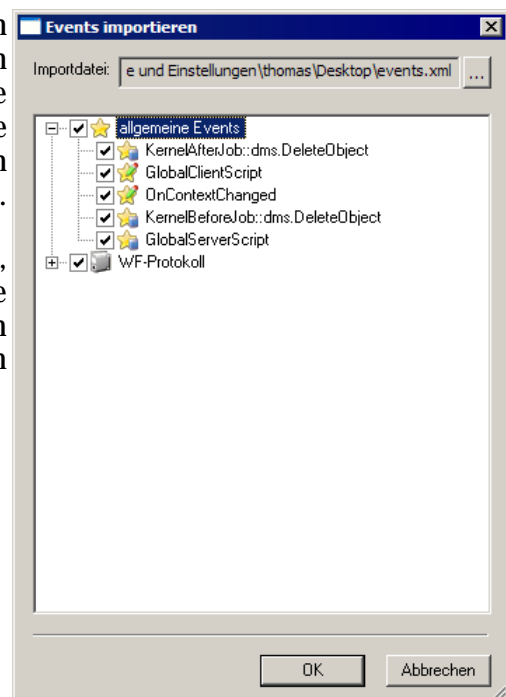
Sie markieren die Events, die exportiert werden sollen und bestätigen Ihre Auswahl über die Schaltfläche OK.



Der Eventexport erzeugt mehrere Dateien: In einer XML-Datei werden die Zuordnungen von Objekttyp und Script gespeichert. Die eigentlichen Skripte werden einzeln und verschlüsselt in das gleiche Verzeichnis als 'evc'-Dateien exportiert.

Beim analogen Eventimport wählen Sie die XML-Datei mit den Zuordnungen und geben an, welche Events importiert werden sollen. Die Skripte müssen im gleichen Verzeichnis liegen wie die XML-Datei.

Sie erhalten entsprechende Hinweise, falls den Objekttypen bereits Skripte zugeordnet sind und können entscheiden, ob diese überschrieben werden sollen.



Einzelne Skripte können ebenfalls aus dem Eventeditor exportiert und importiert werden.

Workflow

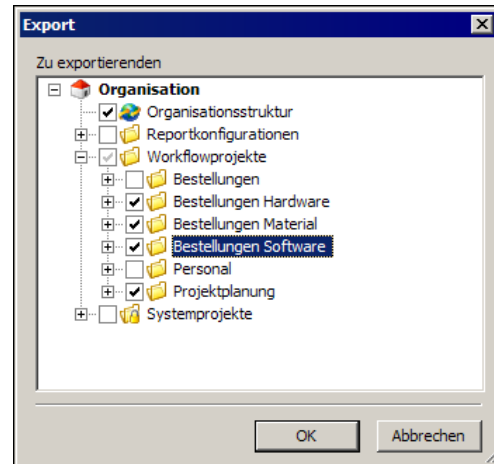
Die Daten für Workflowprozesse bestehen aus Organisationsdaten und Modelldaten. Beide sind aufeinander bezogen. Für den Transport von

Workflowprozessen exportieren Sie sowohl die Daten der Organisationsstruktur wie auch die Modelle.

Im enaio® editor_for Workflow markieren Sie im Arbeitsbereich die gewünschte Organisation und wählen aus dem Menü Datei die Funktion Organisation exportieren.

Im Exportdialog markieren Sie die gewünschten Daten.

Danach geben Sie Pfad und Bezeichnung für die Exportdatei an.



Diese Datei importieren Sie analog in das Zielsystem.

enaio® webclient / Webanwendungen

Der enaio® webclient wird über die Administrationskonsole konfiguriert. Die Konfigurationsdaten werden in der Datei `osweb.properties` im Verzeichnis `\webapps\osweb\WEB-INF\conf\` gespeichert.

Diese Konfigurationsdatei kann in das Zielsystem übertragen werden. Die Datei enthält die Verbindungsparameter zum enaio® server. Diese Parameter müssen angepasst werden.

Analog können die Konfigurationsdateien anderer enaio®-Komponenten übertragen werden, die als Webanwendung eingebunden sind: enaio® documentviewer, enaio® contentfeeder und enaio® feedreader, enaio® fulltext.

Die Daten enthalten ebenfalls Verbindungsparameter, die Sie anpassen müssen.

Protokollierung

Mit der enaio® Protokollierung können Sie für jede OS-Komponente, jeden Arbeitsplatz und jeden Benutzer eine differenzierte Protokollierung einrichten. Die Konfigurationen werden im jeweiligen Arbeitsverzeichnis in Konfigurationsdateien mit der Bezeichnung `oxrpt.cfg` gespeichert.

Diese Konfigurationsdatei kann in das entsprechende Verzeichnis des Zielsystems übertragen werden. Pfade für die Protokolle müssen gegebenenfalls angepasst werden.

SQL-Anfragen

Die Daten der SQL-Anfragen werden in der Datenbank gespeichert und können nicht übertragen werden. SQL-Anfragen müssen also im Zielsystem neu erstellt werden. Dabei können SQL-Befehl, Konfiguration der Kopfzeile und Skripte kopiert und in den Konfigurationsdialog des Zielsystems übertragen werden.

enaio® document-storage

Konfiguration für enaio® document-storage werden im Ordner `etc` im Datenverzeichnis in der Datei `axvbdocstorage.xml` gespeichert. Diese Konfigurationsdatei kann in das Zielsystem übertragen werden.

enaio® capture

Konfigurationen für enaio® capture werden sowohl in der Datenbank wie auch in der Konfigurationsdatei `axindex.cfg` im Verzeichnis `\clients\asindex` und der Konfigurationsdatei `asform.cfg` im Ordner `etc` im Datenverzeichnis gespeichert. Diese Konfigurationen können nicht transportiert werden.

Protokollierung

Protokollierung - Einführung

Mit der enaio® Protokollierung können Sie für jede einzelne OS-Komponente eine differenzierte Protokollierung einrichten.

Jeder Meldung aus einer Komponente ist mindestens eine Aktions-ID zugeordnet. Über diese Aktions-ID unterscheidet die Protokollierung Flow-, SQL-, Fehler- und Log-Meldungen, die in unterschiedliche Kanäle gelenkt und in unterschiedlichen Dateien gespeichert werden können.

Ein Kanal nimmt Meldungen mit gleicher Aktions-ID aus einer oder mehreren Komponenten auf und speichert die Meldungen im internen Protokollformat oder im OXMISC-Formate. Für jeden Kanal kann der Level der Protokollierung eingestellt werden.

Über detaillierte Einstellungen der Kanaleigenschaften kann eine arbeitsplatz- und komponentenspezifische Protokollierung eingestellt werden.

Die Konfiguration erstellen Sie mit dem enaio® manager-for-logfiles `axprotocolcfg.exe`. Die Konfigurationen werden in Konfigurationsdateien mit der Bezeichnung `oxrpt.cfg` gespeichert. Jede Komponente protokolliert über die Einstellungen aus der Konfigurationsdatei im Anwendungsverzeichnis.

Über periodische Jobs können Kanäle ein- und ausgeschaltet und der Level geändert werden.

Mit dem enaio® protocol-viewer können die Protokolle im internen Protokollformat geöffnet und sehr differenziert aufgeschlüsselt werden. Meldungen können beispielsweise nach Benutzer, Session, Thread und Rechner sortiert werden. Filter und Suchmöglichkeiten erlauben ein schnelles Lokalisieren von Meldungen.

Neben dieser Protokollierung werden weitere Protokolle erzeugt:

- § Der enaio® server schreibt automatisch Protokolldateien beim Starten und beim Beenden.
- § Über den enaio® enterprise-manager kann eine Zugriffsprotokollierung eingeschaltet werden. Protokolliert werden dann alle Datenzugriffe und alle Anfragen.
- § Für das Archivieren können zusätzliche Protokollierungen eingeschaltet werden.
- § Einige automatische Aktionen protokollieren zusätzlich zu den Protokolleinstellungen.

Bei der Installation von enaio® wird sowohl für die Clients wie auch für den Server ein Protokollpfad angegeben. Dieser Pfad wird in die jeweiligen Konfigurationsdateien eingetragen.

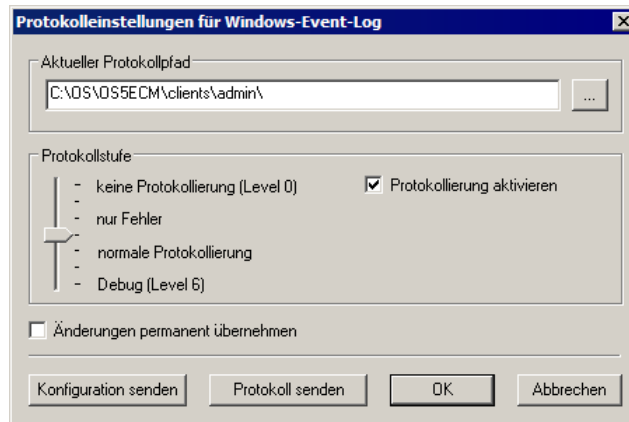
In der Server-Konfigurationsdatei sind vier Kanäle vordefiniert, jeweils ein Kanal für die Flow-, SQL-, Fehler- und ob-Call-Meldungen mit folgenden Bezeichnungen:

```
flwddmmyy.evn  
sqlddmmyy.evn  
logddmmyy.evn
```

errddmmyy.evn

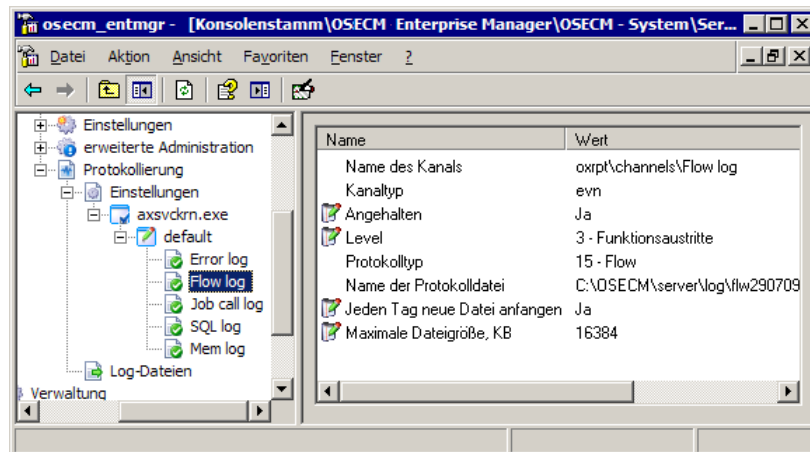
In den Client-Konfigurationsdateien ist ein Kanal für die Flow-Meldungen vordefiniert.

Auf die Einstellungen für diesen Kanal kann aus den Clients zugegriffen werden:



Jeder Benutzer kann für die laufende Programmausführung die Daten ändern. Benutzer mit der Systemrolle 'Konfiguration Gesamtsystem' können die Änderung dauerhaft übernehmen.

Für den Server können über den enaio® enterprise-manager Protokolleinstellungen für die laufende Ausführung geändert werden.



enaio® manager-for-logfiles

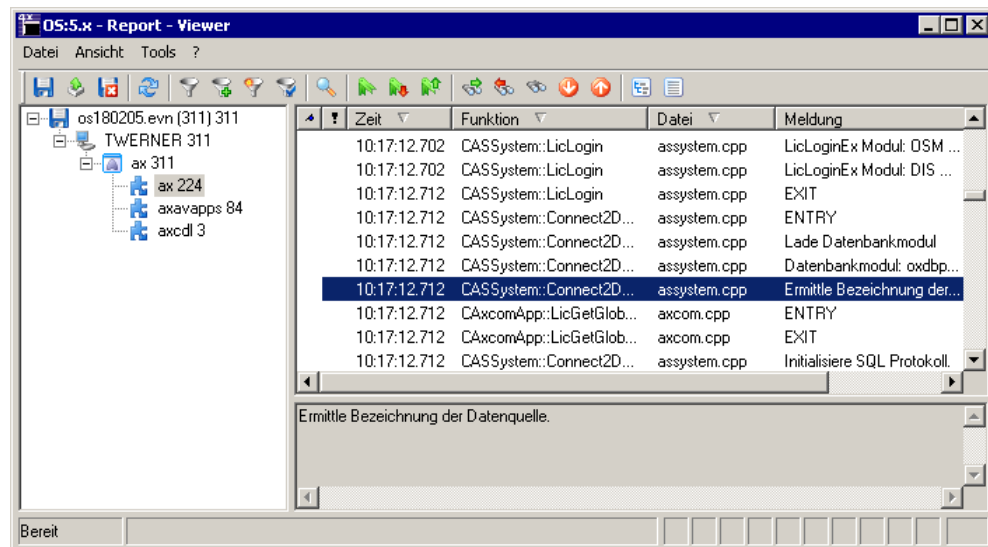
Bei der Installation von enaio® wird die Protokollierungs-Bibliothek `oxrpt.dll`, die Konfigurationsdatei `oxrpt.cfg` und der enaio® manager-for-logfiles, `axprotocolcfg.exe`, in jedes Verzeichnis mit OS-Komponenten installiert.

Mit dem enaio® manager-for-logfiles öffnen Sie die Konfigurationsdatei und können die Einstellungen bearbeiten.

Jede Komponente verwendet beim Start die Einstellungen aus der Konfigurationsdatei im Anwendungsverzeichnis.

enaio® protocol-viewer

Protokolle im internen Protokollformat (*.evn) öffnen Sie mit dem enaio® protocol-viewer. Das Programm `axrptview.exe` finden Sie in jedem Verzeichnis mit Komponenten.



Sowohl für den Navigationsbereich links wie auch für den Ereignisbereich rechts sind unterschiedliche Einstellungen möglich, um die Meldungen übersichtlich aufzulisten.

COM-Schnittstelle

Überblick

enaio® stellt eine COM-Schnittstelle zur Kommunikation mit dem Client zur Verfügung. Diese Schnittstelle besteht seit optimal_AS Version 3.x. In den frühen Versionen von optimal_AS 3.x bestand die COM-Schnittstelle parallel zur DDE-Schnittstelle, davor war die Kommunikation mit dem optimal_AS-Client nur über DDE möglich.

COM steht für 'Component Object Model' und ist ein von Microsoft eingeführtes Modell zur Kommunikation von Windowsanwendungen. Häufig wird dafür auch der Begriff 'OLE' verwendet, der für 'Object Linking and Embedding' steht. In diesem Kapitel wird ausschließlich der Begriff 'COM' verwendet.

Die COM-Schnittstelle von enaio® client hat mehrere Einsatzbereiche. Zum einen ist es aus anderen Anwendungen heraus möglich, auf alle DMS-Objekte des enaio® client zuzugreifen. Damit können z.B. Anfragen ausgeführt werden, die Objekt-Trefferlisten zurückgeben. Es ist auch möglich, neue Objekte im enaio®-System zu erzeugen. Andererseits lässt sich der enaio® client über die COM-Schnittstelle in begrenztem Maße steuern. Es lassen sich Fenster schließen, der enaio® client aktivieren u. ä. Die Möglichkeiten der Anwendungssteuerung sind jedoch nicht so umfangreich wie z.B. bei MS Word.

Genaue Informationen finden Sie in der Technischen Information 'COM-Schnittstelle von enaio® client'.

Interfaces der COM-Schnittstelle

Die COM-Schnittstelle bietet ihre Interfaces über die Type Library `ax.tlb` an.

Folgende Interfaces sind vorhanden:

- § Application. Dieses ist das umfangreichste Interface mit unterschiedlichsten Funktionen
- § Application2. Dieses Interface stellt zusätzliche Funktionen zur Anwendungssteuerung und für erweiterte Informationen zur Verfügung.
- § Window. Dieses Interface stellt Funktionen zur Steuerung einzelner Fenster im enaio® client zur Verfügung.
- § Windows. Dieses Interface stellt drei Funktionen zur Bearbeitung der Sammlung aller enaio® client-Fenster zur Verfügung.

Das Interface, das die meisten Funktionen zur Verfügung stellt, ist 'Application'. Die COM-Funktionen des Application-Interfaces können in folgende Funktionsbereiche gegliedert werden.

- § Objektbezogene Befehle,
- § Einfügen, Aktualisierung und Löschen von Objekten,

- § Dokument-Verarbeitung,
- § Anwendungskontrolle,
- § Information,
- § Konvertierung und Dateierzeugung.

Eine Zuordnung von COM-Funktionen zum jeweiligen Funktionsbereich erfolgt weiter unten im Abschnitt 'Beschreibung der COM-Interfaces'.

COM-Schnittstelle und enaio®-Sicherheitssystem

Der Zugriff auf DMS-Objekte des enaio®-Systems ist durch das Sicherheitssystem geregelt. Die COM-Schnittstelle kann diese Objektrechte beachten oder ignorieren. Wie sich die dabei COM-Schnittstelle verhält, wird durch die Eigenschaft 'Sicherheitssystem beachten' jedes Objekttyps festgelegt, die im enaio® editor eingestellt werden kann.

Die COM-Schnittstelle kann alle Objekttypen ansprechen, die der am enaio® client angemeldete Benutzer sehen kann. Zusätzlich kann sie alle Objekte ansprechen, deren Eigenschaft 'Sicherheitssystem beachten' den Wert 'Nein' hat. Bei Objekten, die der angemeldete Benutzer sehen kann und bei denen das Sicherheitssystem nicht beachtet wird, werden die für den angemeldeten Benutzer eingestellten Rechtebeschränkungen ignoriert.

Wird der Wert der Eigenschaft 'Sicherheitssystem beachten' eines Objekttyps auf 'Nein' gesetzt, dann können Benutzer z. B. mit den Übernahmemaakros von Word Werte von Objekten ermitteln, auf die sie sonst keinen Zugriff haben. Dazu ist aber die Kenntnis der Feldnamen von Objekten erforderlich. Diese Feldnamen zu ermitteln ist nicht trivial, aber mit Programmierkenntnissen möglich. Aus diesem Grund sollte die COM-Schnittstelle für entsprechende Objekttypen das Sicherheitssystem unbedingt beachten.

enaio® capture

Überblick

enaio® capture ermöglicht die Erfassung, Verschlagwortung, Validierung und Import von großen Belegmengen. Dabei werden die Schritte getrennt und so weit wie möglich automatisiert. Bei jedem Schritt werden dann Belege aus verschiedenen Vorgängen bearbeitet. Einsatzgebiet für diese Verfahrensweise sind Vorgänge, bei denen regelmäßig eine große Menge von Papierbelegen erfasst wird.

Eine genaue Beschreibung finden Sie im Handbuch enaio® capture.

Die Funktion von enaio® capture kann grundlegend in vier Schritte eingeteilt werden:

§ Erfassung von Dokumentdateien (Scannen)

Hierbei kommen Scannen und Dateiimport zum Einsatz. Beim Scannen werden Hochleistungsscanner eingesetzt. Beim Dateiimport werden Dateien erfasst, die erzeugt werden, in der Regel durch andere DV-Systeme und regelmäßig in größerer Anzahl.

§ Zeichenerkennung

Die erfassten Dokument-Images werden durch OCR-Engines verarbeitet. Aus den vorliegenden Bilddaten werden Texte erkannt, die später für die Verschlagwortung verwendet werden.

§ Validierung

Hier werden die erfassten Dokument-Images und die daraus erkannte Verschlagwortung durch Bearbeiter optisch geprüft. Eine automatische Validierung durch VB-Skripte unterstützt die Bearbeiter dabei.

§ Import

Im letzten Schritt werden DMS-Objekte im enaio®-System erzeugt. Dieser Schritt wird gewöhnlich durch eine Komponente durchgeführt, die auch in der automatischen Import-Aktion eingesetzt wird. Abweichend davon können auch dBase-Dateien für den Import in andere Systeme erzeugt werden, die erzeugten enaio®-Objekte können auch automatisch per E-Mail an enaio®-Benutzer verteilt werden.

Für unterschiedliche Belegtypen können verschiedene Bearbeitungsabläufe erstellt werden. Diese Abläufe werden Konfigurationen genannt. Konfigurationen bestehen aus verschiedenen Arbeitsschritten, jeder Schritt wird durch ein Subprogramm bearbeitet. Die Bearbeitung von Papierbelegen erfolgt in Stapeln, den sogenannten Batches. Sobald die Belege eingescannt werden, wird nur noch mit den Images dieser Belege gearbeitet. Die Papierbelege können gelagert und, nachdem die Objekte im DMS erfasst und gesichert wurden, vernichtet werden, soweit die gesetzliche Lage das erlaubt.

Der produktive Einsatz von enaio® capture wird durch administrative Einrichtungs-Tätigkeiten vorbereitet. Zunächst werden die vorgesehenen Arbeitsplätze eingerichtet. Nachdem die Voraussetzungen für die Arbeit mit enaio® capture geschaffen sind, dazu gehört die Einrichtung der erforderlichen Hardware und Software, werden die Scanstationen über die Netzwerkinstallation mit den erforderlichen Client-Komponenten ausgestattet. Dann wird die Konfiguration von enaio® capture durchgeführt. Danach steht enaio® capture zur Verfügung und kann produktiv eingesetzt werden.

Administration von enaio® capture

In diesem Abschnitt soll die Vorbereitung der enaio® capture-Arbeitsplätze, die Installation, Konfiguration und der Betrieb von enaio® capture beschrieben werden. Nach Installation definieren Sie Konfigurationen und Subprogramme. Eine Konfiguration besteht aus Subprogrammen, die Daten bearbeiten und übergeben. Die Ausführung von Subprogrammen kann auf Arbeitsstationen verteilt werden, es ist auch eine automatische Ausführung von Subprogrammen einstellbar. Die Steuerung des Ablaufs eines Batches erfolgt in zwei Konfigurationsdateien. Die während der Bearbeitung erzeugten Daten werden in einer ODBC-Datenbank (empfohlen MS Access) zwischengespeichert.

Voraussetzungen für den Einsatz von enaio® capture

Technische Voraussetzungen

Für den Einsatz von enaio® capture wird zusätzliche Hardware benötigt und Software, die nicht zum enaio®-System gehört:

- § Anschlusskarten für Scanner: Scanner können über Kofax und Twain angesprochen werden, als Hardware können Kofax- und SCSI-Karten verwendet werden. Möglich ist auch der Anschluss über USB.
- § Scanner-Hardware und -Treiber: Es sind geeignete Scanner einzusetzen, die das vorliegende Belegvolumen bewältigen können. Als Treiber-Software kommen Kofax und TWAIN in Frage.
- § OCR-Engine: Unterstützt wird Abbyy FineReader.
- § Kofax Retrieval Engine: Auf Rechnern ohne Kofax-Scanengine ist für Scan-Subprogramme die Kofax Retrieval Engine zu installieren. Diese dient zur Anzeige von Bildinhalten.
- § E-Mail-Clients: enaio® capture ist in der Lage, mit Subprogrammen die erfassten Images oder Verweise auf die erzeugten DMS-Objekte per E-Mail zu versenden. Dazu muss ein MAPI-Client, z.B. MS Outlook installiert sein.

Organisatorische Voraussetzungen

In enaio® capture werden einzelne Arbeitsschritte bei der Dokumentenerfassung so weit isoliert, dass möglichst viele Aufgaben automatisch durchgeführt werden können, so z.B. das Scannen der Belege, die Barcode-Erkennung (mit eindeutiger Ordner-ID) und der Dokumentenimport. Zur Bearbeitung wird eine größere Anzahl gleichartiger Dokumente zu Dokumentenstapeln, sogenannten Batches, zusammengefasst. Die Bearbeitung von Belegen mit enaio® capture ist in das Umfeld der Organisation eingebettet.

Berechtigungen: Im enaio® administrator sind Benutzer festzulegen, die

- § enaio® capture konfigurieren dürfen

§ enaio® capture starten dürfen

Die enaio® capture-Dateien werden durch das enaio®-Setup als Fileserver installiert. An Arbeitsplätzen werden Netzwerkinstallationen durchgeführt, wodurch die Verbindung zu den enaio® capture-Dateien hergestellt wird.

Alternativ ist auch die lokale Clientinstallation möglich. Mit dieser Installationsform können aber die Aktionen verschiedener enaio® capture -Stationen nicht koordiniert werden, falls die lokale Clientinstallation nicht durch Verzeichnisfreigabe zu einer Fileserver-Installation umgewandelt wird. Zur Koordination von enaio® capture -Stationen ist der gemeinsame Zugriff auf zwei Konfigurationsdateien und auf die Batch-Datenbank erforderlich.

Für eine erfolgreiche Installation sind folgende Voraussetzungen zu erfüllen:

§ Hardware und Software entsprechen den Anforderungen.

§ Ein laufender enaio®-Server ist vorhanden.

§ enaio® capture kann nur mit gültiger Lizenzierung gestartet werden.

enaio® capture und die Subprogrammtypen verfügen über eine automatische Aktualisierungsfunktion. Wird beim Start eine aktuellere Version im enaio® capture-Verzeichnis `etc\update\index` gefunden, werden die Komponenten automatisch aktualisiert, wenn die Aktualisierungsfunktion in enaio® administrator konfiguriert wurde.

Standard-Subprogramme

Die folgende Tabelle soll einen Überblick über die installierten enaio® capture -Komponenten gegeben werden. Die Programme sind nach ihrer Funktion geordnet.

| Steuerung | Konfig. | Scannen | OCR/DB | Validierung | Import |
|-----------|----------|----------|----------|-------------|-----------|
| AXINDEX | AXRCCONF | AXICSCAN | | AXVALID | AXIMPORT |
| AXIASTOP | | AXDSCAN | AXICSVR | SCRIPTS | AXMAILDC* |
| | | AXTWSCAN | AXNOOCR | | AXIMPMDC* |
| | | AXIMGIMP | AXFINER7 | | AXPARTDC |
| | | | | | AXSTARTWF |

Tabelle 46: enaio® capture -Subprogramme

Die mit * markierten Subprogramme konfigurieren sich selbst, d.h. dass sie die Komponente AXRCCONF nicht dafür verwenden. Für die Funktionen 'Konfiguration', 'Scannen' 'OCR/DB', 'Validierung' und 'Import' können eigene Programme eingesetzt werden, sofern die Schnittstellen von enaio® capture korrekt genutzt werden. Informationen dazu sind bei OPTIMAL SYSTEM erhältlich.

Die Subprogramme sollen hier kurz beschrieben werden. Sie werden gruppiert nach folgenden Grundfunktionen: Image-Input, Basis-Indizierung, Validierung und Output.

Steuerung

§ axindex

Dahinter steht die ausführbare Datei `axindex.exe`. Sie kann ohne Parameter gestartet werden. Dann stehen die enaio® capture -Funktionalitäten in Abhängigkeit vom angemeldeten Benutzer bereit. Wird die Datei mit dem Startparameter `/auto` gestartet, dann startet enaio® capture im Automodus, in dem keine Konfiguration und kein manuelles Bearbeiten von Batches möglich ist. Batches werden bei den Subprogrammen automatisch bearbeitet, die entsprechend konfiguriert sind.

§ axiastop

Dahinter steht die ausführbare Datei `axiastop.exe`. Sie wird automatisch gestartet, wenn enaio® capture im Automodus ausgeführt wird. Es erlaubt das Beenden von enaio® capture, das sonst sehr umständlich ist, wenn viele Batches im Automodus schnell nacheinander bearbeitet werden müssen.

Konfiguration

Dies sind die Programme, die Konfigurationsfunktionen für Subprogramme beinhalten:

§ axrconf

Die zugehörige ausführbare Datei ist `axrconf.exe`. Sie wird für die Konfiguration der meisten Standard-Subprogramme gestartet. Darin werden die erforderlichen Einstellungen gemacht:

§ Felddefinition

Es werden Felder definiert, in denen die während der Batchverarbeitung erfassten Werte gespeichert werden.

§ Seitentrennung

Hier wird das Kriterium eingestellt, anhand dessen entschieden wird, wann im Dokumentenstapel ein neues Dokument beginnt.

§ Import

Hier erfolgt die Auswahl der ODBC-Quelle zur Batch-Datenbank und die Zuordnung von Batch-Feldern zu Importfeldern in enaio®.

§ Axmaildc und aximpmdc

Diese beiden Programme konfigurieren sich selbst. Bei der Einrichtung ist lediglich der Standort einer vorbereiteten Konfigurations-Datei anzugeben.

§ Andere

Für Subprogramme außerhalb des Standard-Lieferumfangs müssen eigene Subprogramme angegeben werden.

Scannen

Dies sind die Programme, die Bilddateien in enaio® capture erzeugen oder importieren. Folgende Subprogramme sind im Standard-Lieferumfang:

§ axicscan

Die zugehörige ausführbare Datei ist `axicscan.exe`. Sie erlaubt das Scannen und integriert die Barcodeerkennung mit Kofax-Engines. Bei Beendigung des Scanvorgangs und Weiterleiten des Batches legt `axicscan` eine neue Tabelle in der Batch-Datenbank an. Daher ist dieses Subprogramm auch in der Funktion OCR/DB eingetragen.

§ axdscan

Die zugehörige ausführbare Datei ist `axdscan.exe`. Sie erlaubt das Scannen mit Kofax-Engines. `axdscan` legt keine neue Tabelle in der Batch-Datenbank an.

§ atwscan

Die zugehörige ausführbare Datei ist `atwscan.exe`. Sie erlaubt das Scannen mit TWAIN-Scannern. `atwscan` legt keine neue Tabelle in der Batch-Datenbank an.

§ aximgimp

Die zugehörige ausführbare Datei ist `aximgimp.exe`. Sie erlaubt das Importieren von Bilddateien in den Verarbeitungsablauf von enaio® capture. `aximgimp` legt keine neue Tabelle in der Batch-Datenbank an.

OCR/DB

Hier sind Subprogramme beschrieben, die eine Tabelle in der Batch-Datenbank anlegen und die evtl. durch Zeichenerkennung erfasste Verschlagwortung in die Tabelle einfügen.

§ axicsrv

Die zugehörige ausführbare Datei ist `axicsrv.exe`. Sie erlaubt die Barcodeerkennung mit Kofax-Engines. Bei Beendigung des Scanvorgangs und Weiterleiten des Batches legt `axicsrv` eine neue Tabelle in der Batch-Datenbank an.

§ axnoocr

Die zugehörige ausführbare Datei ist `axnoocr.exe`. Sie führt keine Zeichenerkennung durch und legt eine neue, aber leere Tabelle in der Batch-Datenbank an.

§ axfiner

Die zugehörige ausführbare Datei ist `axfiner.exe`. Sie erlaubt die Zeichenerkennung mit FineReader-Engines. Bei Beendigung des Scanvorgangs und Weiterleiten des Batches legt `axrecog` eine neue Tabelle in der Batch-Datenbank an.

Validierung

Hier sind die Subprogramme aufgeführt, die für die Daten- und Bildvalidierung eingesetzt werden können.

§ axvalid

Die zugehörige ausführbare Datei ist `axvalid.exe`. In dieser Anwendung werden die erfassten Images und die vorhandene Verschlagwortung durch Bearbeiter geprüft und, wenn erforderlich, korrigiert. Außerdem ist `axvalid` ausgezeichnet für die manuelle Nacherfassung der Verschlagwortung geeignet. Eine weitere wichtige Eigenschaft ist die Möglichkeit, per VB Script die erfassten Werte zu validieren.

Import

Hier sind die Subprogramme aufgeführt, mit denen die Daten und die Dokumentdateien in enaio® importiert werden oder für den Import in andere Systeme aufbereitet werden.

§ aximport

Die zugehörige ausführbare Datei ist `aximport.exe`. Zunächst wird damit der Batch in die Einzeldokumente getrennt. Dann werden die Daten aus der Batch-Datenbank und die erzeugten Bilddateien in das enaio®-System importiert, mit dem enaio® capture verbunden ist. Basis ist derzeit die Import-Engine, die auch von der automatischen Aktion 'Import/Export' eingesetzt wird.

§ axmaildc

Die zugehörige ausführbare Datei ist `axmaildc.exe`. Damit werden die Daten aus der Batch-Datenbank und die erzeugten Bilddateien per E-Mail an konfigurierte Empfänger versandt, ohne sie in das enaio®-System zu

importieren. Der Empfänger kann nun damit die Erzeugung eines enaio®-Objekts manuell vornehmen.

§ aximpmdc

Die zugehörige ausführbare Datei ist axmaildc.exe. Zunächst wird damit der Batch in die Einzeldokumente getrennt. Dann werden die Daten aus der Batch-Datenbank und die erzeugten Bilddateien in das enaio®-System importiert, mit dem enaio® capture verbunden ist. Bis dahin entspricht die Funktionalität der von aximport. Dann werden aber Verweise auf die erzeugten Objekte per E-Mail an konfigurierte Empfänger versandt, ohne sie in das enaio®-System zu importieren. Der Empfänger kann nun die erzeugten enaio®-Objekte weiter bearbeiten.

§ axstartwf

Die zugehörige ausführbare Datei ist Datei axstartwf.exe. Dieses Subprogramm importiert das erfasste Dokument in die Workflow-Komponente von enaio®. Es werden dabei keine recherchierbaren DMS-Objekte erzeugt. Die definierten Batchfelder können Workflow-Variablen zugeordnet werden, eine Zuordnung zu Objektfeldern erfolgt nicht.

Eigene Subprogramme

Zusätzlich zu den aufgeführten Subprogrammen lassen sich auch eigene Programme in die Verarbeitung einbinden. Voraussetzung ist die Einhaltung einer von OS definierten Schnittstelle. Für die Beschreibung der Schnittstelle wenden Sie sich bitte an die Consulting-Abteilung von OS.

Verteilung auf mehrere Arbeitsstationen im Netz

enaio® capture ist so konfigurierbar, dass Verarbeitung der Batches auf einer einzelnen Station möglich ist. Die Arbeit mit lediglich einer Station ist aber nur bei begrenzten Belegaufkommen sinnvoll. Wenn das Aufkommen eine bestimmte Größe überschreitet, kann die Arbeit nicht mehr an einer Arbeitsstation und von einer Person bewältigt werden. Dann muss die Arbeit auf mehrere Arbeitsstationen verteilt werden.

Da einige Subprogramme automatisch ausführbar sind, z.B. Zeichenerkennungs- und Import-Subprogramme, und einige immer einen Benutzereingriff erfordern, z.B. Scannen und Validieren, ist es wenig sinnvoll, die Arbeitsstationen lediglich zu duplizieren und an jeder Station alle Arbeitsschritte durchzuführen. Viel sinnvoller ist eine funktionale Strukturierung der Verarbeitung. Die beiden Typen der Verarbeitung, automatische Arbeitsschritte und solche Arbeitsschritte, die Benutzereingriffe erfordern, sollten getrennt werden.

Der Arbeitsablauf würde wie folgt aussehen:

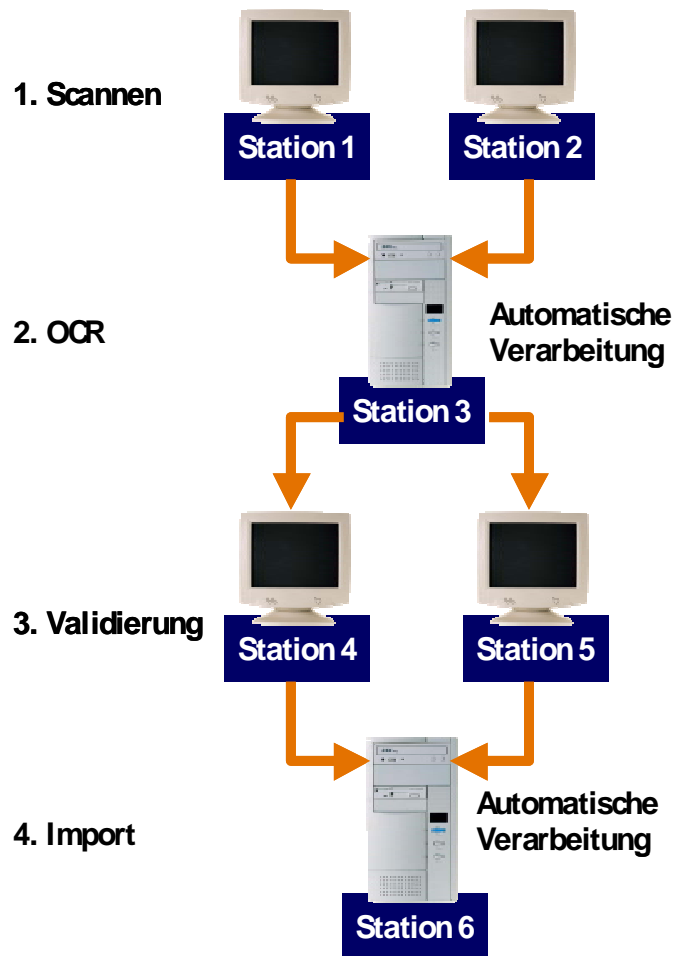


Abbildung 58: Verteilte Verarbeitung

Hier ist nur der Verlauf der Verarbeitung einzelner Batches dargestellt, der Papiereingang ist hier nicht dargestellt. Zu bemerken ist, dass auf Station 3 zwei Subprogramme ausgeführt werden, OCR und Import.

Um den dargestellten Ablauf einzurichten, sind folgende Konfigurations-Schritte erforderlich:

- § Die Lizenzierung der einzelnen Stationen im enaio® administrator
- § Berechtigungen der Benutzer im enaio® administrator
- § enaio® capture -Berechtigungen für die einzelnen Konfigurationen

Lizenzierung

Die Lizenz-Einrichtung erfolgt im enaio® administrator. Folgende Schritte zur Lizenzierung sind erforderlich:

- § Eintragung der einzelnen Stationen
- § Zu den einzelnen Stationen sind dann die einzelnen Lizenzmodule hinzuzufügen

Folgende Lizenzen sind zu vergeben, sofern die jeweilige Eigenschaft eingesetzt werden soll:

| Modul | Anwendung |
|-------|------------------|
| AIX | enaio® capture |
| SCA | Scan-Komponenten |
| STW | Scannen Twain |

| | |
|-----|---------------------------------|
| SIC | Scannen und Recognition (Kofax) |
| REK | Recognition (Kofax) |
| RER | Recognition (FineReader) |
| W2D | W- zu D-Konvertierung (Barcode) |
| SFI | Scan File-Import |
| VAL | Validierung |
| AIE | Import |
| AMT | MAPI-Export |
| AVG | Workflow starten |

Tabelle 47: Lizenzmodule

Wenn die Lizenzen korrekt verteilt sind, können an den einzelnen Stationen die jeweiligen Anwendungen ausgeführt werden. Damit ist aber nicht gesichert, dass Benutzer sich an enaio® capture auch anmelden können. Als nächstes sind daher Objekt- und Anwendungsberechtigungen im enaio® administrator einzustellen.

Detaillierte Informationen zur Verteilung von Lizenzen können Sie dem Handbuch zum enaio® administrator entnehmen.

Objekt- und Anwendungsberechtigungen

Bei der Einrichtung der Berechtigungen für die enaio® capture -Benutzer gibt es zwei Stellen, die berücksichtigt werden müssen. Die Sichtbarkeit einzelner enaio® capture -Konfigurationen wird über die Benutzergruppen eingestellt. Die Berechtigungen zum Konfigurieren und Ausführen von enaio® capture werden dagegen für jeden Benutzer einzeln eingestellt.

Wenn Sie die Sichtbarkeit von enaio® capture -Konfigurationen strukturieren wollen, dann legen Sie verschiedene Benutzergruppen an. Anschließend legen Sie einzelne Benutzer an, erteilen die erforderlichen Anwendungsberechtigungen und ordnen den Benutzern Benutzergruppen zu. In enaio® capture stellen Sie später für jede einzelne Konfiguration die Gruppen ein, die die Konfigurationen sehen dürfen.

Detaillierte Informationen zum enaio®-Sicherheitssystem können Sie dem Handbuch zum enaio® administrator entnehmen. Ein enaio® capture -Administrator darf grundsätzlich alle Konfigurationen sehen.

Zusätzliche Informationen zur Verarbeitung im Netz

Damit sind die Stationen im Netz so eingerichtet, dass sie ausschließlich die geplante Funktion erfüllen. Die Benutzergruppen und in enaio® eingetragene Benutzer sind auch entsprechend eingerichtet.

Zusätzlich ist noch zu berücksichtigen:

- § Wenn mit den Stationen eine Lastverteilung bei der Verarbeitung der Batches erreicht werden soll, müssen alle Stationen mit einem Batchverzeichnis arbeiten. Dies ist damit zu erreichen, dass die Installation von enaio® capture nur einmal zentral durchgeführt wird. Auf den einzelnen Stationen ist jeweils eine Verknüpfung mit der zentralen axindex.exe zu erstellen.
- § Abhängig von den installierten Modulen sind Anzeige-, Scan- oder Erkennungs-Engines auf den Stationen zu installieren. Für Scanstationen ist eine Scan-Engine, für die Validierungsstationen zumindest eine reine Anzeige-Engine zu installieren. Station 3 der o.g. Beispiel-Grafik erfordert eine Erkennungs-Engine.

- § Die Verteilung der Batches erfolgt durch die Steuerkomponente automatisch nach dem FIFO-Prinzip: der erste eingegangene Batch wird von der ersten anfragenden Station zuerst bearbeitet. Dies gilt für die automatische Ausführung. Manuell kann jeder Batch gestartet werden.

Anhang

Systemrollen

Hier soll eine vollständige Übersicht über die Systemrollen, die im System zur Verfügung stehen, erfolgen. Verschiedene Systemrollen sind für verschiedene Clientanwendungen gültig und regeln den möglichen Aktionsumfang des an diesem Client angemeldeten Benutzers.

Folgende Systemrollen können in enaio® administrator den einzelnen Benutzern zugewiesen werden:

| ID | Systemrolle | Beschreibung |
|----|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| 66 | DMS: Lokale Sicherheitsgruppen einrichten | Konfiguration dezentraler Bereiche |
| 18 | DMS: Supervisor | Alle Systemrollen außer die für Workflow-Funktionen, enaio® enterprise-manager und Server: Jobkontext wechseln |
| 1 | Administrator: Starten | enaio® administrator starten |
| 2 | Administrator: Konfiguration Gesamtsystem | Konfiguration des Gesamtsystems (LDAP, Automatische Aktionen etc.) |
| 4 | Administrator: Konfiguration Sicherheitssystem | Konfiguration des Sicherheitssystems (Benutzer- und Rechteverwaltung) |
| 65 | Administrator: Konfiguration lokale Sicherheitsgruppen | Eingeschränkte Konfiguration des Sicherheitssystems für dezentrale Bereiche |
| 5 | Administrator: Konfiguration W-Vorlagen | Verwaltung der W-Vorlagen (Konfiguration von W-Vorlagen und zugeordneten Programmen) |
| 6 | Administrator: Konfiguration Archivdruck | Konfiguration des Archivdrucks |
| 8 | Administrator: Automatische Aktionen starten | Eingerichtete Automatische Aktionen starten |
| 9 | Administrator: Automatische Aktionen konfigurieren | Automatische Aktionen einrichten und konfigurieren |
| 10 | Administrator: Benutzerablage einsehen | Private Ablagen aller Benutzer in enaio® administrator einsehen |

| | | |
|----|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 50 | Administrator: Objektrechte verändern | Andere Benutzern als Besitzer angeben oder Objektbesitz selbst übernehmen |
| 11 | Editor: Starten | enaio® editor starten |
| 12 | Editor: Objektdefinition bearbeiten | Objektdefinition ändern und ergänzen Mit dieser Rolle allein können die Änderungen nicht in der Datenbank gespeichert werden. |
| 13 | Editor: Datenbank anpassen | Datenbankanpassungen vornehmen |
| 14 | Start: Starten | enaio® start zum Ausführen Automatischer Aktionen starten |
| 15 | Capture: Starten | enaio® capture starten |
| 16 | Capture: Konfigurieren | Konfigurationen für enaio® capture einrichten |
| 36 | Client: Starten | enaio® client starten |
| 17 | Client: Eigene Einstellungen speichern | Individuelle Benutzereinstellungen vornehmen und Anfragen und/oder Dokumente im Archivbereich von enaio® client speichern |
| 80 | Client: Externe Anwendungen konfigurieren | Externe Anwendungen in enaio® client einbinden |
| 77 | Client: Menüband anpassen | Menüband anpassen |
| 28 | Client: Historie öffnen | Bearbeitungshistorie einsehen, sofern diese aktiviert ist |
| 32 | Client: Daten aus Historie wiederherstellen | Frühere Daten über die Bearbeitungshistorie wiederherstellen (Voraussetzung: Systemrolle 'Client: Historie öffnen' ist vorhanden) |
| 56 | Client: Historie für einzelnes Objekt konfigurieren | Konfiguration der Historie für ein einzelnes Dokument, sofern dies in der Objektdefinition vorgesehen ist (Voraussetzung: Systemrolle 'Client: Historie öffnen' ist vorhanden) |
| 29 | Client: Expertenmodus starten | Erweiterte Suche nach Objekten |
| 30 | Client: Aus Trefferliste exportieren | Indexdaten und Dokumentdateien aus einer Trefferliste exportieren |
| 74 | Client: Dokumente drucken | Dokumenten über enaio® client drucken |
| 71 | Client: Notizen öffnen | Notizen an Objekten anzeigen |

| | | |
|----|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31 | Client: Notizen bearbeiten | Notizen zu Objekten erstellen und bearbeiten |
| 33 | Client: Persönlichen Papierkorb anzeigen | Persönlichen Papierkorb anzeigen, aus dem gelöschte Daten wiederhergestellt werden können |
| 37 | Client: Systempapierkorb anzeigen | Systempapierkorb anzeigen, in dem die gelöschten Objekte aller Benutzer sichtbar sind (Voraussetzung: Systemrolle 'Client: Persönlichen Papierkorb anzeigen' ist vorhanden) |
| 40 | Client: Objekte aus dem Papierkorb löschen | Objekte im Papierkorb unwiederbringlich löschen (Voraussetzung: Systemrolle 'Client: Persönlichen Papierkorb anzeigen' ist vorhanden) |
| 34 | Client: SQL-Anfragen ausführen | SQL-Anfragen ausführen |
| 35 | Client: Inhalt als E-Mail versenden | Objekte per E-Mail an einen Empfänger, der nicht im enaio®-System eingerichtet ist, senden |
| 44 | Client: Eigenschaft öffnen | Eigenschaftsdialog zu einem Objekt anzeigen |
| 63 | Client: Objekte verschieben | Objekte verschieben, sofern dies in enaio® administrator erlaubt ist |
| 79 | Client: Schrankübergreifend verschieben | Dokumente schrankübergreifend verschieben |
| 19 | Client: Workflow benutzen | Workflows benutzen |
| 61 | Client: Workflow Stellvertreterkonfiguration | Workflow-Stellvertreter für sich selbst einrichten |
| 62 | Client: Workflow Prozessadministration | Eingeschränkte Administration laufender Prozesse aus enaio® client |
| 67 | Client: Workflow Laufliste bearbeiten | Workflow-Laufliste bearbeiten |
| 68 | Client: Workflow private Lauflistenvorlagen verwalten | Vorlagen für private Lauflisten verwalten und löschen |
| 69 | Client: Workflow öffentliche Lauflistenvorlagen verwalten | Verwalten und Löschen von Vorlagen öffentlicher Lauflisten |
| 51 | Client: Objektsuche verwenden | Bereich 'Objektsuche' anzeigen |

| | | |
|----|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 52 | Client: Navigation verwenden | Bereich 'Navigation' anzeigen |
| 78 | Client: Standardsuche ausführen | Wenn Benutzer ausschließlich über gespeicherte Anfragen Suchen ausführen sollen, kann die Standardsuche (Suche nach Indexdaten und Volltextsuche) ausgeschaltet werden. |
| 59 | Client: Icon hinzufügen | Icons in den Datenbestand des Systems aufnehmen |
| 60 | Client: Icon löschen | Icons aus dem Datenbestand des Systems entfernen |
| 64 | Client: Statische Folien anderer Benutzer bearbeiten | Statische Folien, die nicht selbst erstellt wurden, bearbeiten |
| 38 | Client: SQL-Anfragen bearbeiten | SQL-Anfragen erstellen und bearbeiten |
| 39 | Client: Abonnements administrieren | Abonnements aller Benutzer administrieren und löschen und anderen Benutzern Abonnements einrichten |
| 55 | Client: Wiedervorlagen administrieren | Wiedervorlagen aller Benutzer administrieren und löschen |
| 45 | Client: Events debuggen | Events für den Client im Debug-Modus ausführen |
| 49 | Client: Events erstellen | Events für Client und Server bearbeiten |
| 46 | Client: Relationen darstellen | Relationsdarstellung/Relationsliste anzeigen, sofern Relationsverknüpfungen im System eingerichtet sind |
| 47 | Client: Visualisierung erstellen | Relationen visualisieren und die Visualisierung im zugeordneten Dokumenttyp speichern |
| 48 | Client: Visualisierung administrieren | Allgemeine Konfiguration für die Relationsvisualisierung |
| 75 | Client: Archivierungsstand ändern | Dokumenten die Eigenschaft 'archivierbar' oder 'nicht archivierbar' geben |
| 53 | Client: Archivierte Dokumente löschen | Dokumente mit dem Status 'archiviert' löschen |
| 57 | Client: Sammeländerungen durchführen | Sammeländerungen für mehrere ausgewählte Objekte in einer Trefferliste durchführen |

| | | |
|----|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 73 | Client: Variantenverwaltung immer anzeigen | Benutzer, die keine Schreibrechte auf Indexdaten und oder Objekte haben, können die Variantenverwaltung schreibgeschützt öffnen, jedoch keine neuen Varianten erstellen oder anderen Varianten den Status aktive Variante zuordnen. |
| 76 | Client: Favoriten anzeigen | Für jeden Benutzer werden Favoriten verwaltet, deren Objekte auf mobilen Geräten über enaio® apps als Favoriten zugänglich sind. |
| 81 | Client: Vorschauanmerkungen ansehen | Anmerkungen in der optionalen PDF-Vorschau anzeigen |
| 82 | Client: Vorschauanmerkungen bearbeiten | Anmerkungen in der optionalen PDF-Vorschau anlegen, bearbeiten und löschen |
| 70 | WebClient: Starten | enaio® webclient starten |
| 20 | WF-Admin: Starten | enaio® administrator-for-workflow starten |
| 21 | WF-Editor: Starten | enaio® editor-for-workflow starten |
| 22 | WF-Editor: Organisation bearbeiten | Workflow-Organisation bearbeiten |
| 58 | WF-Editor: Benutzer an-/abwesend melden | Anwesenheit anderer Workflow-Benutzer einstellen |
| 23 | WF-Editor: Modell erstellen | Workflow-Modell erstellen |
| 24 | WF-Prozesse: Per Import starten | Workflow-Prozesse über einen Systemimport starten, der durch die Automatische Aktion 'Daten-/Dokumentenimport' ausgeführt wird |
| 25 | WF-Simulation: Starten | Workflow-Simulation starten |
| 26 | WF-Script: Starten | Laufzeitumgebung für Workflow-Skripte starten |
| 27 | Enterprise-Manager: Starten | enaio® enterprise-manager starten |
| 72 | Server: Jobkontext wechseln | |

enaio® enterprise-manager

In der Vorgängerversion enaio® wurde zur Verwaltung serverspezifischer Einstellungen der enaio® enterprise-manager entwickelt, über den sich mehrere Anwendungsserver konfigurieren lassen. Der Enterprise Manager wurde als Snap-In für die Microsoft Management Console realisiert.

Durch den enaio® enterprise-manager sind mehrere Einstellungsmöglichkeiten, die auf Admin, Lizenzmonitor etc. verteilt waren, zentralisiert worden und sollten nur noch aus dem enaio® enterprise-manager heraus benutzt werden. Zusätzlich dazu besteht jetzt die Möglichkeit, aus dem enaio® enterprise-manager heraus auf die Datenbank zuzugreifen, den Inhalt von Tabellen einzusehen und per SQL die Datenbank zu manipulieren.

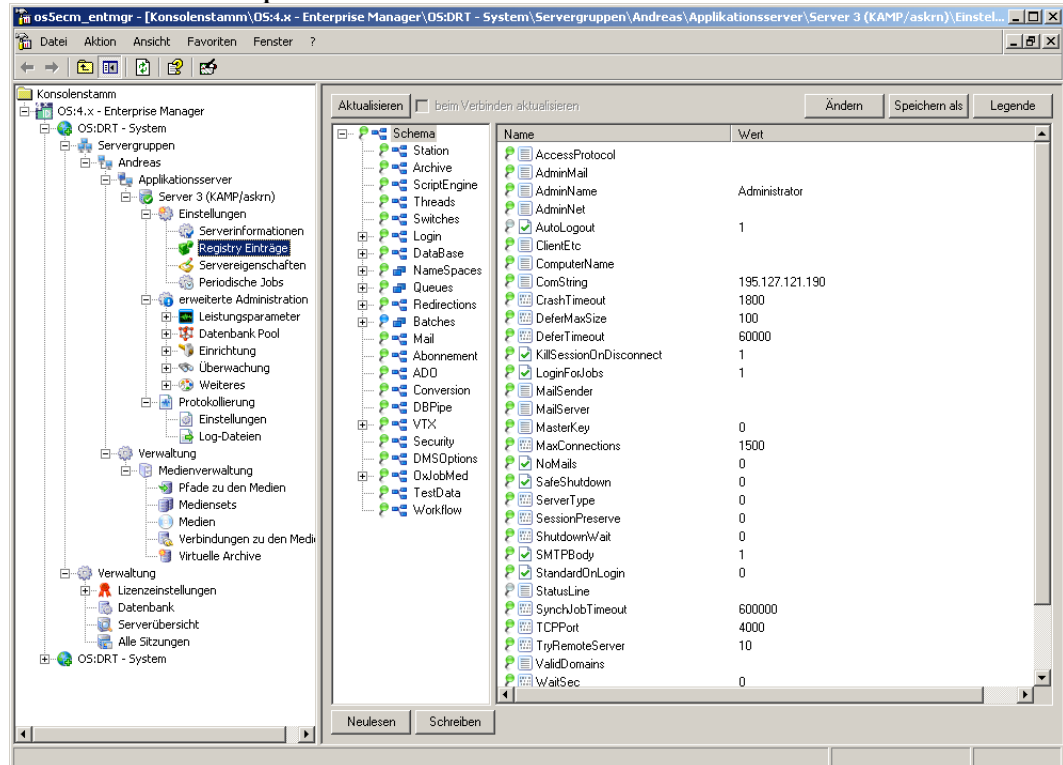


Abbildung 59

Mit Hilfe des enaio® enterprise-managers wird eine Visualisierung der Serverinfrastruktur möglich. So werden alle Servergruppen eines Systems in Serverfamilien zusammengefasst. Innerhalb einer Servergruppe können die Einstellungen aller Instanzen ermittelt werden. Alle Operationen, die mit enaio® enterprise-managers durchgeführt werden können, finden Sie im Administrationshandbuch.

Registrierungseinträge für enaio® server

Die Registrierungseinträge für aktuelle Version von enaio® server werden in einer HTML-Datei dokumentiert. Diese HTML-Datei können sie sich speziell für die bei Ihnen installierte Server Version aktuell erzeugen, indem Sie das Java-Skript *oxentmgr_XML_to_HTML.js* ausführen.

Diese Skript-Datei befindet sich im Installationsverzeichnis des enaio® administrator. Es wird von diesem Skript im selben Verzeichnis die Datei *oxentmgr_sp.html* erzeugt, in der die verfügbaren Registrierungseinträge der aktuell installierten enaio® server-Version dargestellt werden. Die dort aufgeführten Registrierungseinträge können alle mit dem enaio® enterprise-manager bearbeitet werden.

Start von enaio® server und die Kommandozeile

Switches und Variablen

Der Server hat einige interne Switches und Variablen, die in der Kommandozeile gesetzt werden können.

Debug: definiert, ob gleich nach der Analyse der Kommandozeile ein Debugger angefordert wird. Standardwert: OFF. Falls auf ON eingestellt, wird das Betriebssystem aufgefordert, einen Debugger zu starten, dem der startende Serverprozess übergeben wird. Dies ist hilfreich beim Debuggen des Dienstes. Für den laufenden Betrieb ist davon abzuraten, diesen Switch auf ON zu setzen. Standardwert: OFF

Service: wenn dieser Switch auf ON steht, kommuniziert der Server mit dem Service Control Manager, und versucht als Dienst zu starten. Wenn der Switch auf OFF steht, startet der Server als normale exe. Standardwert: ON

Console: wenn dieser Switch auf ON steht, wird versucht, beim Serverstart eine Konsole zu starten und eine Verbindung herzustellen. Standardwert: OFF

Verbose: wenn dieser Switch auf ON steht, wird der Fehler, der den Serverstart verhindert, in einem Nachrichtenfenster angezeigt. Das geschieht abhängig davon, ob der Server als Dienst startet oder nicht. Standardwert: OFF

ReportEnabled: wenn dieser Switch auf ON steht, wird ein Bericht über den Server-Startup so wie über den Server-Shutdown erstellt. Der Standardwert ist ON.

Diese Berichte werden in das Verzeichnis `\server\ostemp\` gelegt.

Sie tragen folgende Bezeichnung:

`startup.txt` und `shutdown.txt`

Durch den Switch 'ReportDateTime' kann die Bezeichnung um das Erstellungsdatum ergänzt werden:

`startup_YYYYMMDD_HHMMSS.txt` und `shutdown_YYYYMMDD_HHMMSS.txt`

ReportRegistry: steht dieser Switch auf ON, dann werden im Startbericht, falls vorhanden, auch Parameter berichtet, die der Kernel aus der Registrierung gelesen hat. Dafür wird Platz im Bericht benötigt und macht ihn weniger übersichtlich. Standardwert: OFF

ReportLicense: wenn dieser Switch auf ON steht, werden auch alle Module protokolliert, die in der `oslicense` aktuell stehen, und alle Module, die in der Tabelle `osresources` eingetragen sind. Standardwert: OFF

ReportFiles: wenn dieser Switch auf ON steht, werden auch alle Systemdateien berichtet, die in der Tabelle `osresources` eingetragen sind. Standardwert: OFF

ReportDatetime: wenn dieser Switch auf ON steht, wird aktuelle Zeit benutzt, um die Berichtdatei zu benennen. Standardwert: OFF

Name: diese Variable definiert den Namen des Dienstes und den Namen des Registrierungsschlüssels, woraus alle Einstellungen auszulesen sind.

Zeitintervall: Legt Zeitintervalle für einzelne Startschritte fest. Wird dieses Zeitintervall überschritten, erfolgt eine Warnung. Mehrere aufeinander folgende Zeitintervalle sind möglich.

Startvorgang

Die `oxrpt.dll` wird geladen und initialisiert. Dabei wird der Alias 'axsvckrn' verwendet. Wenn diese Initialisierung scheitert, wird der Prozess mit dem Fehlercode `SRETCODE_REPORTINIT` beendet.

Alle Switches und Variablen werden mit Standardwerten initialisiert. Dabei wird vermerkt, dass die Kommandozeile noch nicht analysiert ist und der Server noch nicht weiß, ob der Benutzer ihn als Dienst startet will oder als normale Anwendung. Diese Einstellungen werden in einer internen Variable `ServerSwitchDefined` gespeichert.

Dann wird die Kommandozeile analysiert.

Danach, wenn der Switch `Debug` auf `ON` steht, wird sofort nach einem Debugger verlangt.

Dann wird der Name geprüft. Falls dieser leer ist, wird der Standardwert `'askrn'` verwendet.

Danach wird `perflib` geladen, aber noch nicht aktiviert.

Danach wird geprüft, ob ein Prozess mit gleichem Namen bereits läuft. Dafür wird ein systemweites Flag benutzt, der den Namen als Teil seines Namens hat. Das heißt, es wird geprüft, ob wir die ersten sind, die diesen Wert als Name benutzen. Wenn die Prüfung scheitert, wird ein Protokoll geschrieben, im `Verbose`-Mode eine `MessageBox` angezeigt, `Windows-EventLogEvent` erstellt, und der Prozess wird mit dem Fehlercode `SRET_CODE_ALREADY_RUNNING` beendet.

Anschließend wird die Variable `ServerSwitchDefined` geprüft. Falls sie den Wert `FALSE` hat, heißt es, der Benutzer hat den `Service-Switch` nicht definiert, und es ist herauszufinden, wie gestartet werden soll. Es werden mit Hilfe der `ToolHelp-Library` alle Prozesse aufgelistet, 'unser' Prozess gefunden, und der Elternprozess unseres Prozesses ermittelt. Falls es `'services.exe'` ist, dann geht der Server davon aus, dass er vom `ServiceControlManager` als Dienst gestartet wurde. Dementsprechend wird der Switch `Service` auf `ON` gesetzt. Falls dem nicht so ist, wird er auf `OFF` gesetzt.

Wenn der Switch `Service` auf `OFF` steht, dann wird der Switch `Console` zwangsläufig auf `ON` gestellt. D.h. wenn nicht als Dienst, dann immer mit der Konsole. Wenn als Dienst, dann entweder mit der Konsole, oder auch ohne, je nach Kommandozeilenparameter.

Es wird jetzt mit dem `SCM` kommuniziert, falls `Service` auf `ON` steht, und letztendlich der `MainThread` gestartet. Was hierbei passiert, muss getrennt beschrieben werden. Danach wird der letzte Protokolleintrag erstellt.

Kommandozeile

Die Kommandozeile hat folgendes Format (in BNF Syntax):

```
Axsvrkrn.exe parameter-list
parameter-list ::= [ parameter [ blank parameter-list ] ]
blank ::= " " - ein Leerzeichen, schwer im Word darstellbar.
parameter ::= { '-' | '/' | '\' } { debug | action | console | service | name
| paramfile | verbose | report }
action ::= { install | uninstall }
install ::= { ,I' | ,i' } - setzt der Wert des Switches Mode auf INSTALL
uninstall ::= { 'U' | 'u' } - setzt der Wert des Switches Mode auf UNINSTALL
debug ::= { 'D' | 'd' } - setzt der Wert des Switches Debug auf ON
console ::= { 'C' | 'c' } - setzt der Wert des Switches Console auf ON
service ::= { 'S' | 's' } [ '0' | '1' | ... ] - setzt der Wert des Switches
Service auf ON, falls '1' und auf OFF falls '0'. Ansonsten bleibt der Wert
unverändert.
verbose ::= { 'V' | 'v' } - setzt der Wert des Switches Verbose auf ON
report-registry ::= { 'RR' | 'rr' } - setzt der Wert des Switches
ReportRegistry auf ON
```

```
report-license ::= { 'RL' | 'rl' } - setzt der Wert des Switches
ReportLicense auf ON
report-disable ::= { 'R-' | 'r-' } - setzt der Wert des Switches
ReportEnabled auf OFF
report-files ::= { 'RF' | 'rf' } - setzt der Wert des Switches ReportFiles
auf ON
report-datetime ::= { 'RT' | 'rt' } - setzt der Wert des Switches
ReportDatetime auf ON
name ::= { 'N' | 'n' } blank filename - definiert den Wert der Variable
Name
zeitintervall ::= { 't1' | 'T1' } blank Millisekunden - definiert den Wert
der Variable Zeitintervall. Mehrere aufeinander folgende Zeitintervalle
können angegeben werden: -t1 3000 -t2 6000 -t3 9000 etc.
paramfile ::= { 'E' | 'e' } blank filename - definiert den Wert der Variable
ParamFile
filename ::= eine Zeichenkette ohne Leerzeichen
```

Es können mehrere Parameter gleichzeitig definiert werden, auch wiederholt, und auch die, die nicht kompatibel sind, z.B. -U und /i. In diesem Fall wird der letzte benutzt.

Die Parameter geben Sie über folgende Registrierungsschlüssel an:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\'osecmdienst\'
```

Bei der Installation wird automatisch die Zeichenfolge ImagePath mit dem Pfad zum enaio® server als Wert und dem Parameter 'n' mit Dienstnamen eingetragen. Der Standarteintrag hat dieses Schema:

```
C:\OSECM\server\axsvckrn.exe -n 'osecmdienst'
```

Wollen Sie beispielsweise die Bezeichnung des Berichts über den Serverstart um Datum und Uhrzeit ergänzen (ReportDatetime) und die Lizenzinformationen in den Bericht schreiben lassen (ReportLicense), ergänzen Sie den Wert der Zeichenfolge so:

```
C:\OSECM\server\axsvckrn.exe -n 'osecmdienst' -rt -rl
```

Main Thread

Bisher ist nur folgendes geschehen: Der Wunsch des Benutzers wurde verstanden, und dementsprechend wurden alle Switches gesetzt. Jetzt kann die eigentliche Initialisierung stattfinden. Es ist wichtig zu verstehen, dass im Falle des Dienstes der MainThread von SCM verwaltet wird. Dies ist eine ziemlich empfindliche Stelle, weil ein nicht korrektes Beenden des MainThreads dazu führt, dass der Dienst nicht korrekt beendet wird, was nicht der Fall ist, wenn der Server als Anwendung startet.

Der Prozess der Initialisierung besteht aus mehreren aufeinander folgenden Schritten. Jeder Schritt hat eine eigene Nummer (ähnlich einer ID) und einen Namen. Einige Schritte haben nur eine rein technische Bedeutung, andere aber laden Lizenzen, stellen die ServerID fest und setzen die Logik des Systems um. Anzahl und Reihenfolge der Schritte können in der Zukunft geändert werden.

Wenn irgendein Schritt nicht erfolgreich war, wird die Initialisierung abgebrochen, die Nummer des Schrittes als Fehlercode dem Betriebssystem zurückgegeben, und dabei:

- § eine MessageBox angezeigt, falls Verbose auf ON steht.
- § ein WindowsEventLog geschrieben
- § eine Meldung an den Administrator geschickt
- § das Startup-Protokoll geschrieben
- § ein Report geschrieben

Alle Nachrichten haben ein relativ ähnliches Format, und überall wird mindestens die dem Server bekannte Ursache mitgeteilt. Das kann man leicht testen, indem man die ServerID in der Tabelle ändert, und den Server startet.

Die Meldung an den Administrator wird als E-Mail oder als Net-Nachricht gesendet. Dafür sind folgende Einträge in der Registrierung verantwortlich:

§ AdminMail definiert die e-Mail Adresse des Administrators

§ AdminNet ist der Name des Rechners, der die Net-Nachricht erhält.

Die E-Mails werden per MAPI geschickt.

Wenn Initialisierung erfolgreich war und der Server läuft, wird ebenfalls eine Nachricht an den Administrator verschickt. Das gleiche passiert, wenn der Server beendet (nach einem erfolgreichen Start) wird.

Report

In der Reportdatei sind alle Schritte erfasst, die beim Start des Servers abgelaufen sind. Dort stehen auch die (nicht kritischen) Fehler und Warnungen, die den Start des Servers nicht verhindern, aber wichtig sein können z.B. 'Exekutor nicht geladen'. Ein nicht geladener Exekutor wäre ein Fehler. In diesem Fall kann der Kernel weiter laufen, nur muss es der Administrator wissen, sonst werden die Clientprogramme nicht richtig arbeiten können. Außerdem gibt es Warnungen z.B. 'dieser Schritt hat zu lange gedauert'. Der Kernel startet trotzdem, weißt aber auf gewisse Netzwerk-Probleme o.ä. hin. Oder auf bestimmte Systemdateien kann nicht zugegriffen werden. Diese sind zwar in der Tabelle beschrieben, und wenn der Kernel, prüft, ob die Datei lesbar ist, bekommt er negative Antworten von Betriebssystem. Das kann darauf hindeuten, dass die Datei nicht vorhanden ist.

Die Anzahl der Fehler und Warnungen werden in der Nachricht an den Administrator mitgeteilt, so dass man sofort erkennt, ob etwas unternommen werden muss, obwohl der Start erfolgreich war.

Diese Anzahl der Fehler steht natürlich auch in der Reportdatei.

Hier folgt eine Datei, erstellt beim Start des Servers auf dem Server OSDEV

```
===== Protokoll vom Start des AppServers =====
===== ausgef³hrt am 02/09/2002 =====
=====
701 - 14:59:46,723 | MainThread started
    | CommandLine : -n askrn
    | Instance : askrn
    | HasConsole : 0
    | IsService : 1
    | Verbose : 0
    | ReportRegistry : 0
    | ReportLicense : 0
    | ParamFile :
    | Temp Dir : D:\os\server\ostemp\
    | Root Dir : D:\os\server\
    | File Path : D:\os\server\axsvckrn.exe
    | Conf Path : D:\os\server\etc\
    | User Path : D:\os\server\etc\user\
    | ok: 0,0 sec
703 - 14:59:46,723 | CoInitialize
    | ok: 0,0 sec
705 - 14:59:46,723 | CreateMsgQueue
    | ok: 0,0 sec
707 - 14:59:46,723 | ChangeAccess
    | ok: 0,0 sec
709 - 14:59:46,723 | Activate PerfMonitor
    | ok: 0,0 sec
711 - 14:59:46,723 | InitialLoadParams
    | CurrentSchema : 4.0
    | ok: 0,468 sec
713 - 14:59:47,192 | Check OS Version
```



```

    | OS : NT 5.0 - Service Pack 2
    | ok: 0,0 sec
715 - 14:59:47,192 | Init winsock
    | ok: 0,16 sec
717 - 14:59:47,207 | Set Threads context parameters
    | ok: 0,0 sec
721 - 14:59:47,207 | GetIdentStrings
----> 14:59:47,239 | Win32API Error. Function: <NetWkstaUserGetInfo>.
Parameter: <> Error: 0x520
    | Eine angegebene Anmeldesitzung ist nicht vorhanden. Sie wurde
    | gegebenenfalls bereits beendet.
----> 14:59:47,239 | De Server startete unter LocalSystem Account.
M+glicherweise sind nicht alle Netzwerk Resources verf³gbar.
    | Address found : 195.127.121.49, 11D3-00B0D0F0C8CB
    | IP Address : 195.127.121.49
    | GUID : 11D3-00B0D0F0C8CB
    | TCP Port : 4010
    | Computer name : OSDEV
    | Domain name :
    | User name : SYSTEM
    | ok: 0,31 sec
723 - 14:59:47,239 | DBConnect
    | DB Parameter:
    | DSN : peter400ms
    | USER : sysadm
    | Real DB Parameter:
    | DSN : peter400ms
    | UID : sysadm
    | APP : optimal_AS
    | WSID : OSDEV
    | DATABASE : peter40
    | LANGUAGE : Deutsch
    | ok: 0,31 sec
725 - 14:59:47,270 | Server feststellen
    | SELECT id, group_id FROM server WHERE (name = 'OSDEV') AND (instance =
    | 'askrn') AND (port = 4010)
    | ServerID : 5
    | ServerGroupID : 5
    | ok: 0,32 sec
727 - 14:59:47,301 | check resources locked after crash
    | DELETE FROM oslockedres WHERE (sessionguid IN (SELECT ss.sessionguid FROM
    | ossession ss WHERE ss.serverid = 5))
    | DELETE FROM ossession WHERE serverid = 5
    | ok: 0,62 sec
729 - 14:59:47,364 | Resource manager initialization: License
    | ok: 0,78 sec
731 - 14:59:47,442 | Resource manager initialization: system files
----> 14:59:47,442 | Access to File c:\os\server\etc\as.cfg denied: 4
    | Das System kann den angegebenen Pfad nicht finden.
----> 14:59:47,442 | Access to File c:\os\server\etc\aslisten.dat denied: 4
    | Das System kann den angegebenen Pfad nicht finden.
    | ok: 0,16 sec
733 - 14:59:47,473 | Servicename prüfen
    | ok: 0,0 sec
735 - 14:59:47,473 | Expires prüfen
    | ok: 0,0 sec
737 - 14:59:47,473 | Server - Lizenz prüfen
    | ok: 0,0 sec
739 - 14:59:47,473 | Create ConnectThread
    | ok: 0,0 sec
741 - 14:59:47,473 | Create ReadThreads
    | ok: 0,0 sec
743 - 14:59:47,473 | Create ExLoadThread
    | ok: 0,0 sec
745 - 14:59:47,473 | Create SD
    | ok: 0,0 sec
747 - 14:59:47,473 | Create PeriodicBatches
    | osrevisit : abn.CheckOsrevisit, 3000
    | WFM : wfm.WorkerJob, 5000
    | ok: 0,0 sec
749 - 14:59:47,473 | Create PeriodicThread
    | ok: 0,15 sec
751 - 14:59:47,489 | Create JobQueues
    | common : 4, -1
    | wfm : 2, -1
    | deferred : 1, 100

```

```

| ok: 0,0 sec
753 - 14:59:47,489 | Create NameSpaces
| abn : common, oxjobabn.dll
| adm : common,
| dbp : common, oxjobdbp.dll
| krn : common,
| kts : common,
| lic : common,
| std : common, oxjobstd.dll
| tst : common, oxjobtst.dll
| wfm : wfm, oxjobwfm.dll
| ok: 0,16 sec
755 - 14:59:47,504 | Initialize Executors
| abn : ok
| adm : ok
| dbp : ok
| krn : ok
| kts : ok
| lic : ok
| std : ok
| tst : ok
| wfm : ok
| ok: 0,219 sec
757 - 14:59:47,723 | Initialize Performance Collector
| ok: 0,31 sec
763 - 14:59:47,754 | Create JobThread: common, 0
| ok: 1,281 sec
763 - 14:59:49,036 | Create JobThread: common, 1
| ok: 1,266 sec
763 - 14:59:50,301 | Create JobThread: common, 2
| ok: 1,250 sec
763 - 14:59:51,551 | Create JobThread: common, 3
| ok: 1,250 sec
763 - 14:59:52,801 | Create JobThread: wfm, 0
| ok: 1,250 sec
763 - 14:59:54,051 | Create JobThread: wfm, 1
| ok: 1,250 sec
759 - 14:59:55,302 | Create SendThread: 0
| ok: 0,0 sec
761 - 14:59:55,302 | Resume Threads
| ok: 0,0 sec
=====
===== Protokol abgeschlossen
===== Fehler: 0, Warnungen: 4

```

Folglich werden Nummer, Zeit, Ausdauer des Schrittes protokolliert, die Fehler und Warnungen werden vermerkt.

Die Server-Konsole

Wird der Server mit dem Parameter 'console' gestartet, erzeugt der Server eine Konsole, in der alle von ihm ausgeführten Aktionen und alle Fehlermeldungen angezeigt werden. Über die Konsole kann der Benutzer Befehle an den Server schicken/eingeben.

Das Konsolenfenster ist in mehrere Teile gegliedert. Der erste Teil der untersten Zeile des Fensters zeigt den Betriebsmodus der Konsole an. Nach einem Prompt kann in dieser Zeile ein Befehl eingegeben werden. In der vorletzten Fensterzeile wird der Text und Status (Ok oder Fehler) des letzten ausgeführten Befehls angezeigt.

Der obere Teil des Konsolenfensters ist für die Darstellung der Information für den jeweiligen Betriebsmodus vorgesehen.

Die Konsole kann in einen der folgenden Modi ausführen:

§ Flow

| OSDRT450 | | | | |
|---------------------|----------|-----------|----------|-----------------------------------------------------------------------------------------------------|
| Time | ThreadID | User | Task | Message |
| 14:00:55 | c34 | SERVER_R | CNameSpa | Die Funktion CNameSpace::DispatchJob beendet mit dem R³ckgabewert 0x0 |
| 14:00:55 | c34 | SERVER_R | CASThrea | Die Funktion CASThreadJob (workflow0)::InvokeJob beendet mit dem R³ckgabewert 0x0 |
| 14:00:55 | c34 | \$askrn-u | CASThrea | Die Funktion CASThreadJob (workflow0)::ProcessClientJob beendet mit dem R³ckgabewert 0x0 |
| 14:00:55 | c34 | \$askrn-u | CArchivD | Die Funktion CArchivDBFuncPool::FreeDB gestartet mit Eingabeparametern: pDBFunc = 0x5404c10; |
| 14:00:55 | c34 | \$askrn-u | CArchivD | Die Funktion CArchivDBFuncPool::unsafe_FreeDB gestartet mit Eingabeparametern: pDBFunc = 0x5404c10; |
| 14:00:55 | c34 | \$askrn-u | CArchivD | Die Funktion CArchivDBFuncPool::unsafe_FreeDB beendet mit dem R³ckgabewert 0 |
| 14:00:55 | c34 | \$askrn-u | CArchivD | Die Funktion CArchivDBFuncPool::FreeDB beendet mit dem R³ckgabewert 0 |
| 14:00:55 | c34 | \$askrn-u | CASThrea | Die Funktion CASThreadJob (workflow0)::OnEvent beendet mit dem R³ckgabewert 0x0 |
| OK: flow | | | | |
| flow : run nocut \> | | | | |

Ablaufverfolgung und Fehlermeldungen. Folgende Befehle werden im flow-Modus unterstützt:

| | | |
|------------------------|---|--------------------------------------|
| pause oder set pause 1 | - | Meldungsanzeige ausschalten |
| run oder set pause 0 | - | Meldungsanzeige einschalten |
| cut oder set cut 1 | - | Meldung abschneiden |
| no cut oder set cut 0 | - | Meldung in nächster Zeile fortsetzen |
| clear | - | Anzeige in der Konsole löschen |

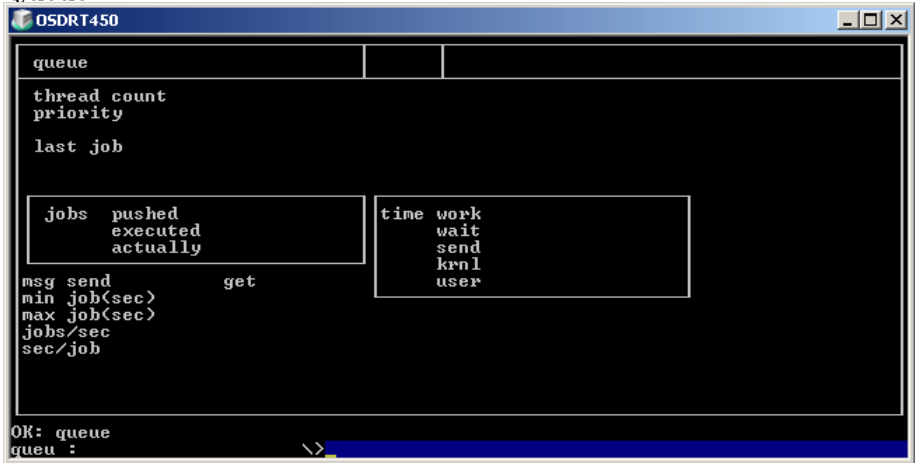
§ Libs

| OSDRT450 | | | | |
|----------|------------------|----------------|-------------------------------------|--|
| | Time | Version | Filename | |
| advapi32 | 04/08/2004 00:57 | 5.1.2600.2180 | c:\windows\system32\advapi32.dll | |
| atl71 | 19/03/2003 05:05 | 7.10.3077.0 | c:\windows\system32\atl71.dll | |
| axsvckrn | 16/08/2005 04:15 | 5.10.5228.732 | d:\os4x\server\axsvckrn.exe | |
| clbcatq | 04/08/2004 00:57 | 2001.12.4414.2 | c:\windows\system32\clbcatq.dll | |
| comctl32 | 04/08/2004 00:54 | 6.0.2900.2180 | c:\windows\winsxs\x86_microsoft.win | |
| comdlg32 | 04/08/2004 00:57 | 6.0.2900.2180 | c:\windows\system32\comdlg32.dll | |
| comres | 04/08/2004 00:57 | 2001.12.4414.2 | c:\windows\system32\comres.dll | |
| crypt32 | 04/08/2004 00:57 | 5.131.2600.218 | c:\windows\system32\crypt32.dll | |
| dbghelp | 04/08/2004 00:57 | 5.1.2600.2180 | c:\windows\system32\dbghelp.dll | |
| dbnetlib | 04/08/2004 00:57 | 2000.85.1117.0 | c:\windows\system32\dbnetlib.dll | |
| dnsapi | 04/08/2004 00:57 | 5.1.2600.2180 | c:\windows\system32\dnsapi.dll | |
| gdi32 | 04/08/2004 00:57 | 5.1.2600.2180 | c:\windows\system32\gdi32.dll | |
| hnetcfg | 04/08/2004 00:57 | 5.1.2600.2180 | c:\windows\system32\hnetcfg.dll | |
| imagehlp | 04/08/2004 00:57 | 5.1.2600.2180 | c:\windows\system32\imagehlp.dll | |
| iphlpapi | 04/08/2004 00:57 | 5.1.2600.2180 | c:\windows\system32\iphlpapi.dll | |
| kernel32 | 04/08/2004 00:57 | 5.1.2600.2180 | c:\windows\system32\kernel32.dll | |
| mfc42 | 04/08/2004 00:57 | 6.2.4131.0 | c:\windows\system32\mfc42.dll | |
| mfc42loc | 23/08/2001 14:00 | 6.0.8665.0 | c:\windows\system32\mfc42loc.dll | |
| msado15 | 04/08/2004 00:57 | 2.81.1117.0 | c:\programme\gemeinsame dateien\sys | |
| OK: libs | | | | |
| libs : > | | | | |

zeigt die DLL an, die vom Server momentan geladen sind. Folgende Befehle werden im libs-Modus unterstützt:

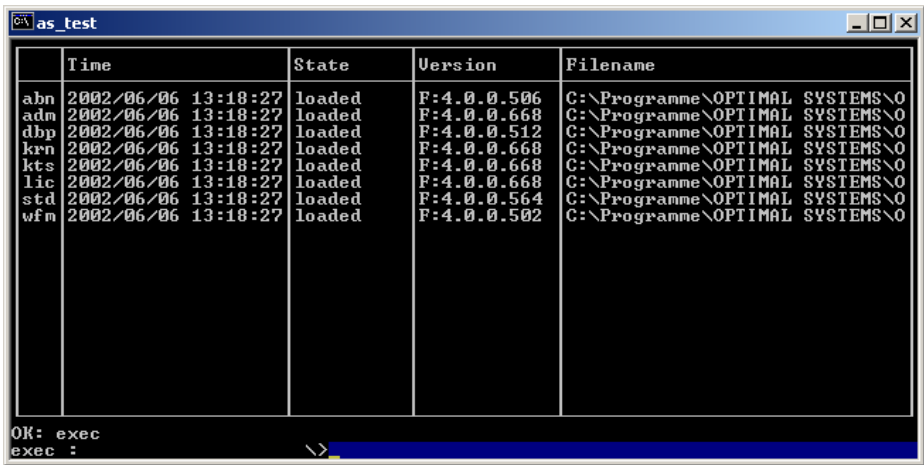
| | | |
|---------------------|---|-----------------------------|
| refresh oder update | - | Liste der DLL wird erneuert |
|---------------------|---|-----------------------------|

§ Queue



zeigt den laufenden Zustand der Jobqueues;

§ Exec



zeigt die Liste der geladenen Exekutoren. Folgende Befehle werden im exec-Modus unterstützt:

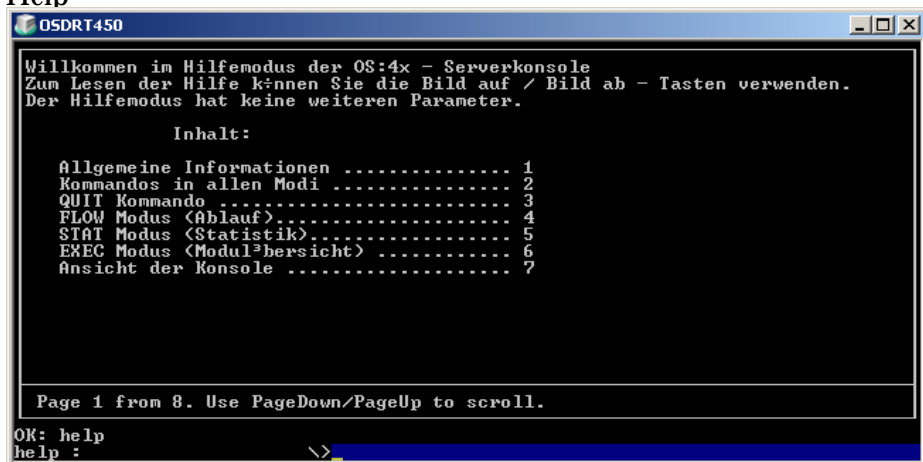
- load <Exekutor> - Laden des <Exekutor>
- unload <Exekutor> - <Exekutor> entfernen

die Liste der geladenen Exekutoren wird nach jedem Befehl aktualisiert;

§ Stat

nicht implementiert

§ Help



Hilfe zu den implementierten Betriebsmodi

Der gewünschte Betriebsmodus kann durch Eingabe des Modusnamens in der Befehlszeile aufgerufen werden. Für den help-Modus kann help, h oder ? eingegeben werden.

In allen Modi sind die Befehle q, quit oder exit erlaubt. Mit diesen Eingaben wird die Konsole beendet und das Fenster geschlossen. Wurde der Server als die gewöhnliche Anwendung gestartet, so wird er mit dem Schließen der Konsole auch beendet. Ansonsten läuft der Dienst weiter.

Die Konfiguration der Konsole (Farbschema, Breite und Darstellung der Informationsfelder, genutzte Codeseite usw.) wird in der ini-Datei axsvckrn.ini eingestellt. Wurde die Datei bei laufender Konsole geändert, so können die Änderungen durch Eingabe von cfg (im jedem Betriebsmodus) für Die aktuelle Darstellung übernommen werden.

Beispiel für axsvckrn.ini:

```
[Console]
CodePage=1251
ActiveMode=flow
```

Checkliste bei Problemen beim Serverstart

Auf folgenden Seiten sollen Hinweise darauf gegeben werden, welche Aktionen unternommen werden können, wenn der enaio® server nicht startet.

Server

Der Anwendungsserver verfügt über diverse Mechanismen, um eine genaue Fehlerdiagnose durchzuführen.

Dazu existieren folgende Varianten:

- § Start über start.bat als Anwendung
- § Fehleranzeige über verbose-Modus
- § Protokollierung der Start- und Shutdownsequenz

Start über start.bat als Anwendung

Im Serververzeichnis befindet sich eine Batchdatei, die den Server als Anwendung startet. Damit läuft der Server unter dem Account des angemeldeten Benutzers. Während des Starts werden die einzelnen Protokollschritte in der Konsole angezeigt.

Startet der Server als Anwendung und nicht als Dienst, so gibt es höchstwahrscheinlich Probleme mit der Berechtigung. Zu Prüfen ist dann der Account, unter dem der Dienst starten soll.

Fehleranzeige über verbose-Modus

Wird der Server mit dem Kommandozeilenparameter -v gestartet, so werden Fehlermeldungen bei der Initialisierung in einer Messagebox dargestellt. Anhand der Fehlerliste kann ermittelt werden, was in der konkreten Situation zu unternehmen ist.

Protokollierung beim Serverstart

Unabhängig von der Protokollierung über die oxrpt.dll können die Initialisierungsschritte in einer separaten Protokolldatei, die in das Serververzeichnis geschrieben wird, mitgeführt werden. Die Datei heißt standardmäßig startup_YYYYMMDD_hhmmss.txt und schreibt die einzelnen Protokollierungsschritte mit. Darin enthalten sind mögliche Fehlerursachen.

Checkliste mit möglichen Aktionen

| Symptom | Mögliche Ursache | Aktionen |
|------------------------------|------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dienst startet nicht | Diverse | Protokoll im OS-Temp-Verzeichnis lesen Fehlermeldung anhand Tabelle unten interpretieren Start.bat ausführen |
| Lizenzfehler | Lizenzdatei nicht vorhanden oder nicht korrekt | Über axlicimpexp aktuelle Lizenzdatei exportieren und prüfen. - Stationskennung, Dienstname, IP-Adresse |
| DB-Fehler | Konfigurationsfehler in der DB, wahrscheinlich Änderung der Datenbank oder Übertragung auf anderen Rechner | Anhand der Einträge Comstring, und IPPort aus der Registrierung (CurrentSchema) + Dienstname wird ein gültiger Eintrag in der Tabelle server gesucht. Ggf. Einträge in der Datenbank oder in der Registrierung anpassen. |
| DB-Connect schlägt fehl | DSN nicht eingerichtet | Prüfen, ob eine System-DSN existiert. Name steht in dem DB-Abschnitt der Registrierung |
| | Benutzername/Passwort falsch | Ändern des Passwortes mit Axchgpwd.exe |
| | Fehlende Netzwerkrechte | Ändern des Accounts, unter dem der Dienst läuft |
| Jobs schlagen teilweise fehl | Exekutoren werden nicht geladen | Über die Console prüfen, ob die Exekutoren geladen sind. Alternativ kann das Startup-Protokoll analysiert werden. |

Fehlercodes beim Start des Servers

| Returncode | Int | Initialisierungsschritt | Beschreibung | Mögliche Ursache/Aktivität |
|--------------------------|-----|------------------------------------|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SRETCODE_START | 701 | MainThread started | Hauptthread wird gestartet, interne Variablen werden initialisiert | à DEV |
| SRETCODE_COINITIALIZE | 703 | CoInitialize | Windows COM-Bibliothek wird initialisiert | Service Pack prüfen, Rechte des Serveraccounts prüfen, ansonsten à DEV |
| SRETCODE_CREATEMSGQUEUE | 705 | CreateMsgQueue | Interne Initialisierung der Messagequeue | à DEV |
| SRETCODE_CHANGEACCESS | 707 | ChangeAccess | Initialisierung der Windows Prozess-Security | Account für den Dienst prüfen (lokale Adminrechte vergeben) |
| SRETCODE_ACTIVATEPERFMON | 709 | Activate PerfMonitor | Aktivierung des Performance-Monitors | à DEV |
| SRETCODE_LOADPARAMS | 711 | InitialLoadParams | Registrierungsparameter werden geladen | Registrierung prüfen Schlüssel: Schemata, Current Schema (Value), Current Schema (Schlüssel) müssen existieren. Rechte zum Lesen und Schreiben der Registrierung prüfen |
| SRETCODE_CHECKOS | 713 | Check OS Version | Prüfung des Betriebssystems | à DEV |
| SRETCODE_INITWINSOCK | 715 | Init winsock | Initialisierung der Netzwerkschicht | Servicepack, TCP IP-Einstellungen prüfen |
| SRETCODE_THREADCONTEXT | 717 | Set Threads context parameters | Interne Initialisierung | à DEV |
| SRETCODE_CREATECONSOLE | 719 | Create Console | Erstellung der Konsole, falls in den Startparametern angegeben | Prüfen der Einstellung im Dienstmanager 'Interaktiven Datenaustausch mit dem Desktop erlauben' |
| SRETCODE_IDENTSTRINGS | 721 | GetIdentStrings | Ermittlung des Computernamens, IP-Adresse, GUID, Usernamen und Domainnamen | SP prüfen, Netzwerkkonfiguration prüfen, Account prüfen |
| SRETCODE_DBCONNECT | 723 | DBConnect | Datenbankverbindung wird hergestellt | DSN, Registrierung (DB-Parameter), Connectivity, Account prüfen |
| SRETCODE_SERVERID | 725 | Server feststellen | ServerID und GroupID wird aus der Datenbank gelesen | Tabellen Server und ServerGroup auf Konsistenz prüfen. Ggf. Registrierung (ComString, Port und Dienstname sind relevant) s. SQL-Statement im Report |
| SRETCODE_CHECKRES | 727 | check resources locked after crash | Nicht freigegebene Ressourcen nach dem letzten möglichen Servercrash werden freigegeben | Datenbankconnection prüfen, OSLockedRes in der Struktur prüfen, s. SQL-Statement im Report |

| | | | | |
|----------------------------------|-----|--------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SRETCODE_RE SMGRLIC | 729 | Resource manager initialization: License | Auslesen der Lizenz, Entschlüsselung der Lizenz | Lizenz möglicher falsch importiert, Tabelle oslicense prüfen, ggf. Lizenzdatei mit axlicimpexp exportieren und prüfen |
| SRETCODE_RE SMGRFILES | 731 | Resource manager initialization: system files | Systemdateien aus OS- Ressources werden ermittelt und auf deren Existenz geprüft | Tabelle osresources prüfen, ggf. die Pfadeinstellungen für die Resources mit Type = 1 prüfen |
| SRETCODE_C HECKNAME | 733 | Check Servicename | Servicename aus der Lizenz wird mit realem Dienstnamen verglichen | Lizenz und Dienstname Dienstmanager (Eigenschaften) |
| SRETCODE_C HECKEXPIRES | 735 | Check Expires | Ablauf der Lizenzdatei wird geprüft | Lizenzdatei und Systemzeit prüfen |
| SRETCODE_C HECKLICENSE | 737 | Server - Lizenz prüfen | Serverlizenz wird geprüft | Zuweisung der Station für den Server in der aslic.cfg/im axliccfg.exe oder in der Tabelle oslicresources prüfen, ggf. Stationsnummer ändern |
| SRETCODE_CR EATECONNEC T | 739 | Create ConnectThread | Initialisierung des Connectthreads zum Empfang von Clientconnections | TCP-Einstellungen und prüfen, ob Port besetzt ist (z.B. durch bereits laufende Instanz des Servers) |
| SRETCODE_CR EATEREAD | 741 | Create ReadThreads | Lesethreads werden erstellt | à DEV |
| SRETCODE_CR EATEEXLOAD | 743 | Create ExLoadThread | Exekutor-Ladethread wird erstellt | à DEV |
| SRETCODE_CR EATESD | 745 | Create SD | Security Descriptor wird erstellt | Account prüfen, dannà DEV |
| SRETCODE_CR EATEPEBATCH ES | 747 | Create PeriodicBatches | Periodische Jobs werden initialisiert | Registrierung [CurrentSchema]\Batc hes prüfen à DEV |
| SRETCODE_CR EATEPERIODI C | 749 | Create PeriodicThread | Thread für Timer wird erstellt | à DEV |
| SRETCODE_JO BQUEUES | 751 | Create JobQueues | JobQueues werden je nach Registrierungseinstellun gen erstellt | Registrierung [CurrentSchema]\Que ues\ prüfen |
| SRETCODE_N AMESPACES | 753 | Create NameSpaces | Namespaces werden aus der Registrierung gelesen und initialisiert | Registrierung [CurrentSchema]\Na meSpaces\ prüfen Queue zugewiesen, DllName korrekt |
| SRETCODE_IN ITEXEC | 755 | Initialize Executors | Exekutoren werden initialisiert | Prüfen, ob die DLLs vorhanden sind, Version der DLLs muss dem Kernel entsprechen |
| SRETCODE_IN ITPERFCOLL | 757 | Initialize Performance Collector | Interne Initialisierung der Systemumgebung | à DEV |
| SRETCODE_CR EATESEND | 759 | Create Send thread | Threads für Notifications werden initialisiert | à DEV |
| SRETCODE_RE SUME | 761 | Resume threads | Threads werden fortgesetzt (waren vorher noch nicht gestartet) | à DEV |
| SRETCODE_CR EATEJOB | 763 | Create Job threads | Jobthreads für Bearbeitung werden erstellt | Registrierung prüfen Registrierung [CurrentSchema]\Que |

| | | | |
|--|--|--|-----------------------------------------------------------|
| | | | ues\ prüfen (NumberOfThreads und Priority) à DEV |
|--|--|--|-----------------------------------------------------------|

Zu bemerken ist, dass die Nummer der Schritte gleichzeitig die Rückgabewerte sind, die der Service Control Manager anzeigt, wenn der Dienst nicht gestartet werden konnte.

| Returncode | Int | Beschreibung | Ursache |
|-------------------------------|-----|---------------------------------------------------------------------------|-----------------------------------------------|
| SRETCODE_REPORTINIT | 790 | Protokollierungssystem konnte nicht aktiviert werden | Oxrpt.dll nicht vorhanden, nicht konfiguriert |
| SRETCODE_SERVICEINDEBU G | 791 | Der Dienst kann nicht in Debugmodus gestartet werden | Debug - exe mit release ersetzen |
| SRETCODE_WRONGSYSTEM | 792 | Der ArchivServer ist auf dem vorhandenen Betriebssystem nicht ausführbar. | Betriebssystem prüfen. |
| SRETCODE_SYSTEMFAILED | 793 | Der Aufruf einer Standardfunktion gab einen Fehler zurück | Reportdatei an DEV |
| SRETCODE_ALREADYRUNNI NG | 794 | Ein Server mit gleichem Dienstnamen läuft bereits | Taskmanager prüfen, Registrierung prüfen |
| SRETCODE_INVALIDPARAME TER | 795 | Command Line Parameter sind inkorrekt | Kommandozeile prüfen |
| SRETCODE_TLSSLOT | 796 | Tls Slot konnte nicht ermittelt werden | -> DEV |
| SRETCODE_TAKEINSTANCEN AME | 797 | Der Dienstname konnte nicht gesperrt werden | -> DEV |
| SRETCODE_GETBREAK EVEN T | 798 | Der Break-Event konnte nicht erstellt werden | -> DEV |

Adminmail und net-send

Der enaio® server kann nun beim Starten und Beenden des Dienstes dem enaio® administrator eine Benachrichtigung übermitteln. Das funktioniert entweder per E-Mail oder per Net-send. Wird die Benachrichtigung per E-Mail versandt, so ist als Attachment die Startup.txt und die shutdown.txt mit beigefügt. Der E-Mail-Versand kann wahlweise über MAPI (Installation eines MAPI-Clients erforderlich) oder per SMTP erfolgen.

Um dies am Server einzustellen, müssen folgende Registrierungseinträge vorgenommen werden. Standardmäßig sind keine Werte gesetzt, d.h. wenn kein Eintrag vorhanden ist, dann erfolgt keine Benachrichtigung.

Unter dem Schlüssel:

HKEY_LOCAL_MACHINE\SOFTWARE\Optimal Systems\ASArchive\Schema\4.0

| Zeichenkette | Wert | Beschreibung |
|--------------|--------------------------------|--------------------------------------------------------------------------------------------------------------------|
| AdminMail | z.B. reibis@optimal-systems.de | E-Mail-Adresse des Empfängers |
| AdminNet | z.B. reibismobil | Rechnername des Empfängers für net send |
| MailSender | z.B. robert | Absender der Mail, muss beim Versenden der Mail angegeben werden |
| MailServer | z.B. osapp | Name des Mailservers. Wird dieser Name nicht angegeben, so wird versucht, über MAPI zu senden, ansonsten über SMTP |

Net-send-Nachricht beim Shutdown:

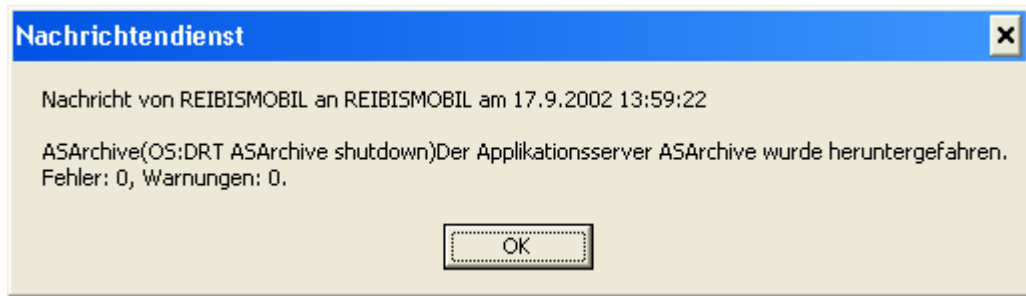


Abbildung 60

Net-send-Nachricht beim Startup:

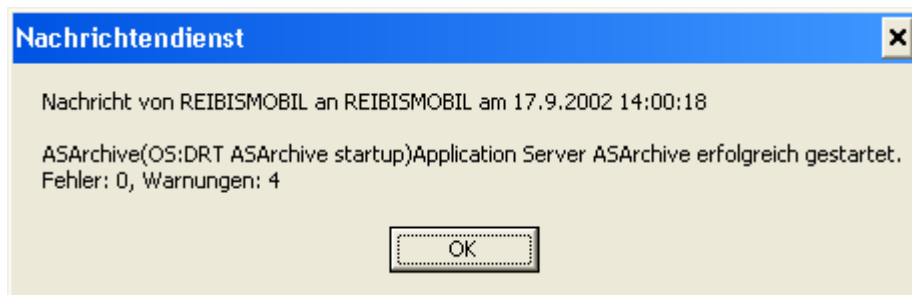


Abbildung 61

Sprache der Oberfläche in enaio® umschalten

Die folgende Liste fasst zusammen, wie in den verschiedenen enaio®-Komponenten die Sprache der Oberfläche umgestellt wird. Nicht aufgeführte Komponenten verfügen über keine Sprachumschaltung.

| enaio® Komponente | Sprachumstellung |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| enaio® client ax.exe | <p>Die Sprache wird in enaio® client im Einstellungsdialog festgelegt, den Sie auf der Menübandregisterkarte ENAIO über die Funktion Einstellungen öffnen. Die Sprache wählen Sie über Arbeitsbereich 'Sprache wählen...'. Die Sprache der verwendeten Objektdefinition (sofern eine Übersetzung vorliegt) wird im gleichen Dialog unter Sprache der Objektdefinition festgelegt.</p> <p>Bitte beachten Sie, dass einige Teile der Oberfläche erst nach dem Neustart der Anwendung übersetzt werden und andere vom Windows Betriebssystem abhängig sind und somit in der Sprache des verwendeten Windows angezeigt werden.</p> <p>enaio® client steht auch als Webanwendung zur Verfügung (enaio® webclient).</p> <p>Die Sprache von enaio® webclient wird über die Spracheinstellungen im MS Internet Explorer über Extras > Internetoptionen > Allgemein > Sprachen festgelegt. Ist die Sprache des Betriebssystems weder Deutsch noch Englisch,</p> |

| | |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | wird enaio® webclient standardmäßig auf Englisch angezeigt. |
| enaio® administrator axadmin.exe | Das Programm folgt den Spracheinstellungen von enaio® client. |
| enaio® enterprise-manager osecm_entmgr.msc | Die Sprache wird über die Datei oxentmgr.cfg festgelegt, die sich im Verzeichnis ..\clients\admin befindet. Um die Sprache umzuschalten, muss das Semikolon vor dem Eintrag LOCALIZATION der jeweiligen Sprache entfernt und vor dem LOCALIZATION Eintrag der aktuell zu verwendenden Sprache gesetzt werden. |
| enaio® editor axgredit.exe | Das Programm folgt den Spracheinstellungen von enaio® client. |
| enaio® capture asindex.exe | Die Sprache wird auf der Menübandregisterkarte ENAIO über die Funktion Arbeitsbereich festgelegt. |
| enaio® editor_for Workflow axwfedit.exe | Das Programm folgt den Spracheinstellungen von enaio® client. |
| enaio® workflow-administrator axwfadm.exe | Das Programm folgt den Spracheinstellungen von enaio® client. |
| enaio® office-utilities oxvbofficeutil.dll | Die Sprache der Office Utilities wird in den MS Office Programmen gewählt. Auf der enaio® Multifunktionsleiste finden Sie das Symbol Spracheinstellungen, das die Sprachauswahl ermöglicht. |
| enaio® document-storage axvdocstorage.exe | Die Sprache wird über das Kontextmenü des Taskleistensymbols unter Sprache festgelegt. |
| enaio® manager-for-logfiles axprotocolcfg.exe | Für die Protokollkonfiguration wird die Sprache über ? > Sprache wählen... festgelegt. Danach muss das Programm neu gestartet werden. |
| enaio® protocol-viewer axrptview.exe | Die Sprache wird im Protokolldatei-Viewer über Ansicht > Sprache wählen... festgelegt. |
| enaio® file-trigger axfiletrigger.exe | Das Programm folgt den Spracheinstellungen des Windows Betriebssystems. Ist die Sprache des Betriebssystems weder Deutsch noch Englisch, wird enaio® file-trigger standardmäßig auf Englisch angezeigt. |
| enaio® file-system-archiver axfsarch.exe | Das Programm folgt den Spracheinstellungen von enaio® client. |
| enaio® contentviewer | Die Sprache wird über die Spracheinstellungen im MS Internet Explorer über Extras > Internetoptionen > Allgemein > Sprachen festgelegt. |
| enaio® server | Die Sprache kann über einen Registrierungseintrag auf Englisch umgestellt werden: HKEY_LOCALE_MACHINE/SOFTWARE/OPTIMAL SYSTEMS/'DIENST'/locale=1033 |

