



Deliverable Identification Sheet

Contract No.:	EDC- 22225
Project full title	NEDINE - Intelligent News Distribution Network Network for multinational business news exchange and dissemination

Deliverable number/name	<i>D-6.2 Personalized Business News Distribution System</i>
Security (distribution level)	<i>Confidential</i>
Contractual date of delivery	<i>T22 = February 2006</i>
Actual date of delivery	<i>February 27, 2006</i>
Type	<i>Demonstrator</i>
Status & version	<i>Final, Annotation to Demonstrator</i>
Number of pages	<i>49</i>
Deliverable contributing to WP	<i>WP 6</i>
Author(s)	<i>Markus Schranz, PTE</i>
WP / Task responsible	<i>PTE</i>
Other contributors	<i>SITA, CIA, UPM, TUV, NVC, HPR</i>

Abstract (for dissemination)

The heterogeneous service descriptions identified in the early stages of the Nedine project have been identified as a major obstacle to the internationalisation of news delivery services. As expected in advance, Nedine has scheduled work package 6 to distinctly focus on the news delivery mechanisms and services to unify, create and design a modern, standardized mechanism and software to scaleably distributed personalized business news throughout a pan-European network.

As proposed in the technical Annex I to project NEDINE, the objectives of WP6 are to define adequate mechanisms to distribute business news to journalists and opinion leaders, to optimize bulk information transfer for hundreds of thousands of addresses in a real-time environment, to provide personalized content filtering for a maximum of potential subscribers, and to identify future delivery channels to utilize modern technology throughout Europe.

Based on available and affordable technology, NEDINE is targeted at delivering to a maximum quantity and quality of subscribers. To provide easy and wide-spread access throughout all European countries, the delivery medium is defined as the Electronic Mail Service (X.400, currently SMTP).

This document annotates textually the software algorithms, implementation and web-based subscription interfaces that have been created as Nedine's deliverable D6.2.

Table of Contents

1	Executive Summary	4
2	Version history	7
3	Introduction	8
4	Motivation	10
4.1	Technical preparations and preparatory project work.....	11
5	Personalized Business News Distribution System	12
5.1	Personalized business news distribution algorithm	12
5.1.1	Algorithm excerpt to business mail delivery system.....	13
5.2	Using the business news delivery system	15
5.2.1	Business news delivery service usage at PTE.....	16
5.2.2	Business news delivery service at CIA	22
5.2.3	Business news delivery service at SITA.....	25
6	Summary	28
7	Attachment.....	29
	Example implementation of Personalized Business News Distribution System at presstext.....	29

1 Executive Summary

One of the initially defined objectives of the project NEDINE has been *Objective II: Direct access to the “desktop” of opinion leaders and decision makers*. As described in the Technical Annex, the NEDINE project will focus to distribute news not only to journalists of European Newspapers, but also directly to the desktop of opinion leaders. This provides direct access for information distributors and independency for the recipients.

The abstractly described goal was in focus of several work packages, but with major emphasis on news distribution technology, the consortium addressed the news distribution goal in work package 6. The objectives of WP6 are to define adequate mechanisms to distribute business news to journalists and opinion leaders, to optimize bulk information transfer for hundreds of thousands of addresses in a real-time environment, to provide personalized content filtering for a maximum of potential subscribers, and to identify future delivery channels to utilize modern technology throughout Europe.

During the project requirements analysis in the first months the news agency partners, which employ different news presentation and distribution mechanisms, identified common marketing and business goals and described technical requirements and services to meet the above described goals on an international level. By managing the subproject “news distribution service and framework description” in work package 6 the consortium has identified, designed and implemented a powerful, scaleable and integrative news distribution service for a modern news exchange and distribution network in Europe, as described in the Technical Annex of the project Nedine.

This document acts as an annotation to the joint technical development of a news distribution service. The administrative and technical concept to organize and manage customers, handle information categories, structurize, format and compile business news for individually composed business distribution units, bulk email distribution and further details are described in

the foreseen report D6.1 “Multi-Platform Business News Distribution Framework”, which is due end of March 2006. The technical implementation of the news distribution system mainly focuses on high performance and high volume mail distribution features and the creation of individually tailored subscriber news, based on personal profiles and interest areas.

The technical result is a set of mechanisms and a software description which has been implemented at all news agency partner sites and is available in production service since January/February 2006 at PTE, SITA and CIA. Within this annotation document we include screenshots of the Web interfaces to the news distribution service and attach source codes of an example implementation.

.....

Document information

Document ID	WP6_PersBusinessNewsDistribution_0.1
Title	Personalized Business News Distribution System
Short title	PersBizNewsDistribution
Workpackage	6
Deliverable	D6.2
Subject	Description of the personalized business news distribution on local Nedine partner services
Description	Annotation of technical implementation of local news distribution services, used methodologies and interfaces. The focus is on the unification and standardization of services in order to prepare local distribution systems for international integration in Nedine.
Security level	public

Table 1: Document information

2 Version history

Version	Date	Author(s)	Description
0.1	January 14, 2006	Markus Schranz	First prototype implementation & description at PTE, based on technical meeting in Bratislava, Jul 15, 2005
0.2	January 30, 2006	Markus Schranz, Robert Tschögl, Tomas Barta, Tomas Tuleja	Implementation of local deliver services at PTE, CIA, SITA; upgrades of system according to technical meeting January 27, 2006 in Vienna
0.3	February 13, 2006	Markus Schranz, Tomas Barta, Tomas Tuleja	Integration of current data model for news subscribers and news distribution items as updated in news distribution framework (to be described)
1.0	February 27, 2006	Markus Schranz	Integration of screenshots of local access to news delivery system of all partners. Annotation of running service as a prototype for all participating and future network partners

Table 2: document version history

3 Introduction

In order to fulfill the specifications of the Technical Description of the Nedine project according to the contents of workpackage 6, the partners involved in technical development have conducted a contiguous project work, initiated at the start of the News Distribution workpackage in July 2005 up to the publication and deployment of the market-ready product “news distribution system” in January 2006.

As proposed in the technical Annex I to project NEDINE, the objectives of WP6 are to define adequate mechanisms to distribute business news to journalists and opinion leaders, to optimize bulk information transfer for hundreds of thousands of addresses in a real-time environment, to provide personalized content filtering for a maximum of potential subscribers, and to identify future delivery channels to utilize modern technology throughout Europe.

Based on available and affordable technology, NEDINE is targeted at delivering to a maximum quantity and quality of subscribers. To provide easy and wide-spread access throughout all European countries, the delivery medium is defined as the Electronic Mail Service (X.400).

The described result, which is the core of Deliverable D6.2, is the “Personalized Business News Distribution System”, which is a collaboratively designed and implemented software apt to fulfill the goals defined in work package 6. This document acts as textual annotation to the software which has been designed and implemented since July 2005 until its deployment within the news agency partner services in January resp. February 2006.

The documentation within this annotation to Deliverable 6.2 includes the following parts: The motivation to jointly develop a business news delivery system, the access to control the system via comfortable web interfaces, the results visible for the customers (such as email compositions and screenshots of typical business news deliveries), and the documentation and annotation of most relevant source code sections of the personal news delivery system and the used algorithms in order to prepare future network partners to participate in the Nedine network.

Conceptual details, subscriber management or news categorization and management are not part of this system annotation, since they will be separately described in D6.1 “Multi-Platform Business News Distribution Framework”.

4 Motivation

The heterogeneous service descriptions identified in the early stages of the Nedine project have been identified as a major obstacle to the internationalisation of news delivery services. As expected this in advance, Nedine has scheduled work package 6 to distinctly focus on the news delivery mechanisms and services to unify, create and design a modern, standardized mechanism and software to scaleably distributed personalized business news throughout a pan-European network.

As proposed in the technical Annex I to project NEDINE, the objectives of WP6 are to define adequate mechanisms to distribute business news to journalists and opinion leaders, to optimize bulk information transfer for hundreds of thousands of addresses in a real-time environment, to provide personalized content filtering for a maximum of potential subscribers, and to identify future delivery channels to utilize modern technology throughout Europe.

Based on available and affordable technology, NEDINE is targeted at delivering to a maximum quantity and quality of subscribers. To provide easy and wide-spread access throughout all European countries, the delivery medium is defined as the Electronic Mail Service (X.400, currently SMTP).

Business news are most relevant at the time they are created. Therefore business news delivery is handled within real-time environments. Time to receive news is critical for a significant part of the addressed customers. To optimise the intelligent news network for a maximum of potential subscribers, the service provides several levels of news delivery modes: Immediate delivery, grouped by news categories, group by time spans, etc. A dedicated database and a web based manipulation interface have been designed and implemented to manage user details and complex queries for addressee generation. In order to optimize the value for the service users, each consumer will be able to specify a personalized profile for receiving contents. The profile manages international distribution attributes, national preferences, language preferences, categorisation, keywording, news relation types, etc. which will be described in detail in D6.1 "Multi-Platform Business News Distribution Framework".

4.1 Technical preparations and preparatory project work

As scheduled in the project plan, important basic work in the area of data management, storage formats and user classification has been provided by other work packages and documented in preceeding deliverables.

The data contents that have to be integrated in business news delivery have been described in detail in D3.1 “Business Data Repository and Management Description”, where NewsML has been identified as the most up-to-date standardized content format for business news.

In D4.1 “Description of User types, roles and network workflows” the user class of business news subscribers has been defined and typical workflows on how to subscribe content areas and create personalized profiles have been outlined. Finally, the web-based interfaces have been integrated and documented also for the purpose of controlling and administrating the news delivery portions of the Nedine service in D5.1 “Functional Web Service Prototype”.

In this document, we focus on technical functionality and the annotations to the interfacing to the implemented service “business news distribution system”, as it appears to the customers of the involved news agency partners PTE, SITA and CIA.

5 Personalized Business News Distribution System

The main actor described in this document is the software itself, the purpose and type of this deliverable is a demonstrator.

Nevertheless, within this written annotation the consortium would like to annotate the software in order to document important sections and make special features visible.

Besides the fact, that the jointly developed software is up and running at all news agency partner sites since beginning of 2006, the annotational description in the following sections include

- Algorithm and methodology details of the news distribution software
- Documentation of multinational user interfaces for news subscribers (both administrating subscriptions and receiving business news)
- Screenshots of personalized business news as they appear exemplarily in the mail folders of the subscribers.

A full concept description, user management, software architecture and integration will be presented in the scheduled framework documentation and description D6.1 “Multi-Platform Business News Distribution Framework” in March 2006.

5.1 Personalized business news distribution algorithm

This section explains and annotates most relevant portions of the jointly developed software code for the personalized news distribution system.

As initiated in the technical meeting from July 15, 2005 in Bratislava, the technical experts at the participating news agencies together with the experts from Technical University of Vienna and UPM have started a code review of all existing mail services at the current system implementations.

According to commonly identified service features, the following design phase identified basic algorithms and methodologies that are to be used locally as well

as internationally in the future system that can be integrated in the proposed Nedine network.

Further asynchronous work and multiple bilateral technical meetings in Vienna, Prague, and Bratislava have supported the mutual development of the technical news delivery system which has been launched at all three agencies in January resp. February 2006.

The software is based on algorithms that involve

- The personalized user profiles that identify news categorization preferences and personal keyword associations
- A time period in which business news are identified to be delivered to users according to their profiles
- Format preferences of subscribers, allowing plain text, HTML or modern XML news distribution (including different end user platforms and devices)
- Language specifications in order to distinguish multilingual contents and create single-language news portions per subscriber

The software at all partner services is implemented in different programming languages, corresponding the local development environments used at the partner sites. Nevertheless, the system follows common algorithm specifications which are outlined briefly in the following subsection.

5.1.1 Algorithm excerpt to business mail delivery system

As preconditions for the algorithm, the consortium identified the following characteristics:

- The algorithm must be capable of handling multilingual business news items, but deliver only single language documents
- The algorithm must be scaleable in terms of number of subscribers and subscriber profiles

- The algorithm must be scaleable in terms of total number of business news deliverables (single mails or “mailing digests” as the consortium calls the composition of personalized news compositions for a single subscriber within the specified time period).
- The algorithm must prepare statistical evaluations such as acknowledgements for news providers and numeric evaluations
- The algorithm must take care of modern mailing formats and personalized subscriber preferences to delivery formats such as HTML, plain text, XML variants, etc.
- The algorithm must meet time, hardware, and network constraints in order to handle mailing distribution within a hard real time environment. Business customers only accept services that timely distribute the contents to the addressed customers.

The resulting algorithm, identified and defined by the technical experts of PTE, SITA and CIA under consultancy of research experts from TUV and UPM can be outlined as follows

- a) a) Create database connection for user profiles, news items and local service parameters; prepare statistical logging;
- b) Set static content sections (headers, design, national assets)
- c) Identify all content categorization and timing constraints from DB and from environment
- d) Read all subscribers according to timing constraints and content categorization
 - a. Identify subscribers interest areas
 - b. Identify subscribers timing preferences
 - c. Identify subscribers format preferences
- e) Read all business news within the timing and language constraints

- a. All business news items within observed time period
- b. All business news items in selected language
- f) Iteration
 - a. For all identified subscribers do...
 - For all identified news items in time period do...
 - Identify association subscriber \leftrightarrow news
 - Based on category or keyword
 - Create personalized business news section
 - Prepare personalized digest based on format preference
 - b. Create statistical entries for evaluation
- g) Finish personalized business news distribution; Print statistical data (numeric evaluation, time duration, logging for business purposes, forensics)

5.2 Using the business news delivery system

The software is part of the backend services of the involved news agencies, so direct triggering of news delivery processes by humans is not foreseen. Typically, business news are delivered event-triggered, i.e. whenever a news item has been entered at the Web-based content management system, the backend service might deliver an email-based message to the subscribers. Although this is also part of the service of the participating agencies, we focus on the periodic, time-triggered delivery of personalized business news.

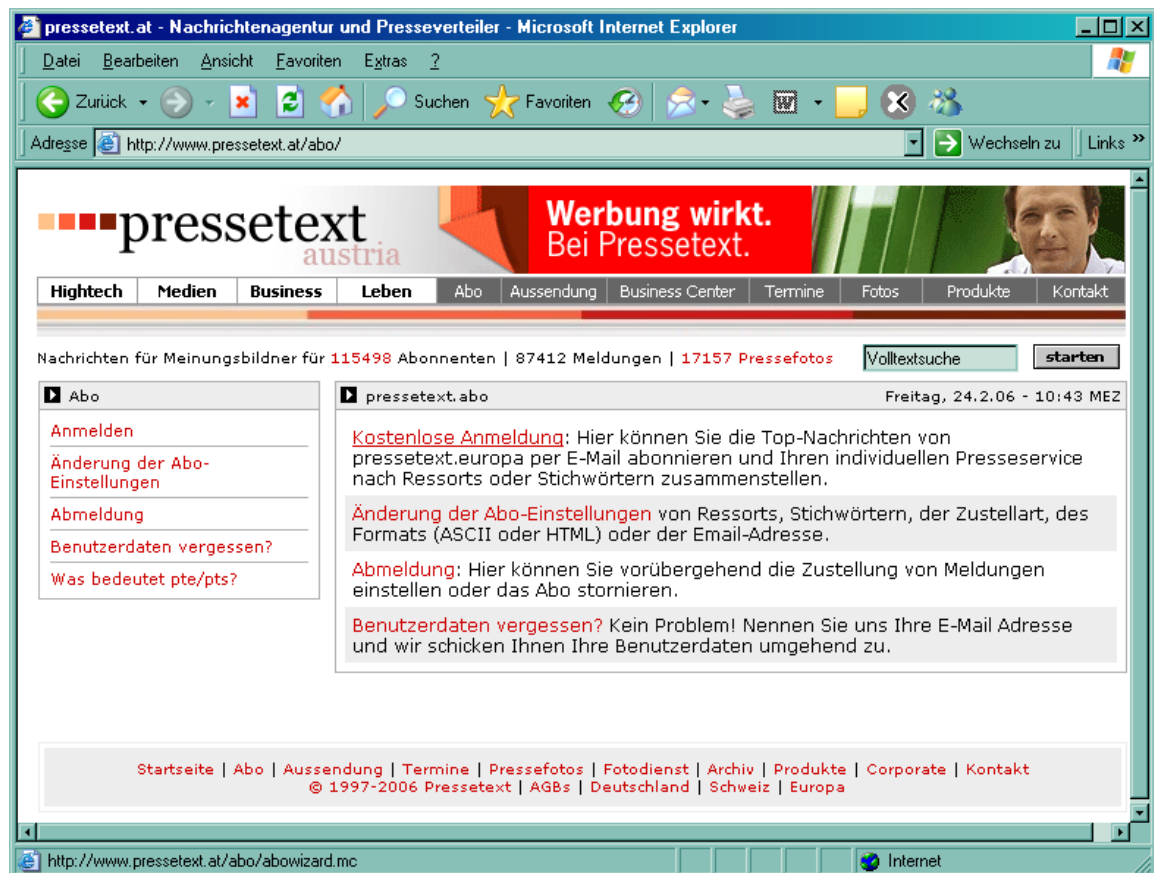
For the latter purpose, customers can enter personal preferences of content categories or keywords and the service automatically decides on specific points in time when business news from a specific time period and of a specific language are delivered to the mass of subscribers.

As a consequence, the major features of the personalized business news delivery system and especially the execution initiation and progress can not be seen by the subscribers or customers, but is managed fully automated or semi-automated with support of technical administration experts.

Since administration interfaces are local service details and out of scope of the international Nedine project, this annotation document aims at providing the small portions of user interfaces visible to the subscribers of the news distribution service. The interface includes the service registration for personalized news distribution and the resulting email messages that can be received individually from the service.

5.2.1 Business news delivery service usage at PTE

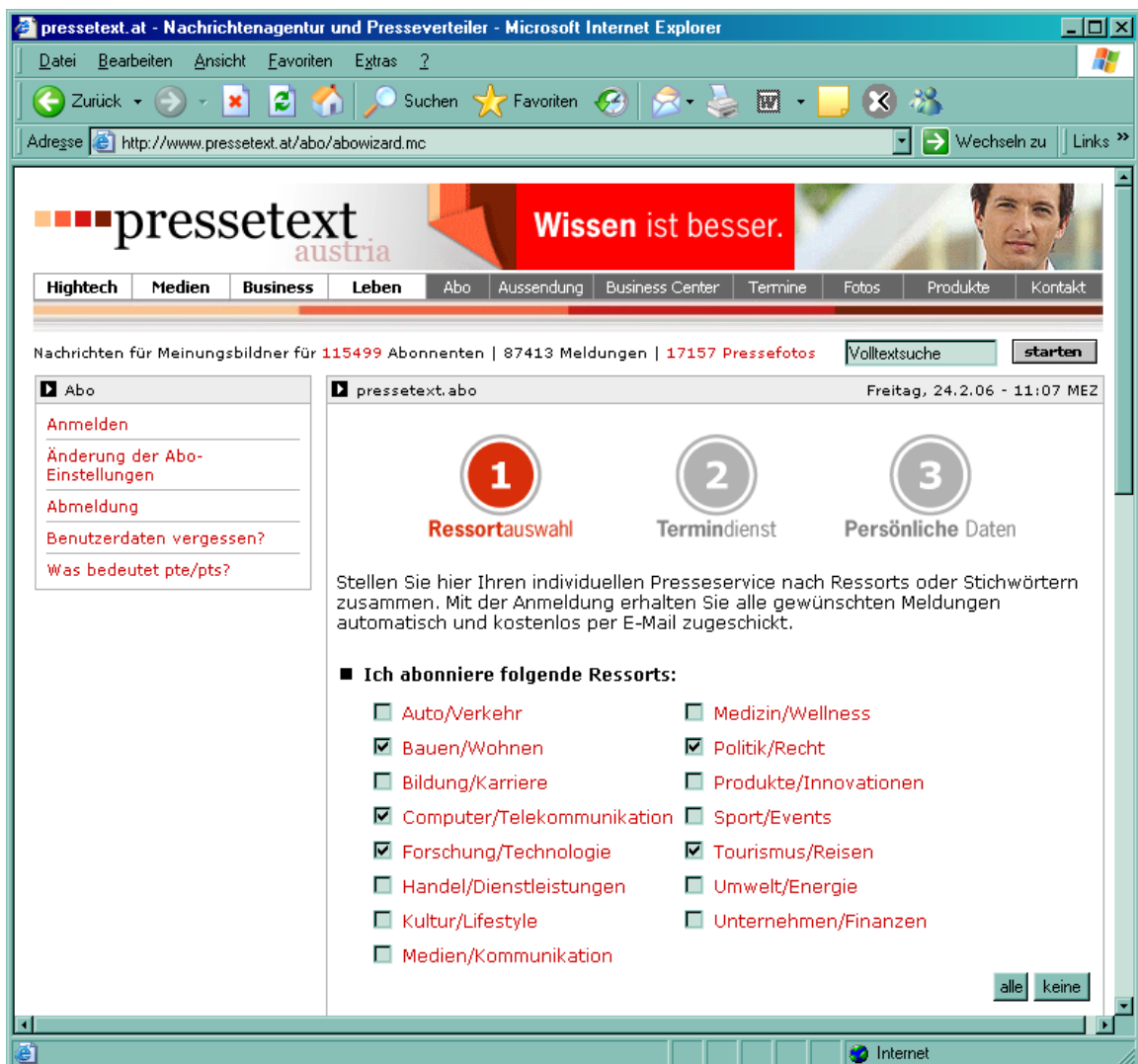
At presstext, the customers can enter their personalized profile from the Web service at <http://www.presstext.at/abo/>



Screenshot 1: Subscriber services at presstext (PTE)

As shown in Screenshot 1, presstext subscribers can choose between subscribing to the personalized news delivery system, removing personalized data from the database, set temporarily offline status or ask for access data to the web-based administration system.

On selection of the subscription service, the potential subscribers follow a multi-level wizard to enter personalized data. Screenshot 2 shows the selection process of content categories, which are stored individually for all subscribers in order to guarantee personalized content selection.



Screenshot 2: Entering preferred news categories

Screenshot 3 requires the presentation of personalized data in order to address the subscriber (email address) and use personalized items in the news delivery (name, gender, address regions, etc.)

The screenshot shows a web browser window with the address bar displaying 'http://www.presstext.at/abo/abowizard.mc'. The page title is 'presstext.at - Nachrichtenagentur und Presseverteiler - Microsoft Internet Explorer'. The main content area is titled 'persstext.abo' and features three numbered steps: 1. Ressortauswahl, 2. Termindienst, and 3. Persönliche Daten. The third step is highlighted in red. Below the steps, there is a notice: 'Ihre Daten werden von uns vertraulich behandelt. Sieben mit rotem Sternchen gekennzeichnete Felder sind auszufüllende Pflichtfelder und dienen Ihrem persönlichen Zugang sowie der regionalen Zuordnung für den Presseversand.' The form is divided into two sections: 'Persönliche Daten' and 'Benutzerdaten (frei wählbar)'. The 'Persönliche Daten' section includes fields for 'Anrede' (Herr), 'Titel', 'Vorname*' (Markus), 'Nachname*' (Schranz), 'Medium/Unternehmen/Institution' (presstext Nachrichtenagentur GmbH), 'Land*' (Österreich), 'Plz*' (1080), 'Ort' (Wien), 'Adresse' (Josefstädter Strasse 44), 'E-Mail*' (schranz@presstext.at), 'Journalist(in)' (Nein), and 'Nutzung' (beruflich). The 'Benutzerdaten' section includes fields for 'Benutzername:*', 'Passwort:*', and 'Passwort wiederholen:'. The browser's status bar at the bottom shows 'Fertig' and 'Internet'.

Screenshot 3: Entering personalized data

After the process of entering personalized data, the subscriber receives an opt_in message to his mailing client and can activate his inbox for retrieving business news from the news distribution service of Nedine.

A typical message, as it is distributed upon entering via the Web content management system and delivered via the news distribution service is shown in Screenshot 4.

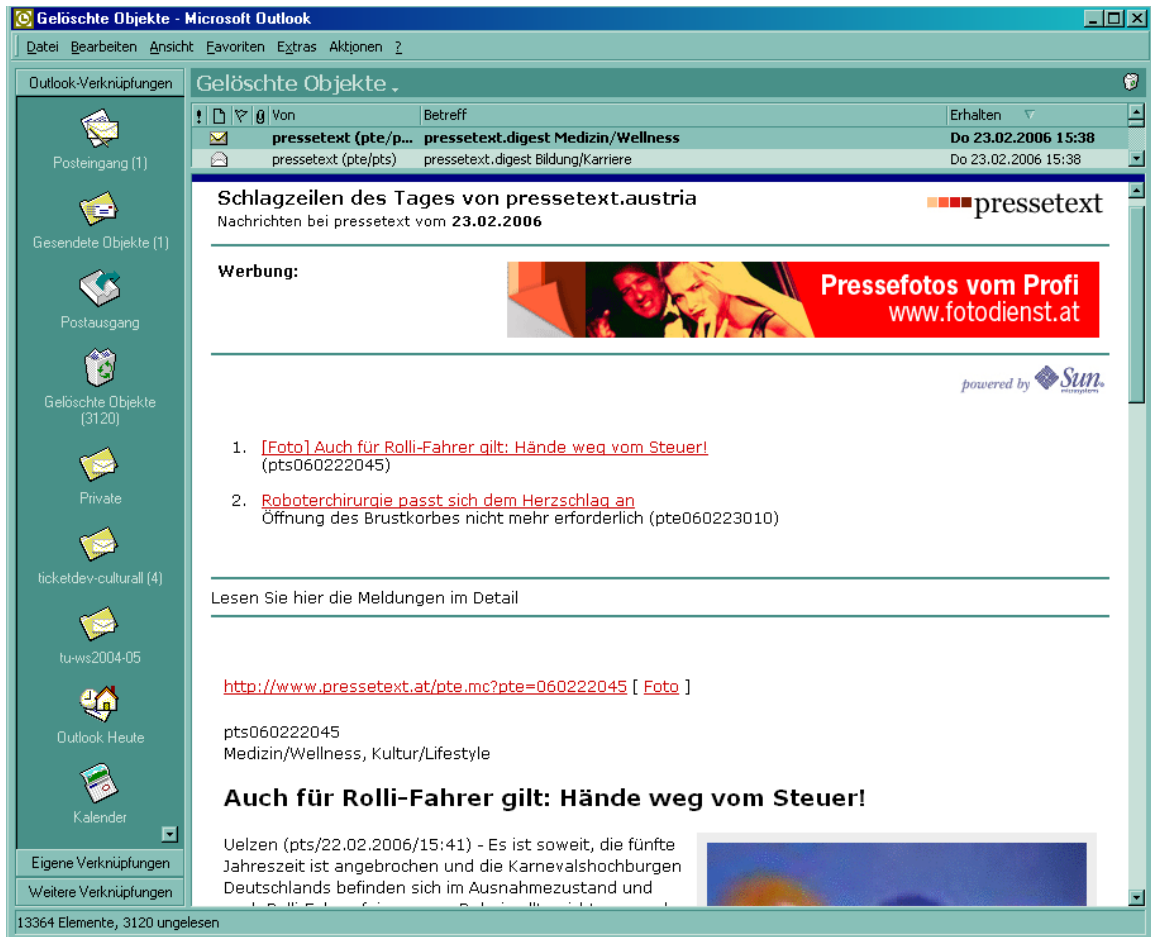


Screenshot 4: Single news item delivered to subscribers mail application

The mail message includes a business news item in the preferred language the user has selected and is associated to one of the categories the subscriber has entered as his /her preferred categories in the subscribing process (see screenshots 2 and 3).

A business news digest is created at presstext on a daily basis and as a result of optimisation discussions in the Nedine consortium is scheduled daily at 3pm in the afternoon. Each subscriber to the message digest (currently more than 105.000 subscribers thanks to Nedine initiatives) receives a daily digest as shown in Screenshot 5 below. Message digests are single-language messages and can be separated by topic categories or integrated email messages,

showing all personalized messages within the latest time period (= one day) for a subscriber.

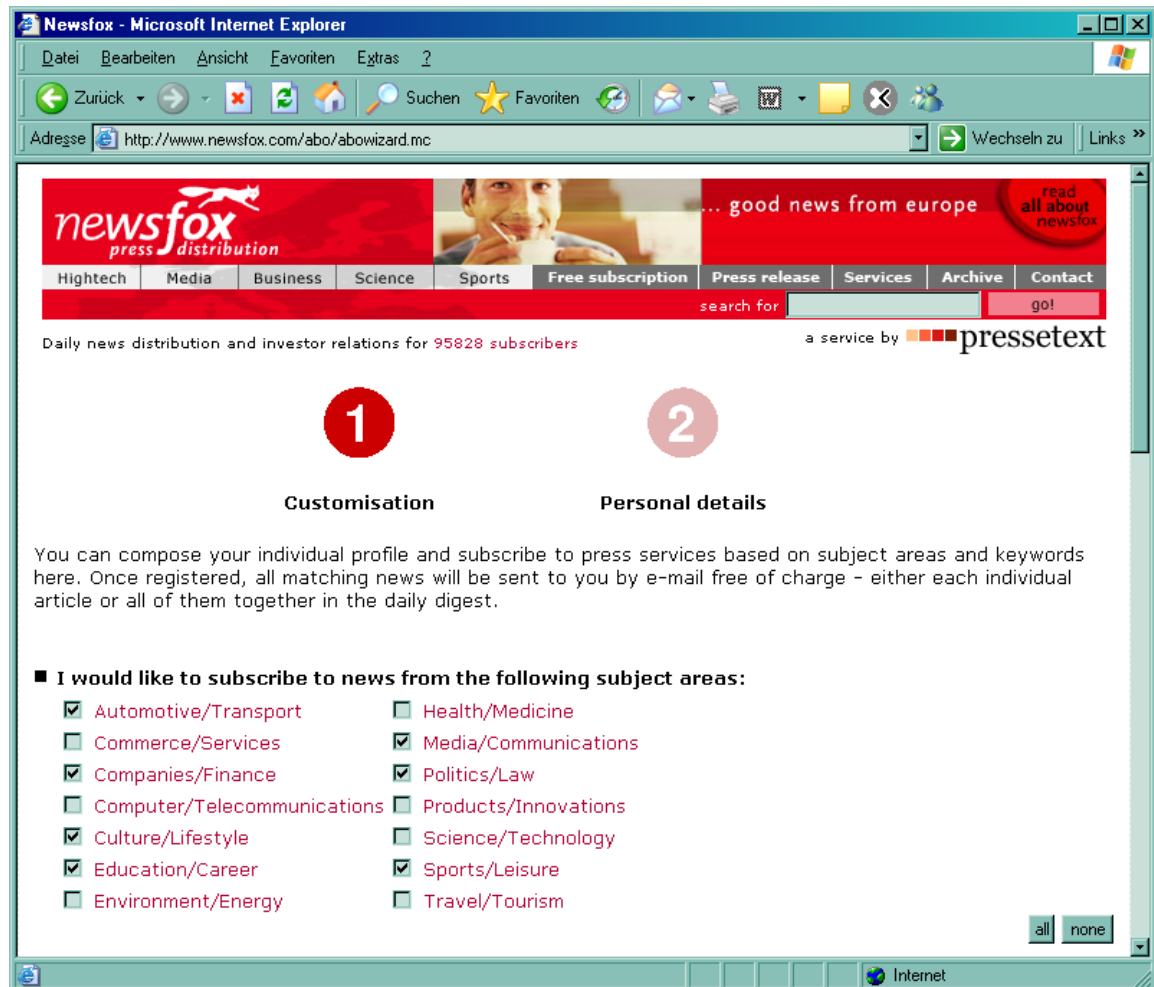


Screenshot 5: Message Digest for individual subscriber, category Health

Currently, with the progress of Nedines dissemination and marketing initiatives, presstext is distributing more than 400.000 email digests per day to more than 105.000 subscribers. This shows an increase of 47% compared to project start in 2004.

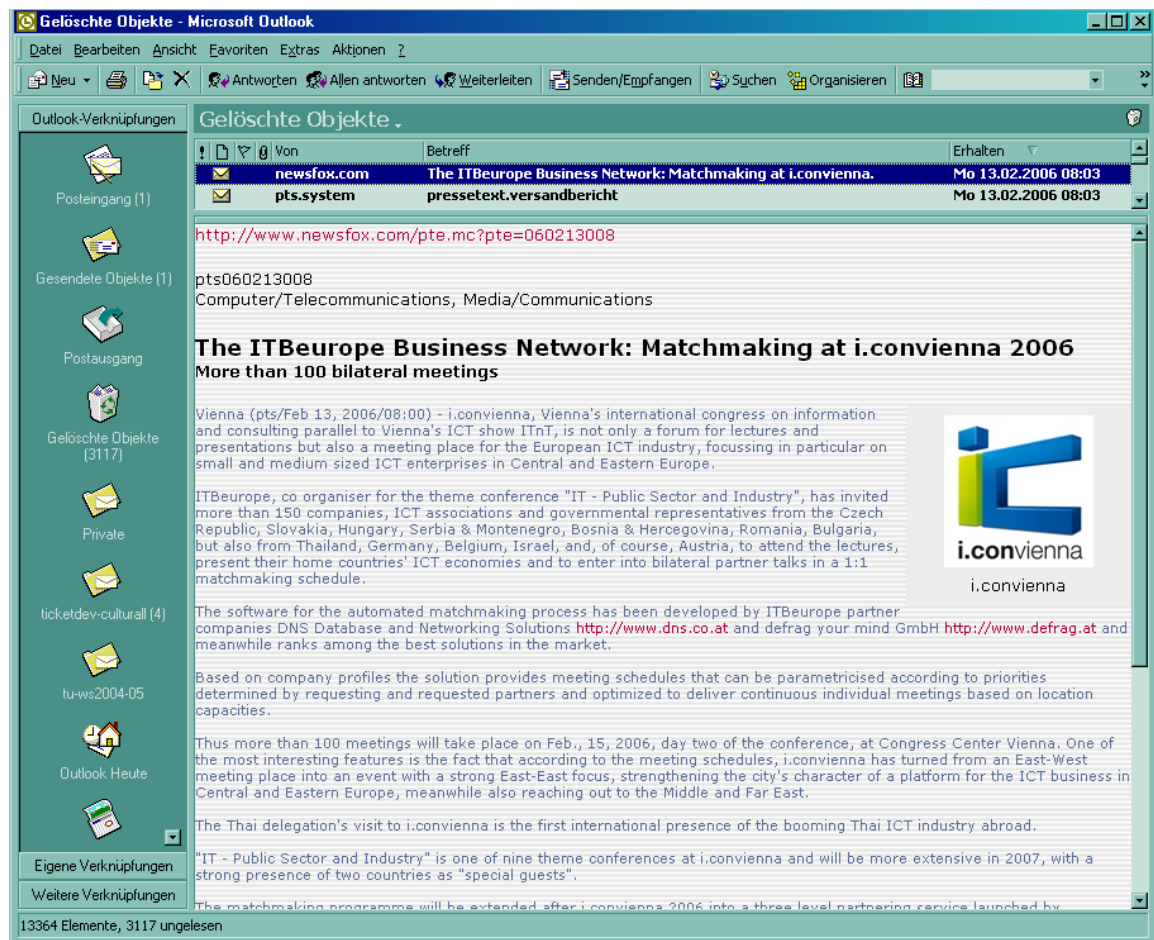
Additionally, presstext has opened an English version of business news delivery for international customers in the target region of Europe, addressed by presstext. At the web-based service www.newsfox.com english speaking subscribers can register to the business news delivery system and messages

are forwarded to their email accounts from the same software as used in the german service described above. Screenshot 6 documents the entering of personalized data for english speaking subscribers.



Screenshot 6: Entering personalized data for English speaking subscribers

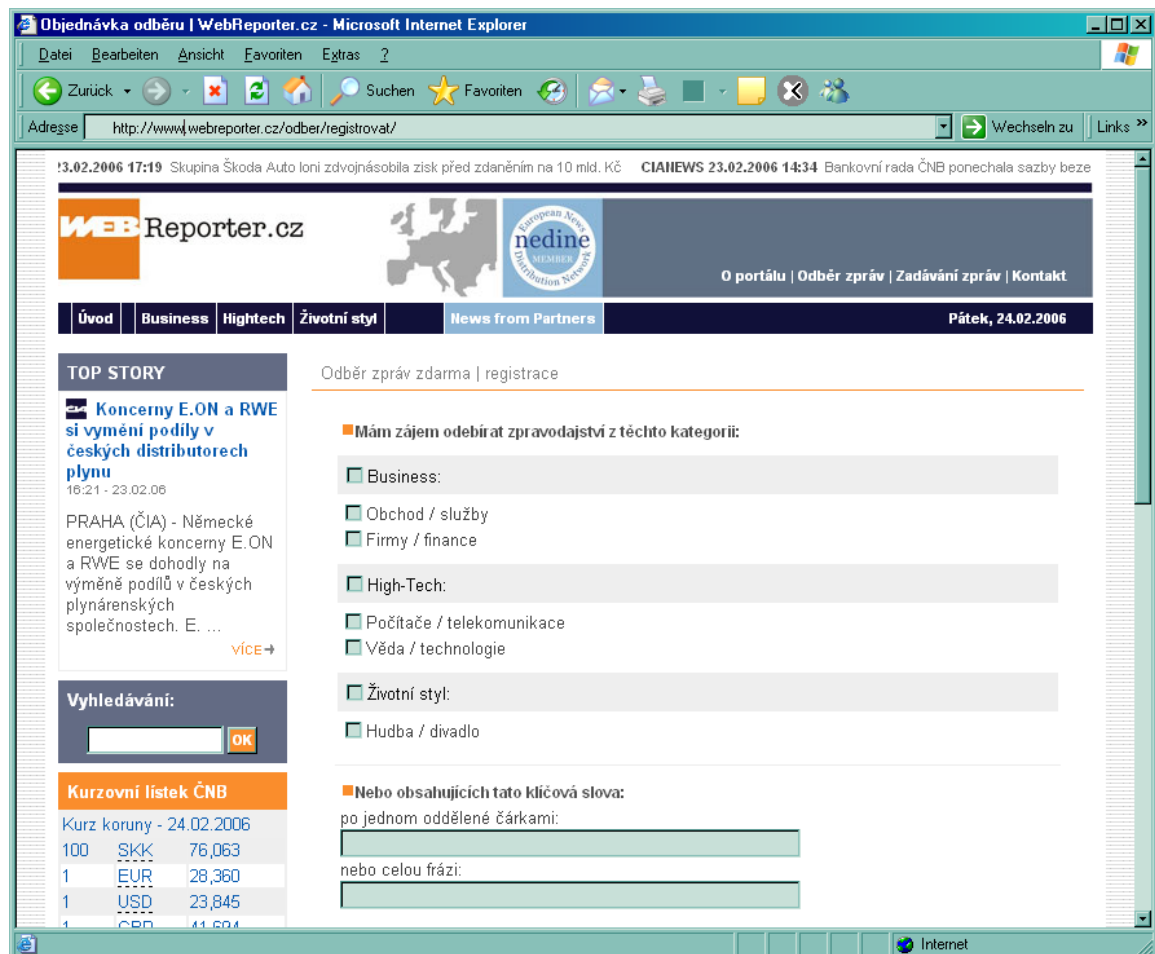
Consequently, the business news delivery system separated the multilingual contents managed in the presstext CMS service and created per-language business news digests. Screenshot 7 shows an English business news digest that is delivered by the software annotated in this document to an English speaking subscriber at presstext services.



Screenshot 7: English business news item sent to subscriber at presstext resp. newsfox service.

5.2.2 Business news delivery service at CIA

Since partner CIA is using the same algorithm as PTE and SITA, the registration and delivery service is technically almost identical to the service described at presstext. The user interface extends the Nedine registration service with its multi-lingual facet, since subscribers at CIA preferably subscribe to business news written in Czech language. Of course, the user guidance and web page content language is also in Czech as shown in Screenshot 8.



Screenshot 8: registration to mail distribution service at CIA

According to the national language and the internationally agreed content categories, CIA's subscriber service offers specific content areas for the subscribers to be selected as preferred categories.

Based on the individual subscription parameters and the addressing data shown in Screenshot 9, the personalized content distribution service based on the algorithm described in section 5.1 is used to create business news digests.

Objednávka odběru | WebReporter.cz - Microsoft Internet Explorer

Adresse <http://www.webreporter.cz/odber/registrovat/>

Každou zprávu ihned po zveřejnění

Formát zaslání zpráv:

Obsah ve formátu

HTML formát jako

Přihlašovací údaje:

*e-mail :

*Uživatel : (min 5 znaků)

*Heslo : (min 5 znaků)

pozn.: jméno i heslo musí být v rozsahu 5-15 znaků - povolené znaky jsou: a-z A-Z 0-9

Prosíme o laskavé vyplnění statistických údajů:

Odkud jste se o nás dozvěděli ?

V jakém oboru pracujete ?

*Souhlasím s podmínkami používání služeb webreporter.cz

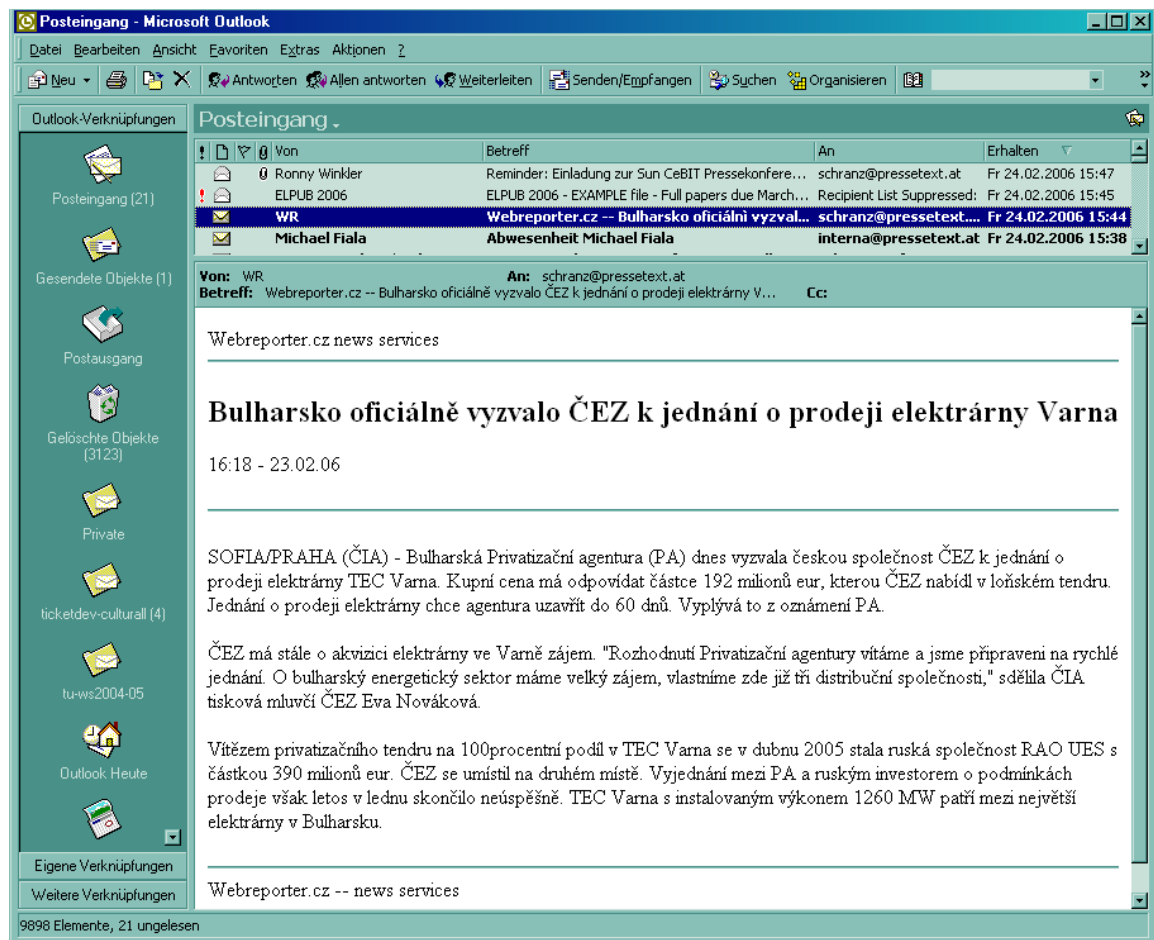
ne ano (nutno souhlasit)

Pro úspěšnou registraci je nutné vyplnit údaje označené *.

Screenshot 9: Entering address data and format preferences for the individual user profile of the subscriber

After entering the personalized data a subscriber is eligible to receive single messages or message digests from the personalized business news delivery service provided from CIA as partner of the Nedine network (the screenshots also provide examples of images of the Nedine logo on all partner sites).

Screenshot 10 shows as a result of the personalized business news delivery service a message digest created for a Czech subscriber according to his individual content category and keyword settings.



Screenshot 10: Message from business news delivery system for Czech subscriber by Nedine partner CIA

5.2.3 Business news delivery service at SITA

The pendant to the subscription service at Nedine partner SITA can be addressed at the Nedine peer site www.webnoviny.sk. Screenshot 11 documents the subscription process, where customers can enter personalized data on preferred categories and keywords as well as address information and additional data.

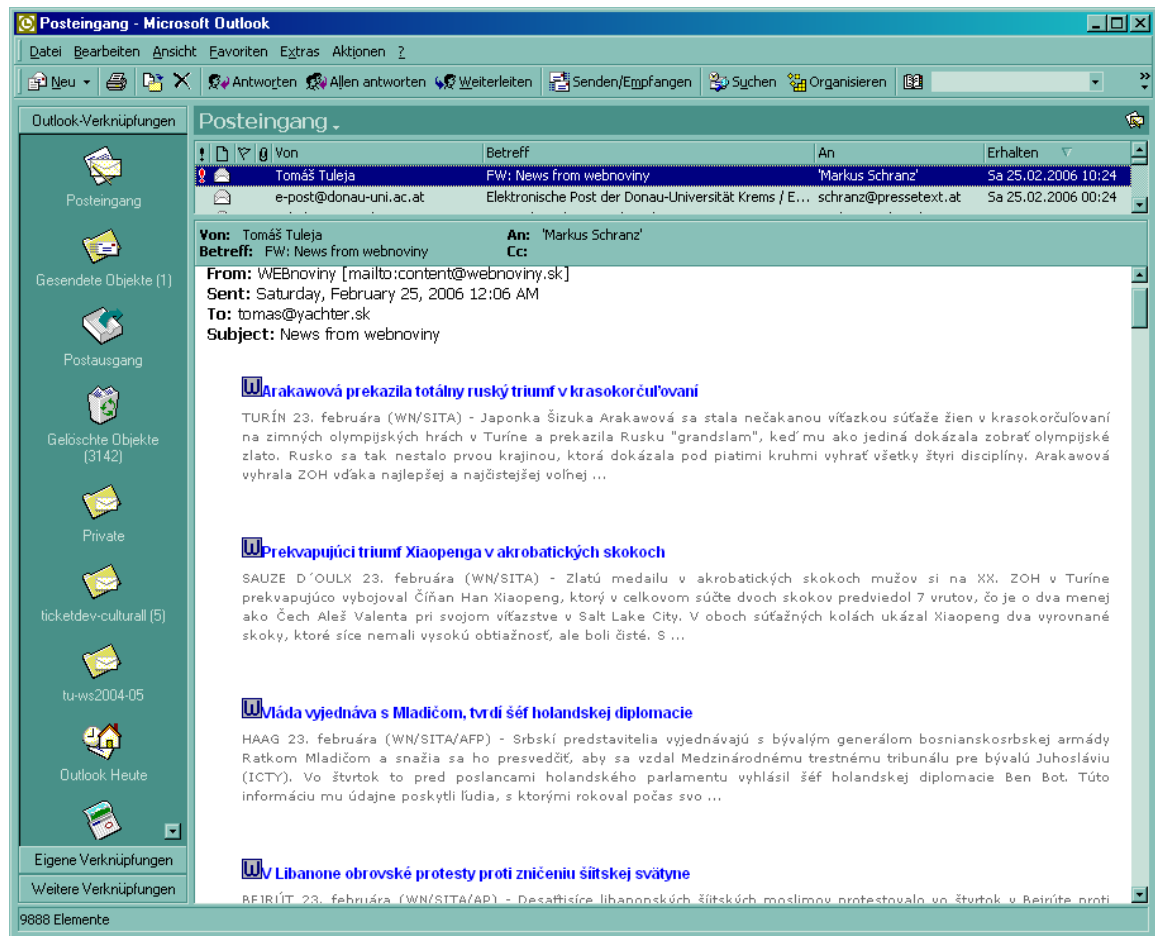
The screenshot shows a web browser window with the URL <http://subscribe.webnoviny.sk:8080/subscription.do>. The page title is 'Objednávka dodávky webnovín do e-mailovej schránky'. The form is divided into several sections:

- Povinné údaje:**
 - E-mail:
 - Frekvencia: po jednej správe (on-line) baľik o 16:30 baľik o 24:00
 - Formát: titulky správ anotácie správ plné texty s fotografiou
 - Výber kategórií správ: všetky kategórie vybrať kategórie
- Kategórie:**
 - Auto-moto
 - Financie a biznis
 - Hi-tech
 - Ľudia a život
 - Politika
 - Svet
 - Šoubiznis
 - Šport
- Podkategórie:**
 - auto MODELÝ
 - auto OBCHOD
 - auto VÝROBCOVIA
 - auto/šport FORMULA 1
 - auto/život INFO PRE VODIČOV
 - biznis OBCHOD
 - biznis PRIEMYSEL
 - biznis SLUŽBY
 - biznis VLÁDA, NRSR, ÚRADY
 - biznis/hitech TELEKOMUNIKÁCIE
 - financie BANKY
 - financie MAKROEKONOMIKA
 - financie POISTENIE A INVESTOVANIE
 - hitech HARDVÉR
 - hitech IT
- Nepovinné údaje:**
 - Meno a priezvisko:
 - Sektor:
 - Pozícia:
 - Región:

Screenshot 11: Entering personalized data for business news delivery services at SITA's webnoviny service, partner of Nedine network.

Based on the data presented in the registration form, the business news delivery system is implemented in similar ways at the service of SITA in Bratislava and each subscriber is consecutively receiving business news digests from the local service.

The international news distribution is performed via the Nedine network, since business items that are delivered between network partners become automatically local news that are delivered via the local news distribution service to the local registered customers. Independent of the origin of each business news, a Slovsak subscriber receives business news digests in the format as shown in Screenshit 12 as an example:



Screenshot 12: Example message digest received by a Slovak subscriber to SITA resp. Nedine services.

6 Summary

As documented in this annotation to deliverable D6.2, the Nedine consortium has implemented a demonstrator prototype of the technical business news delivery system.

Meeting the requirements of the goals presented in the technical Annex I to project NEDINE, the consortium has defined adequate mechanisms to distribute business news to journalists and opinion leaders, to optimize bulk information transfer for hundreds of thousands of addresses in a real-time environment, to provide personalized content filtering for a maximum of potential subscribers, and to identify future delivery channels to utilize modern technology throughout Europe.

Since the technical implementation is up and running in the services of the news agency partners, this document textually annotated some features, algorithms and user interfaces to describe to software deliverable D6.2 "Personalized Business News Distribution System". A full concept description and the documentation of the news distribution framework with actors, components and integration interfaces is part of deliverable D6.1 end of March 2006 which will refer to the software annotated in this document.

7 Attachment

Example implementation of Personalized Business News Distribution System at presstext

The following program code shows a perl5.0 implementation of the algorithm explained in section 5.2 at the partner PTE. Similar systems are running, scripted in different programming languages, at the partners CIA and SITA.

```
#!/usr/bin/perl
# Nedine news distribution system for daily business news delivery
# crontab me with one day intervals

use locale;

use dbserver;
use LWP::UserAgent;
use Data::Dumper;
use HTML::Entities;
use Getopt::Std;

use lib '/home/bin/lib';
use emailads;
use DBIx::Abstract;
use Mail::Mailer;
use Date::Manip;
use URI::Escape;
use MIME::QuotedPrint;
use Text::Wrap;

$Text::Wrap::columns = 72;      # for formatting text mails
$Text::Wrap::unexpand = undef; # do not turn spaces to tabs

select((select(STDOUT), $| = 1)[0]);

$program_name = $0;
$program_args = join(" ", @ARGV);

while (-f "/tmp/lock.db")
{
    $0 = "$program_name [waiting]";
    sleep(10);
}

getopts('hd:npl:b:e:r:t:g:f:u');
$0 = "$program_name [working $opt_r]";
$opt_p = 1; # set the transparent gif

my $statisticsRedirURLbase = "http://wap.ptc.at/";

if ($opt_h) {
    print "$program_name started with option -h:\nUsage: $program_name [-h] [-b <backlog>] [-e
<this@email.only>] [-r <alphabetical range>]\n";
    print " -h\t... print this info\n";
    print " -d\t... print debug infos (is an email-address, because debug mail is sent to <opt_d>\n";
    print " -n\t... do not send any mail, for debugging purposes\n";
    print " -b\t... e.g. 60*60*24 seconds until now (or until time set with -s)\n";
    print " -p\t... set a transparent image for html receivers\n";
    # print " -s\t... set time considered as \"now\" (as unixtime)\n";
    print " -e\t... send message to this email only\n";
    print " -r\t... range, e.g. 'a-k', 'l-', '-q' inklusive and case-independent range of initial
letters for email addresses to send to.\n";
    print " -t\t... set <t>oday to this date (e.g. 24.12.2002)\n";
    print " -l\t... send for language e.g. -l english \n";
    print " -f\t... present a file of email addresses to send the digest to \n";
}
```

```

print " -g\t... add <g>oldbach-media tags\n";
print "exiting without further action.\n";
exit;
}

my $dbs = new dbserver;
my $db = DBIx::Abstract->connect({
    driver => 'mysql',
    host => $dbs->dbhost(),
    dbname => 'pte3',
    user => 'root',
});

print "No mail will be sent at all.\n" if ($opt_n);

$evalstats = $opt_g;
$debug = $opt_d;

my $counter=0;          # total sent mails counter
my %mystats = ();      # key is pte, count is value

my $mailer;
if ($debug || $opt_e) {
    $mailer = new Mail::Mailer 'smtp', Server => 'sonja.pte.at';
} else {
    $mailer = new Mail::Mailer "sendmail";
}

$language = $opt_l || "deutsch";
my $langID = 1;
$langID = 2 if ($language eq "english");

my $digest = "digest";

# correct date format if supplied
if ($opt_t =~ m/^(\\d+)\\. (\\d+)\\. (\\d+)$/ ) {
    $opt_t = sprintf("%04d-%02d-%02d", $3, $2, $1);
}
$today = ParseDate($opt_t || 'today 0:00');
$today = DateCalc($today, "+14hours");
$ftoday = &UnixDate($today, "%e.%m.%Y");
format
$ftoday = &UnixDate($today, "%m/%e/%Y") if ($language eq "english");
$now = &UnixDate($today, "%Y-%m-%d %H:%M");
should be 14:00
# $now = time();
# $now = $opt_s if ($opt_s);
my $emad = new emailads($ftoday, $debug);

$totalbytes = 0; # total number of sent bytes via sendmail
my $logdir = "/home/log/" . &UnixDate("today", "%y%m%d");
mkdir $logdir unless (-d $logdir);

$presselog = "$logdir/presse-digest-".($opt_d ? 'debug-' : $opt_e ? 'single-' : '')."$language-
$.log";
$statlog = "$logdir/presse-digest-stats-".($opt_d ? 'debug-' : $opt_e ? 'single-' : '')."$language-
$.log";

$uglyheader = "+++ pressetext.austria +++ pressetext.deutschland +++ pressetext.schweiz";
$uglyheader = "
+++ newsfox.com +++ newsfox.com +++ newsfox.com +++" if ($language
eq "english");
$uglyfooter = "<font size=\\2\\ face=\\\"Verdana, Arial, Helvetica, sans-serif\\\"> &copy; " .
"<a href=\\\"http://www.newsfox.com\\\">newsfox.com +++ newsfox.com +++ newsfox.com +++ newsfox.com +++
newsfox.com +++</a>".
"<br>
<a href=\\\"http://www.newsfox.com/abo/\\\">Change my subscription</a></font><br>
" if ($language eq "english");

$backlog = 60*60*24; # 24 hours back from now
if ($opt_b) { $backlog = eval $opt_b; }
print "Backlog is $backlog.\n" if ($debug);

if ($opt_e) {
    $opt_r = substr($opt_e,0,1) . "-" . substr($opt_e,0,1);
}
print "email is $opt_e.\n" if ($debug);

$opt_r = "-" unless ($opt_r); # total range unless presented
($range_start,$range_stop) = split(/-/, $opt_r);
print "Range is [$range_start,$range_stop].\n" if ($debug);

```

```

$ad_global = 1;
$real_ad = '';

my $pteURL = "http://www.presetext.at";

open (LOG, ">".>$presselog") or die "Can't open log file: $presselog";
select ((select(LOG), $| = 1)[0]); # make it unbuffered

print LOG "Notify: Digesting started ($opt_r) at ".localtime()." (now=$now, backlog=$backlog)\n";

require 5.0;

# get channels and kuerzel from dbs (deutsch, english)
my $all_channels = $db->select_all_to_hashref({
    fields => 'id, kuerzel, bezeichnung',
    table => 'channels',
    where => { siteid => ($language eq "deutsch")?1:2 }
});
$all_channels->{'0'} = [ 'tn', 'Top-Story des Tages' ];
foreach my $myid (keys(%$all_channels)) {
    $mychannels{$all_channels->{$myid}->[0]} = $all_channels->{$myid}->[1];
    $channelids{$myid} = $all_channels->{$myid}->[0];
}
# print Dumper($all_channels) . "\n\n";
# print Dumper(\%mychannels) . "\n\n";
# print Dumper(\%channelids) . "\n\n";
# exit;

# now get all ressorts
my $all_ressorts = $db->select_all_to_hashref({
    fields => 'id, kuerzel, bezeichnung',
    table => 'ressorts',
    where => { siteid => ($language eq "deutsch")?1:2 }
});
foreach my $myid (keys(%$all_ressorts)) {
    $myressorts{$all_ressorts->{$myid}->[0]} = $all_ressorts->{$myid}->[1];
    $ressortids{$myid} = $all_ressorts->{$myid}->[0];
}

my $where_range = {};
if ($range_start) { $where_range->{'s.email'} = [ ">=", "$range_start"]; }
if ($range_stop) { $where_range->{'s.email'} = ["<=", "$range_stop"]; } # ý is chr(255), also
letztes Zeichen
print "where_range is ", Dumper($where_range), "\n" if ($debug);

# calculate the country specifications of all digest subscribers
$db->select({
    fields => 's.email, s.fkey' .
        (($language eq "english")?"_newsfox":"" ) .
        ' as fkey, s.pkey' .
        (($language eq "english")?"_newsfox":"" ) .
        ' as pkey, s.country, s.countryid, s.emailformat' .
        (($language eq "english")?"_newsfox":"" ) . ' as emailformat, s.emailtype' .
        (($language eq "english")?"_newsfox":"" ) . ' as emailtype ,s.complete' .
        (($language eq "english")?"_newsfox":"" ) . ' as complete, s.md5, s.id, s.journalist',
    table => 'subscribers s',
    where => [
        { 's.emailtype'.(($language eq "english")?"_newsfox":"" ) => ['like', '%D'],
          's.emailformat'.(($language eq "english")?"_newsfox":"" ) => ['!=', 'none'],
          's.onhold'.(($language eq "english")?"_newsfox":"" ) =>
        ['<', "\UNIX_TIMESTAMP(NOW())"]
        },
        "AND",
        "s.opt_in".(($language eq "english")?"_newsfox":"" ),
        "AND",
        1,
        #
        (scalar(keys(%$where_range))?$where_range:1)
    ],
});
print "Before reading all subscribers (" , $db->rows(), ").\n";
my @progressview = ("-", "\\", "|", "/");
while (my $abonment = $db->fetchrow hashref) {
    print $progressview[$cnt++ %4], "\r";
    $identified{$abonment->{id}} = $abonment->{email};
    $details{$abonment->{email}} =
    [
        $abonment->{emailformat}, # 0
        $abonment->{emailtype}, # 1
        $abonment->{complete}, # 2
        $abonment->{md5}, # 3
    ]
}

```

```

        $abonment->{country},          # 4
        $abonment->{id},              # 5
        $abonment->{countryid},      # 6
        $abonment->{journalist}     # 7
    ]
    unless ($details{$email});      # $abonment->{country} (element [4] is obsolete, read [6])
    foreach my $pkey (split(/s*/, $abonment->{pkey})) {
        next if (length($pkey) < 2); # ac 2004-11-16
        push(@{$emails{$abonment->{email}}}, "k:$pkey");
    }
    foreach my $fkey (split(/s*/, $abonment->{fkey})) {
        next if (length($fkey) < 2); # ac 2004-11-16
        push(@{$emails{$abonment->{email}}}, "fk:$fkey");
    }
}

print "Have read all subscribers.\n";
# check the keywords of temmel on this place next time...

$db->select({
    fields => 'sr.subscriberid, sr.ressortid',
    table => 'subscribers_ressorts sr, subscribers s',
    join => [ "sr.subscriberid = s.id" ],
    where => [
        { 's.emailtype'.(($language eq "english")?"_newsfox":"" ) => ['like', '%D'],
          's.emailformat'.(($language eq "english")?"_newsfox":"" ) => ['!=', 'none'],
          's.onhold'.(($language eq "english")?"_newsfox":"" ) => ['!=', 'none']
        },
        ['<', "\UNIX_TIMESTAMP(NOW())"],
        },
        "AND",
        "s.opt_in".(($language eq "english")?"_newsfox":"" ),
        "AND",
        (scalar(keys(%$where_range))?$where_range:1)
    ],
});
print "Have read all subscriber_ressorts (" , $db->rows(), ".\n";

$cnt=1;
while (my $tmp = $db->fetchrow_hashref) {
    print $progressview[$cnt++ %4], "\r";
    print "$cnt\n" unless ($cnt % 100000);
    if ($identified{$tmp->{subscriberid}}) {
        # print "Adding ressort $tmp->{ressortid} to list [" , join(", ", @{$ressorts{$tmp->{subscriberid}}}, "\n";
        push(@{$ressorts{$tmp->{subscriberid}}}, $tmp->{ressortid}) if ($identified{$tmp->{subscriberid}});
    }
}
print "Have built all ressorts (" , scalar(keys(%ressorts)), ".\n";

$db->select({
    fields => 'ss.subscriberid, ss.subregionid',
    table => 'subscribers_subregions ss, subscribers s',
    join => [ "ss.subscriberid = s.id" ],
    where => [
        { 's.emailtype'.(($language eq "english")?"_newsfox":"" ) => ['like', '%D'],
          's.emailformat'.(($language eq "english")?"_newsfox":"" ) => ['!=', 'none'],
          's.onhold'.(($language eq "english")?"_newsfox":"" ) => ['!=', 'none']
        },
        ['<', "\UNIX_TIMESTAMP(NOW())"],
        },
        "AND",
        "s.opt_in".(($language eq "english")?"_newsfox":"" ),
        "AND",
        (scalar(keys(%$where_range))?$where_range:1)
    ],
});
print "Have read all subscriber_subregions (" , $db->rows(), ".\n";

$cnt=1;
while (my $tmp = $db->fetchrow_hashref) {
    print $progressview[$cnt++ %4], "\r";
    print "$cnt\n" unless ($cnt % 100000);
    push(@{$countries{$identified{$tmp->{subscriberid}}}, $tmp->{subregionid}) if ($identified{$tmp->{subscriberid}});
}
print "Have built all subregions (" , scalar(keys(%countries)), ".\n";

$i=0;
$cnt=1;
print "Check now " , scalar(keys(%identified)), " keys of \identified.\n";
print "\%emails is already filled with " , scalar(keys(%emails)), " values.\n";

```

```

foreach my $identid (keys(%identified)) {
    print $progressview[$cnt++ %4], "\r";
    # print $i++, "Check for id $identid...\n";
    while (my $myressortid = pop(@{$ressorts{$identid}})) {
        # print "  Check for ressort $myressortid ($ressortids{$myressortid}).\n";
        push(@{$emails{$identified}{$identid}}, "r:$ressortids{$myressortid}");
    }
    if($ressortids{$myressortid});
    delete $ressorts{$identid};
    delete $identified{$identid};
}
print "Have defined all ressort2email assignments.\n";
print "emails{$opt_e} = ", Dumper($emails{$opt_e}), ".\n" if ($debug);
# exit;

print "Notify: Digesting intermediate ($opt_r) at ".localtime().": fetched ". scalar keys(%emails) . "
mailaddresses\n";
print LOG "Notify: Digesting intermediate ($opt_r) at ".localtime().": fetched ". scalar keys(%emails)
. " mailaddresses\n";

Sua = new LWP::UserAgent;
Sua->agent("Austrian Presstext/0.4");
Sua->timeout("15");

##
# Step 0: Get topnews
##

my $topnews = { };      # hashref - key: countryid, value: topnews
if ($language eq 'deutsch')
{
    foreach my $tmp (1,2,3)      # hardcoded!!! .at, .de, .ch
    {
        $db->select({
            fields => 't.*',
            table => 'topnews t, topnews_regions tr, ptas p',
            where => [
                'p.visible',
                'AND',
                {
                    'p.status' => 'sent',
                    'p.seq' => [ ">=", 0 ],
                    'p.languageid' => $langID,
                },
                'AND',
                {
                    'UNIX_TIMESTAMP(t.date)' => [ ">=", \"(UNIX_TIMESTAMP('$now') - $backlog)" ],
                    't.date' => [ "<", $now ]
                },
                'AND',
                { 'tr.regionid' => $tmp },
                'AND',
                { 'tr.channel' => 0 },
            ],
            join => [
                "p.id = t.ptaid",
                "t.id = tr.topnewsid",
            ],
            order => 'date DESC',
            extra => "limit 1",
        });
        if (my $tmp = $db->fetchrow_hashref)
        {
            $tmp->{dayofyyseq} = sprintf( "%06d%03d", $tmp->{dayofyy}, $tmp->{seq} );
            $topnews->{$tmp} = $tmp;
            print LOG "Notify: Found Topnews for region $tmp: $tmp->{dayofyyseq}: $tmp->{titel}\n";
            print "Notify: Found Topnews for region $tmp: $tmp->{dayofyyseq}: $tmp->{titel}\n" if
($debug);
        }
    }
}

##
# Step 1: Get all messages in the observed time frame...
##

$db->select({
    fields => 'p.id as pteid, p.dayofyy, p.seq, p.titel, p.utitel1, p.channel, p.ressort1, p.ressort2,
p.picture, ps.subregionid, pf.fotoid, s.pts',

```

```

    table => ' (ptas p, ptas_subregions ps, submitters s) LEFT JOIN ptas_fotos pf ON (pf.ptaid =
p.id)',
    where => [
        'p.visible',
        'AND',
        {
            'p.status' => 'sent',
            'p.languageid' => $langID,
        },
        'AND',
        # wenn mal der Hut brennt, einfach hier start          { 'p.subdate' => [ ">=", 1134656200 ] },
        { 'p.subdate' => [ ">=", \"(UNIX_TIMESTAMP(' $now') - $backlog)" ] },
        'AND',
        # bzw. ende-zeit eingeben          { 'p.subdate' => [ "<", 1134997200 ] }
        { 'p.subdate' => [ "<", \"UNIX_TIMESTAMP(' $now')" ] }
    ],
    join => [
        "p.id = ps.ptaid",
        "p.subid = s.id",
    ],
    # extra => "limit 500",
    });

my $skipped=0;
my $messages=0;
my @all_docs = ();
my %pte2pteid = ();
# ...and iterate over it
# while (($dayofyy, $titel, $utitel1, $utitel2, $channel, $ressort1, $ressort2, $seq, $picture, $AUT,
$GER, $SUI, $SUIf, $f_id, $pt_es) = Mysql::Statement::FetchRow $stm)

while (my $doc = $db->fetchrow_hashref)
{
    print LOG "Notify: Preparing message $doc->{pts}", sprintf("%06d%03d", $doc->{dayofyy}, $doc->{seq}),
    " (subregion $doc->{subregionid}, r1 $doc->{ressort1}, r2 $doc->{ressort2}) for mailing at ".
    localtime().".\n";
    print "Preparing message $doc->{pts}", sprintf("%06d%03d", $doc->{dayofyy}, $doc->{seq}), " (subregion
$doc->{subregionid}, r1 $doc->{ressort1}, r2 $doc->{ressort2}) for mailing at ".
    localtime().".\n" if ($debug);

    $doc->{dayofyy} = sprintf "%06d", $doc->{dayofyy};
    next if ($doc->{seq} < 0); # don't distribute frozen messages -- but seq 0 is OK!!
    $pte = $doc->{dayofyy}. sprintf ("%03d", $doc->{seq});
    $doc->{pte} = $pte;
    $pte2pteid{$pte} = $doc->{pteid};

    $SENDER{$pte} = $doc->{pts}?'pts':'pte'; # set to pts or pte
    print "$SENDER{$pte}$doc->{dayofyy} $doc->{seq} = $pte\n" if $debug;

    if ($doc->{fotoid} =~ /\(d+)/) {
        next if ($hires{$pte} && ($hires{$pte} != $1)); # ignore multiple images for same message, message
was already treated
        $hires{$pte} = $doc->{fotoid}; # set the hash for this pte
    }

    # fetch region name
    next if (grep(/$doc->{subregionid}/, @{$regions{$pte}}));
    push(@{$regions{$pte}}, $doc->{subregionid});
    next if (scalar(@{$regions{$pte}}) > 1); # if I have the other data already, stop here
    push(@all_docs, $doc);
    $messages++;

    $R1{$pte} = $doc->{ressort1};
    $R2{$pte} = $doc->{ressort2} if $doc->{ressort2};
    $Channel{$pte} = $doc->{channel};

    $mystats{$pte} = 0;

    $get_URL = "http://www.presetext.at/pte4mail.mc?pte=$pte";
    $get_URL = "http://www.newsfox.com/pteprint.mc?pte=$pte&print=1" if ($language eq "english"); #
changed by MS on dec 8, 2004: was www.presetext.com

    print LOG "Info: >>>$doc->{titel}\n>>>$doc->{utitel1}\n>>>$doc->{utitel2}\n";
    print LOG "Info: Digest fetching $SENDER{$pte}'$pte'.\n";

    my $sleepmin = 1;
    my $req = undef;
    my $res = undef;

    do {

```



```

print "$pte: $outland{$pte}\n" if $debug;
}

# now iterate over all docs and all email addresses to send digests
foreach my $doc (@all_docs)
{
    print "Creating ToDos on doc ", $doc->{'pte'}, ", one of ", scalar(@all_docs), " of today.\n" if
($opt_d);
    $doc->{'stop_addresses'} = &get_stoplist($doc->{'pteid'});

    foreach $email (keys(%emails))
    {
        next if ($opt_e && ($email ne $opt_e));
        next if ($doc->{'stop_addresses'}->{$email}); # Steht auf Stopliste - nächster bitte

        # prüfe keywordzugehörigkeit, die ist ja unabhängig
        # von der Landzugehörigkeit und Ressortzugehörigkeit
        foreach my $filter (@{$emails{$email}}) {
            print "Working on Mail $email, keyword $filter.\n" if ($debug > 1);
            $fkw = $kw = "";
            if ($filter =~ /^fk:(.+)$/) {
                $fkw = $1;
                my $fkwq = quotemeta($fkw);
                if ($HTML_MON{$doc->{pte}} =~ /\b$fkwq\b/i || $doc->{titel} =~ /\b$fkwq\b/i || $doc->{utitell} =~ /\b$fkwq\b/i) {
                    # print "Matched fkw $fkw in ", $doc->{pte}, "(", $doc->{dayofyy}, ":", $doc->{seq}, ")",
                    ($HTML_MON{$doc->{pte}} =~ /\b$fkwq\b/i)?"HTMLtext":"0", ($doc->{titel} =~ /\b$fkwq\b/i)?"doctype":"0",
                    ($doc->{utitell} =~ /\b$fkwq\b/i)?"subtitle":"0", "\n" if ($debug);
                    push @{$ToDo{$email}}, $doc->{pte} unless grep ($doc->{pte} == $_, @{$ToDo{$email}});
                }
                push @{$KWs{$email}{$doc->{pte}}}, $fkw;
            }
            if ($filter =~ /^k:(.+)$/) {
                $kw = $1;
                $kwq = quotemeta($kw);
                # print "Keyword ${markusN}++ is $kw\n";
                if ($HTML_MON{$doc->{pte}} =~ /$kwq/i || $doc->{titel} =~ /$kwq/i || $doc->{utitell} =~ /$kwq/i) {
                    push @{$ToDo{$email}}, $doc->{pte} unless grep ($doc->{pte} == $_, @{$ToDo{$email}});
                }
                push @{$KWs{$email}{$doc->{pte}}}, $kw;
            }
        } # end foreach filter

        # prüfe Landverträglichkeit (wofür ist Text, was will der Abonnent)
        my $isgood = 0;
        foreach $pteregion (@{$regions{$doc->{pte}}}) {
            # print "check region $pteregion of pte $doc->{pte} for email $email (private regions are [" ,
            join(", ", @{$countries{$email}}), "]).\n";
            $isgood ||= grep {$pteregion == $_} @{$countries{$email}};
        }
        # print "isgood is $isgood.\n";
        next unless ($isgood);

        # prüfe Filterverträglichkeit
        foreach my $filter (@{$emails{$email}}) {
            print "Working on Mail $email, filter $filter.\n" if ($debug > 1);

            if ($filter =~ /^r:$doc->{ressort1}$/) {
                push @{$ToDo{$email}}, $doc->{pte} unless grep ($doc->{pte} == $_, @{$ToDo{$email}});
                push @{$Ressorts{$email}}, $doc->{ressort1} unless grep (/^$doc->{ressort1}/, @{$Ressorts{$email}});
            }

            if (length($doc->{ressort2}) && ($filter =~ /^r:$doc->{ressort2}$/)) {
                push @{$ToDo{$email}}, $doc->{pte} unless grep ($doc->{pte} == $_, @{$ToDo{$email}});
                push @{$Ressorts{$email}}, $doc->{ressort2} unless grep (/^$doc->{ressort2}/, @{$Ressorts{$email}});
            }
        } # end foreach filter
    } # end foreach email, after this Ressorts{emails} contains all ressorts of the subscribers that
    actually contain content at that day
}
## exit;

print LOG "Notify: Digesting intermediate ($opt_r) at ".localtime().": fetched and assigned $messages
messages to all mailaddresses\n";

```

```

# Ready, filled %HTML, %TEXT, %R1, %R2, %ToDos,...
if ($messages && ($skipped/$messages > .1)) {
# ALARM, call me again later
# system("echo \"\${program_name} \" . join(" ",@ARGV) . "\" | at now + 5 minutes");
print LOG "process \"\${program_name} \" . join(" ",@ARGV) . "\" stopped for too many errors. Restart in
5 minutes\n";
exit;
}

print "ToDos:\t", join(",", @{$ToDos{$opt_e}}), "\n" if ($opt_e);
print "KWs:\t", Dumper($KWs{$opt_e}), "\n" if ($opt_e);
print LOG "Notify: Digest Step 1 (matching content and addresses) finished at ". localtime()."\n";
##
# Step 2
##
if ($opt_e) {
print "Preparing mail output only for $opt_e, since opt_e was present.\n";
myToDos{$opt_e} = $ToDos{$opt_e};
%ToDos = %myToDos;
}

if ($opt_f) {
open(FILE,$opt_f);
my @fileaddr = <FILE>;
close (FILE);
chomp(@fileaddr);
foreach my $addr (@fileaddr) {
myToDos{$addr} = $ToDos{$addr};
}
%ToDos = %myToDos;
print "Only run $0 for ", scalar(keys %ToDos), " addresses now.\n" if ($opt_d);
# exit;
}

my $matotal = scalar (keys %ToDos);
my $maindex=0;

my $land;
my $mailfrom;
my $mailreply;
my $boundary = "====" . time() . "====";
my $mymd5 = "";
my $mailheaders;
my $mediacom;
my %medialist = ();

# start loop for each mail address
foreach $e (sort keys %ToDos) {
$maindex++;
# print "preparing mail for $e..." if (grep(uc($e) eq uc($_),@testmails)); # MSMSMSM
next unless scalar(@{ $ToDos{$e}});
print $e, "Todos:\t", join(",", @{$ToDos{$e}}), "\n" if $debug;
print $e, "Ressorts:\t", join(",", @{$Ressorts{$e}}), "\n" if $debug;
print $e, "Details:\t", join(",", @{$details{$e}}), "\n" if $debug;

@{$Ressorts{$e}} = grep($_ =~ /\S+/,@{$Ressorts{$e}});
print $e, "Ressorts:\t", join(",", @{$Ressorts{$e}}), "\n" if $debug;

unless ($details{$e}) {
print LOG "Trouble: No details found for email $e at ".localtime()."\n";
next;
}

foreach my $p (@{ $ToDos{$e} }) {
print "p = $p..." if $debug;
# check each pr-news $p item in the list of ToDos for the email address $e
# add one media representative per media domain, if $e is a journalist
if ($details{$e}->[7])
{
my $mediadom = substr($e, index($e,'@') + 1);
$medialist{$pte2pteid{$p}}->{$mediadom}++;
$countjournalists{$pte2pteid{$p}}++;
}
}
print "\%medialist has ", scalar(keys %medialist), " entries.\n" if ($debug);
print LOG "\%medialist has ", scalar(keys %medialist), " entries.\n";

$mymd5 = $details{$e}->[3];
$pteURL = "http://www.presettext.com"; # english and default
$pteURL = "http://www.presettext.at" if ($details{$e}->[6] == 1);
$pteURL = "http://www.presettext.ch" if ($details{$e}->[6] == 3);

```

```

$pteURL = "http://www.presetext.de" if ($details{$e}->[6] == 2);
$pteURL = "http://www.presetext.at" if ($details{$e}->[6] > 3);
$pteURL = "http://www.presetext.at" if ($debug);
$pteURL = "http://www.newsfox.com" if ($language eq "english"); # english and default

$land = "europe";
if ($language eq "deutsch") {
    $land = "austria";
    $land = "austria" if ($details{$e}->[6] == 1);
    $land = "schweiz" if ($details{$e}->[6] == 3);
    $land = "deutschland" if ($details{$e}->[6] == 2);
}

$mailfrom = "\"presetext (pte/pts)\" <system@mail.presetext.com>";
$mailfrom = "\"newsfox.com\" <system@mail.presetext.com>" if ($language eq "english"); # newsfox
(presetext) am 29.11.2004 wegen WS geäert

$mailreply = "redaktion@presetext.at" if ($details{$e}->[6] == 1);
$mailreply = "abo@presetext.de" if ($details{$e}->[6] == 2);
$mailreply = "abo@presetext.ch" if ($details{$e}->[6] == 3);
$mailreply = "abo@presetext.com" if ($language eq "english");

### 1. RESSORTDIGESTS
if ($details{$e}->[1] eq "RD")
{
    # split monitor from ressort ptes
    @e_m = @e_r = ();
    foreach my $p (@{ $ToDos{$e} }) {
        print "kw list for $p and $e: ".join(", ", @{$KWws{$e}{$p}})."\\n" if $debug;
        if (@{ $KWws{$e}{$p} }) {push @e_m, $p;}
        else {push @e_r, $p;}
    }

    print "monitor: ".join(", ", @e_m)."\\n" if $debug;
    print "ressort: ".join(", ", @e_r)."\\n" if $debug;

    # OK get rid of monitor first (i.e. mail the monitor digests first)!
    if (@e_m)
    {
        $counter++;

        $mailto = &Digest($e, [ @e_m ], [()], 0, ($details{$e}->[0] =~ /html/) ? "text/html" :
"text/plain"). "\\n\\n";
        $totalbytes += length($mailto);

        # my $mailtosubject = "=?ISO-8859-1?Q?" . encode_qp("presetext.monitor") . "=?";
        # $mailtosubject =~ s/\\?Q\\?(.*)\\?(.*)\\?=/\\?Q\\?\\$1=3F$2\\?=/g;

        my $mailtosubject = "presetext.monitor";
        if ($mailtosubject =~ m/[\\x80-\\xFF]/) # Nur wenn notwendig!!!
        {
            $mailtosubject = encode_qp($mailtosubject);
            $mailtosubject =~ s/\\?/=3F/g;
            $mailtosubject =~ s/[\\r\\n\\t\\s]+//g;
            $mailtosubject = "=?ISO-8859-1?Q?$mailtosubject?=";
        }

        my $contenttype = ($details{$e}->[0] =~ /html/)?"text/html":"text/plain";
        $contenttype .= ' '; charset="ISO-8859-1";
        $mailheaders = { # 'To' => $e,
            'To' => ($opt_d)"\"$e\" <$opt_d>:$e,
            'From' => $mailfrom,
            'Precedence' => 'bulk',
            'Subject' => $mailtosubject,
            'Reply-to' => $mailreply,
            'Organisation' => "presetext.$land - $pteURL",
            'MIME-Version' => '1.0',
            'Message-Id' => "<SUBSID." . $details{$e}->[5] . ".$counter@mail.presetext.com>",
            'Content-Type' => $contenttype,
            'Content-type' => "multipart/alternative; boundary=\\\"$boundary\\\"",
            'Content-Transfer-Encoding' => 'quoted-printable',
        };
        $boundary = '--' . $boundary;
        $mailto = $boundary. "\\n".
            "Content-Transfer-Encoding: quoted-printable\\n".
            "Content-Type: text/plain; charset=\\\"ISO-8859-1\\\"\\n\\n".
            &Digest($e, [ @e_m ], [()], "text/plain").
            "\\n\\n".
            $boundary. "\\n".
    }
}

```

```

#           "Content-Transfer-Encoding: quoted-printable\n".
#           "Content-Type: text/html; charset=\"ISO-8859-1\"\n\n".
#                                     &Digest($e, [ @e_m ], [ () ], ($details{$e}->[0] =~
/table/)?"table/html":"text/html").
#           "\n\n".
#
#           $boundary. "--\n";

unless ($opt_n) {
    print "Sending mail to $e\n"    if ($debug);
    $mailer->open($mailheaders);
    print $mailer $mailtext;
    $mailer->close();
}
} # if (@e_m)

# now deal with remaining ressort triggered
@e = @e_r;
foreach $r ( @{ $Ressorts{$e} } )
{
    %ptesInThisRessort = (); # undef @kate;
    print "Working on ressort $r for $e\n";
    @e2 = @e; # wird hier immer um eine pte kürzer, wenn diese bereits in einem ressort digest
              # drinnen war (siehe nachste if-clause und das auszuführende statement dazu
    foreach $pte (@e2) {
        #print "    Consider Adding $pte ($R1{$pte},$R2{$pte})";
        if (($R1{$pte} eq $r) || (($R2{$pte} eq $r) && !(grep(/$R1{$pte}/,@{$Ressorts{$e}}))) {
            $ptesInThisRessort{$pte} = 1;
        }
    }
}
next unless scalar(keys (%ptesInThisRessort));

print "Ressort Digest for $r : ".join (",", keys(%ptesInThisRessort))."\n" if ($debug);
$add{"$real_ad\:$r"}++;
$ad_global = 1;
$counter++;

# my $mailsubject = "=?ISO-8859-1?Q?" . encode_qp("presstext.$_digest $myressorts{$r}") . "=?";
# $mailsubject =~ s/\?Q?(.*)\?(.*)\?=/\?Q\?$1=3F$2\?=/g;

my $mailsubject = "presstext.$_digest $myressorts{$r}";
if ($mailsubject =~ m/[x80-xFF]/) # Nur wenn notwendig!!!
{
    $mailsubject = encode_qp($mailsubject);
    $mailsubject =~ s/\?/=3F/g;
    $mailsubject =~ s/=[\r\n\t\s]+//g;
    $mailsubject = "=?ISO-8859-1?Q?$mailsubject?=";
}

my $contenttype = ($details{$e}->[0] =~ /html/)?"text/html":"text/plain";
$contenttype .= '; charset="ISO-8859-1"';
$mailheaders = {
#
#           'To'           => $e,
#           'To' => ($opt_d)"\"$e\" <$opt_d>:$e,
#           'From'        => $mailfrom,
#           'Precedence'  => 'bulk',
#           'Subject'     => $mailsubject,
#           'Reply-to'    => $mailreply,
#           'Organisation' => "presstext.$land - $pteURL",
#           'MIME-Version' => '1.0',
#           'Message-Id'  => "<$mymd5.$$-$counter@mail.presstext.com>",
#           'Content-Type' => $contenttype,
#           'Content-type' => "multipart/alternative; boundary=\"\$boundary\"",
#           'Content-Transfer-Encoding' => 'quoted-printable',
#
#           };
# $mailtext = encode_qp($mailheaders->{'Content-Type'}) . "\n";
# $boundary = '--' . $boundary;
# $mailtext = $boundary. "\n".
#           "Content-Transfer-Encoding: quoted-printable\n".
#           "Content-Type: text/plain; charset=\"ISO-8859-1\"\n\n".
#           &Digest($e, [ () ], [ keys(%ptesInThisRessort) ], "text/plain").
#           "\n\n".
#
#           $boundary. "\n".
#           "Content-Transfer-Encoding: quoted-printable\n".
#           "Content-Type: text/html; charset=\"ISO-8859-1\"\n\n".
#           &Digest($e, [ () ], [ keys(%ptesInThisRessort) ], ($details{$e}->[0] =~
/table/)?"table/html":"text/html").
#           "\n\n".
#
#           $boundary. "--\n";

```

```

    $mailtext = &Digest($e, [()], [ keys(%ptesInThisRessort) ], 0, ($details{$e}->[0] =~
/html/)?"text/html":"text/plain"). "\n\n";

    $totalbytes += length($mailtext);

    unless ($opt_n) {
        $mailer->open($mailheaders);
        print $mailer $mailtext;
        $mailer->close();
    }
} # end RD
### 2. GESAMTDIGESTS
elsif ($details{$e}->[1] eq "GD")
{
    $ad_gd = 1; # set if ads go for GDs
    @e_m = @e_r = ();
    foreach my $p (@{ $ToDos{$e} }) {
        print "kw list for $p and $e: ".join(" ", @{$KWs{$e}{$p}})." \n" if $debug;
        if (@{ $KWs{$e}{$p}}) {push @e_m, $p;}
        else {push @e_r, $p;}
    }

    print "monitor: ".join(" ", @e_m)." \n" if $debug;
    print "ressort: ".join(" ", @e_r)." \n" if $debug;

#   @e_r = sort { $R1{$a} cmp $R1{$b} ? $R1{$a} cmp $R1{$b} : $a <=> $b} @e_r;
    $ad_global = $ad_gd;
    $add{$real_ad}++;
    $ad_global = 1;
    $counter++;

    my $tn = $stopnews->{ $details{$e}->[6] };
    # $tn->{dayofyyseq} = '060110025'; # test case
    $mailtext = &Digest($e, [ @e_m ], [ @e_r ], $tn, ($details{$e}->[0] =~
/html/)?"text/html":"text/plain"). "\n\n";
    $totalbytes += length($mailtext);

    # my $mailsubject = "=?ISO-8859-1?Q? " .
    #   encode_qp(
    #     (($language ne "english") ? "presstext" : "newsfox") .
    #     ".$_digest".(@e_m ? "/monitor": "") .
    #     ($tn ? ': ' : ' ( $TITEL{$tn->{dayofyyseq}} || $tn->{titel} ) : '' )
    #     ) . "?=";
    # $mailsubject =~ s/\?Q?(.*)\?(.*)\?=/\?Q?$1=3F$2\?=/g;

    my $mailsubject =
        (($language ne "english") ? "presstext" : "newsfox") .
        ".$_digest".(@e_m ? "/monitor": "") .
        ($tn ? ': ' : ' ( $TITEL{$tn->{dayofyyseq}} || $tn->{titel} ) : '');

    if ($mailsubject =~ m/[\x80-\xFF]/) # Nur wenn notwendig!!!
    {
        $mailsubject = substr( $mailsubject, 0, 72 ) . '...'
        if (length($mailsubject) >= 75);
        $mailsubject = encode_qp( $mailsubject );
        $mailsubject =~ s/\?/=3F/g;
        $mailsubject =~ s/=[\r\n\t\s]+//g;
        $mailsubject = "=?ISO-8859-1?Q?$mailsubject?=";
    }

    my $contenttype = ($details{$e}->[0] =~ /html/)?"text/html":"text/plain";
    $contenttype .= '; charset="ISO-8859-1"';
    $mailheaders = { # 'To' => $e,
                    'To' => ($opt_d)? "\"$e\" <$opt_d>":$e,
                    'From' => $mailfrom,
                    'Precedence' => 'bulk',
                    'Subject' => $mailsubject,
                    'Reply-to' => $mailreply,
                    'Organisation' => "presstext.$land - $pteURL",
                    'MIME-Version' => '1.0',
                    'Message-Id' => "<$myid5.$$- $counter@mail.presstext.com>",
                    'Content-Type' => $contenttype,
                    'Content-type' => "multipart/alternative; boundary=\"\$boundary\"",
                    'Content-Transfer-Encoding' => 'quoted-printable',
                    };
#   $boundary = '--' . $boundary;
#   $mailtext = $boundary. "\n".
#       "Content-Transfer-Encoding: quoted-printable\n".
#       "Content-Type: text/plain; charset=\"ISO-8859-1\" \n\n".
#       &Digest($e, [ @e_m ], [ @e_r ], "text/plain").

```

```

#           "\n\n".
#
#           $boundary. "\n".
#           "Content-Transfer-Encoding: quoted-printable\n".
#           "Content-Type: text/html; charset=\"ISO-8859-1\"\n\n".
#           &Digest($e, [ @e_m ], [ @e_r ], ($details{$e}->[0] =~
/table/)?"table/html":"text/html").
#           "\n\n".
#
#           $boundary. "--\n";

unless ($opt_n) {
    print "mailheader: ".Dumper($mailheaders)." \n\n";
    $mailer->open($mailheaders);
    print $mailer $mailtext;
    $mailer->close();
}
else
{
    print LOG "Warning: Very unknown digest '$details{$e}->[1]' for $e.\n";
    next;
}
print LOG "Info: Rendered $e \[".join(", ", @{ $ToDos{$e} } )."\] mail nr: $counter (address
$mainindex/$matotal) at ".localtime(). ".\n";
}

##
# Step 3: Statistics
##

$db->ensure_connection();

open (SLOG, ">>$statlog");
select ((select(SLOG, $| = 1)[0]); # make it unbuffered
print SLOG "Notify: Evaluation of statistics started at ".localtime()."\n";
print SLOG "program: $program_name -- range: $opt_r -- language: $language\n";

## count, how often a single pat was sent in this digest
my $sum = 0;
foreach $key (sort keys %mystats)
{
    $sum += $mystats{$key};
    print SLOG "pta: $key (ID $pte2pteid{$key}) count $mystats{$key}\n";

    $db->insert('temp.stats', { pta => $key, 'count' => $mystats{$key}, 'when1' => \NOW(), 'prog' =>
"$program_name range=$opt_r lang=$language" });
    $db->query('INSERT INTO pteV3stats.ptas (id,sent) VALUES (?,?) '.
'ON DUPLICATE KEY UPDATE sent=sent+'.($mystats{$key} + 0), $pte2pteid{$key},
($mystats{$key} + 0));
}

print SLOG "total: $sum texts, $counter digests, $totalbytes bytes\n";

## count the number of total pressetexts sent in this digest
$db->insert('temp.stats', { 'count' => $sum, 'prog' => "$program_name range=$opt_r lang=$language",
'when1' => \NOW(), 'text' => 'total pressetexts'});

## count the number of digests sent here
$db->insert('temp.stats', { 'realcount' => $counter, 'prog' => "$program_name range=$opt_r
lang=$language", 'when1' => \NOW(), 'text' => "total mails for $ftoday"});

## count the number of total bytes sent here
$db->insert('temp.stats', { 'realcount' => $totalbytes, 'prog' => "$program_name range=$opt_r
lang=$language", 'when1' => \NOW(), 'text' => 'total bytes in mails sent'});

## count all media mxnames for all ptas of today
print Dumper(%medialist) if ($debug);
print "before database to \%medialist.\n" if ($debug);

if (scalar(keys %medialist))
{
    $db->query("LOCK TABLES pteV3stats.tmp_ptas_media WRITE, pteV3stats.ptas WRITE");      ### LOCK DB
    !!!

    my $mlist_set = ('.join(',',keys %medialist).');
    $db->select({
        'fields' => 'ptaid,medias',
        'tables' => 'pteV3stats.tmp_ptas_media',
        'where' => { 'ptaid' => [ 'IN', \ $mlist_set ] }
    });
}

```

```

while (my $tmp = $db->fetchrow_hashref) {
    foreach my $elem (split('|', $tmp->{medias})) {
        $elem =~ /^(\d+)\@(.\+)$/;
        $medialistDB{$tmp->{ptaid}}->{$2} = $1;
    }
}

foreach my $docid (keys(%medialist))
{
    foreach my $med (keys(%{$medialist{$docid}})) {
        $medialistDB{$docid}->{$med} += $medialist{$docid}->{$med};          # just count up in the DB...
    }
    my $mlist = join( '|', map { $medialistDB{$docid}->{$_}."@$_" } keys %{$medialistDB{$docid}} );
    my $medias = (scalar(keys %{$medialistDB{$docid}}) + 0);

    print "Going to replace docid $docid with string ", substr($mlist,0,100), " (length = ",
length($mlist), ").\n";
    $db->replace({
        table => 'pteV3stats.tmp_ptas_media',
        fields => { 'ptaid' => $docid, 'medias' => $mlist }
    });

    # add $countjournalists and $countmedia to the DB for each docid
    print SLOG "pta ID $docid counted $countjournalists{$docid} journalists and $medias medias.\n";
    $db->query('INSERT INTO pteV3stats.ptas (id,sent_journalists,sent_media) VALUES (?,?,?) '.
        'ON DUPLICATE KEY UPDATE sent_journalists=sent_journalists+?,sent_media=?',
        $docid, ($countjournalists{$docid}+0), $medias,
        ($countjournalists{$docid}+0), $medias);
}

$db->query("UNLOCK TABLES");          ### UNLOCK DB

print "after database to \%medialist.\n" if ($debug);
print "Digesting stopped.\n" if ($debug);

print LOG "Notify: Digesting stopped ($opt_r) at ".localtime()."\n";
print LOG "Notify: Digesting ads count is ";
foreach $key (keys(%add)) {
    print LOG "$key: $add{$key}, ";
}

print LOG " at ".localtime()."\n";
close LOG;
close SLOG;

exit;

#####
## create Digest as subroutine
#####

sub Digest
{
    my ($towhom, $ptas_m, $ptas, $stopstory, $mimetype) = @_;

    my ($mimetypeOld, $digest, $complete, $mymd5) = @{$details{$towhom}}[0..3];
    my ($tbody, $textcontent);
    my (@ptas_m) = @$ptas_m;
    my (@ptas) = @$ptas;
    my ($ad_aut, $ad_ger, $ad_sui);

    $ad_aut = $ad_global && ($details{$towhom}->[6] == 1)?1:0;
    $ad_ger = $ad_global && ($details{$towhom}->[6] == 2)?1:0;
    $ad_sui = $ad_global && ($details{$towhom}->[6] == 3)?1:0;

    $real_ad = "";
    foreach $ptam (@$ptas_m) { $mystats{$ptam}++; }
    foreach $pta (@$ptas) { $mystats{$pta}++; }

    print "digest: monitor: ".join(" ", @ptas_m)."\n" if $debug;
    print "digest: ressort: ".join(" ", @ptas)."\n" if $debug;
    print "ad_aut, ad_ger, ad_sui: $ad_aut, $ad_ger, $ad_sui\n" if $debug;

    my @channels = ();
    push @channels, 0 if ($stopstory || ($digest ne "GD"));

    if ($digest eq 'GD')
    {

```

```

    foreach my $pta (@ptas_m,@ptas) {
        if (!$stopstory or ($pta ne $stopstory->{dayofyyseq})) {           # Topstory kommt im eigenen 0-er
Channel
            push(@channels, $Channel{$pta})
                unless (grep ($_ eq $Channel{$pta},@channels));
        }
    }
    @channels = sort { $a <=> $b } @channels; # changed at dec 8, 2004 MS

    # now create body for html or plain
    if ($mimetype =~ /html/)
    {
        my $css = "
<style type='text/css'>
body {
    padding: 5px;
}
body,p,a,td,th,li,div {
    font-family: Verdana, Arial, Helvetica;
    font-size: 9pt;
    color:black;
}
a:link { color: #cc0000; text-decoration: underline; }
a:visited { color: #333333; text-decoration: underline; }
a:hover { color: white; background-color: #cc0000; text-decoration: underline; }
a:active { color: white; background-color: #cc0000; text-decoration: underline; }
h1 { font-size: 11pt; font-weight: bold; margin-bottom: 3px; }
h2 { font-size: 11pt; font-weight: bold; }
h3 { font-size: 10pt; font-weight: bold; }
.subh1 { font-size: 8pt; }
.footer { font-size: 8pt; }
.footer a { font-size: 8pt; }
.impressum { font-size: 9pt; font-style: italic; }
.txt { font-size: 9pt; line-height:140%; }
.txtklein { font-size: 7.5pt; }
</style>
";

        print "\n-----> now creating body for HTML style mails\n\n" if ($debug); #ac
        my $statTag = "";
        $statTag = "
<script language='JavaScript' src='http://www.nex-ware.net/cgi-
bin/netnugget.cgi?siteid=goldbhpTEgoldbhdIGEST$ftoday&zf=$mymd5&img=1'>
</script>
<noscript><img src='http://www.nex-ware.net/cgi-
bin/netnugget.cgi?siteid=goldbhpTEgoldbhdIGEST$ftoday&zf=$mymd5&img=2' border='0'>
</noscript>
" if ($evalstats);

        # set general $mbody element
        if ($language eq "deutsch")
        {
            print "\n-----> now creating german content\n\n" if ($debug); #ac
            $mbody = "
<html>
<head>
<title>pressetext.digest</title>
<meta http-equiv='Content-Type' content='text/html; charset=ISO-8859-1'>
$css
</head>
<body bgcolor='#ffffff' text='#000000' link='#cc0000' alink='#000000' vlink='#333333'>
<table width='100%' border='0' cellspacing='5' cellpadding='0'>
<tr>
<td>
<h1>Schlagzeilen des Tages von pressetext.$l and</h1>
<span class='subh1'>Nachrichten bei pressetext vom <b>$ftoday</b></span></td>
<td align='right'>
<a href='$pteURL/' target='_new'><img src='$pteURL/images/pte_logo.gif' border=0
ALT='pressetext'></a>
</td>
</tr>
</table>";

            my $emailad = $emad->tell($ad_aut?'at':$ad_ger?'de':'ch','html',$rotate_ads);
            if (($ad_aut || $ad_ger || $ad_sui) && length($emailad)) {
                $mbody .= "
<hr noshade>
<table width='100%' border='0' cellspacing='5' cellpadding='0'>
<tr>
<td width='80' valign='top'><b>Werbung:</b></td>

```

```

    <td>
";
    $mbody .= $emailad;
    $mbody .= "
</font>
</td>
</tr>
</table>";
}

    $mbody .= "\n<hr noshade>\n";
    $mbody .= " <img src=\"http://www.presetext.at/partner/sun/poweredBySunLogo_116x25.gif\"
align=\"right\"><br clear=\"all\">" if ($ad_aut);
}

    $mbody .= "\n<hr noshade>\n";
    $mbody .= " <img src=\"http://www.presetext.at/partner/sun/poweredBySunLogo_116x25.gif\"
align=\"right\"><br clear=\"all\">" if ($ad_aut);
}
    elsif ($language eq "english") {
        print "\n-----> now creating english text/html content\n\n" if ($debug); #ac
        $mbody = "
<html>
<head>
<title>newsfox.digest</title>
<meta http-equiv=\"Content-Type\" content=\"text/html; charset=ISO-8859-1\">
$css
</head>
<body bgcolor=\"#ffffff\" text=\"#000000\" link=\"#cc0000\" alink=\"#000000\" vlink=\"#333333\">
<table width=\"100%\" border=\"0\" cellspacing=\"5\" cellpadding=\"0\">
<tr>
<td><font face=\"Verdana, Arial, Helvetica, sans-serif\"><b>News of the day from newsfox<br>
<b><font size=\"-1\">Business news collected for your profile at
</font></td>
<td>
<div align=\"right\"><font face=\"Verdana, Arial, Helvetica, sans-serif\" size=\"-1\"><a
href=\"$pteURL/\" target=\"_new\"><img src=\"$pteURL/pics/newsfoxlogo.gif\" border=0 width=\"74\"
height=\"32\" ALT=\"presetext\"></a></font></div>
</td>
</tr>
</table>";

        my $emailad = $mad->tell('newsfox','html',$rotate_ads);
        if (length($emailad))
        {
            $mbody .= "
<hr noshade>
<table width=\"100%\" border=\"0\" cellspacing=\"5\" cellpadding=\"0\">
<tr>
<td width=\"80\" valign=\"top\"><font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\"><b>Advertisement:</b></font></td>
<td><font face=\"Verdana, Arial, Helvetica, sans-serif\" size=\"2\">
$emailad
</font></td>
</tr>
</table>";
        }

        $mbody .= "\n<hr noshade>\n";
        $mbody .= " <img src=\"http://www.presetext.at/partner/sun/poweredBySunLogo_116x25.gif\"
align=\"right\"><br clear=\"all\">" if ($ad_aut);
}
    $mbody .= "<A NAME=\"ueber\">&nbsp;</A>";

    ### NUR TITEL
    if ($complete == 0)
    {
        print "in nurtitel\n" if $debug;

        # Top-Story:
        if ($topstory && ($digest eq "GD"))
        {
            $mbody .= "<h3>$all_channels->{0}->[1]</h3>";
            my $p = $topstory->{dayofyyseq};
            if (grep { $p eq $_ } @ptas_m) {
                $mbody .= "\n<a href=\"$pteURL/pte.mc?pte=$p\">$TITEL{$p}</a><br>$UTITEL1{$p}
($SENDER{$p}$p, $SENDER{$p}.monitor <b>'".join(" ", @{$KWs{$towhom}{$p}})."'</b><br>&nbsp;";
            } else {
                $mbody .= "\n<a href=\"$pteURL/pte.mc?pte=$p\">$TITEL{$p}</a><br>$UTITEL1{$p}
($SENDER{$p}$p)<br>&nbsp;";
            }
        }
    }

```

```

    }
  }

foreach my $channel (@channels)
{
  next if (($channel == 0) && ($digest eq "GD"));

  if ($digest eq "GD") {
    $mbody .= "<h3>$all_channels->{$channel}->[1]</h3>";
  }

  $mbody .= "<OL> <!-- nur_titel is Y -->";

  foreach my $p (sort @ptas_m) {
    next if (($digest eq "GD") && ( ($channel{$p} != $channel) || ($stopstory->{dayofyyseq} eq
    $p) ));
    $mbody .= "\n<LI><a href=\"$pteURL/pte.mc?pte=$p\">$TITEL{$p}</a><br>$UTITEL1{$p}
    ($SENDER{$p}$p, $SENDER{$p}.monitor <b>".join (" ", @{$KWs{$towhom}{$p}})."'/<b><br>&nbsp;</LI>";
  }
  foreach my $p (sort @ptas) {
    next if (($digest eq "GD") && ( ($channel{$p} != $channel) || ($stopstory->{dayofyyseq} eq
    $p) ));
    $mbody .= "\n<LI><a href=\"$pteURL/pte.mc?pte=$p\">$TITEL{$p}</a><br>$UTITEL1{$p}
    ($SENDER{$p}$p)<br>&nbsp;</LI>";
  }
  $mbody .= "\n</OL>\n";
}
}
### KOMPLETTE ARTIKEL
else
{
  print "in not nurtitel\n" if $debug;
  # weiter bei nicht nur titel, sondern volle Nachricht...

  my $mbody2 = "";
  my $backtext = ($language eq "deutsch") ? "Zur Meldungs&uuml;bersicht" :
    "Back to news outline";

  # Top-Story:
  if ($stopstory && ($digest eq "GD"))
  {
    $mbody .= "<h3>$all_channels->{0}->[1]</h3>";
    my $p = $stopstory->{dayofyyseq};
    if (grep { $p eq $_ } @ptas_m) {
      $mbody .= "\n<a href=\"$p\">$TITEL{$p}</a><br>$UTITEL1{$p} ($SENDER{$p}$p,
      $SENDER{$p}.monitor <b>".join (" ", @{$KWs{$towhom}{$p}})."'/<b><br>&nbsp;\n";
    } else {
      $mbody .= "\n<a href=\"$p\">$TITEL{$p}</a><br>$UTITEL1{$p} ($SENDER{$p}$p)<br>&nbsp;\n";
    }
    $mbody2 .= "<br><a name=\"$p\">&nbsp;</a>$HTML{$p}<P><A HREF=\"#ueber\">$backtext</A></P><hr
    noshade>";
  }

  foreach my $channel (@channels) {
    next if (($channel == 0) && ($digest eq "GD"));
    if ($digest eq "GD") {
      $mbody .= "<h3>$all_channels->{$channel}->[1]</h3>";
    }

    $mbody .= "<OL> <!-- nur_titel is N -->";

    foreach my $p (sort @ptas_m) {
      # next if (($digest eq "GD") && ($channel{$p} != $channel));
      next if (($digest eq "GD") && ( ($channel{$p} != $channel) || ($stopstory->{dayofyyseq} eq
      $p) ));
      $mbody .= "\n<LI><a href=\"$p\">$TITEL{$p}</a><br>$UTITEL1{$p} ($SENDER{$p}$p,
      $SENDER{$p}.monitor <b>".join (" ", @{$KWs{$towhom}{$p}})."'/<b><br>&nbsp;</LI>";
      $mbody2 .= "<br><a name=\"$p\">&nbsp;</a>$HTML{$p}<P><A HREF=\"#ueber\">$backtext</A></P><hr
      noshade>";
    }
    foreach my $p (sort @ptas) {
      # next if (($digest eq "GD") && ($channel{$p} != $channel));
      next if (($digest eq "GD") && ( ($channel{$p} != $channel) || ($stopstory->{dayofyyseq} eq
      $p) ));
      $mbody .= "\n<LI><a href=\"$p\">$TITEL{$p}</a><br>$UTITEL1{$p}
      ($SENDER{$p}$p)<br>&nbsp;</LI>";
      $mbody2 .= "<br><a name=\"$p\">&nbsp;</a>$HTML{$p}<P><A HREF=\"#ueber\">$backtext</A></P><hr
      noshade>";
    }
    $mbody .= "\n</OL>\n";
  }
}

```

```

    $mbody .= "<hr noshade>Lesen Sie hier die Meldungen im Detail<hr noshade>" if ($language eq
"deutsch");
    $mbody .= "<hr noshade>Read details on the business messages below<hr noshade>" if ($language eq
"english");
    $mbody .= $mbody2;
}

    if ($language eq "deutsch") {
#     deu, aus, scjhw, termin, news, aender
    $mbody .=
"
<p class=\"footer\">
&copy; " .
"<a href=\"http://www.presetext.de\">presetext.deutschland</a> +++ ".
"<a href=\"http://www.presetext.at\">presetext.austria</a> +++ " .
"<a href=\"http://www.presetext.ch\">presetext.schweiz</a> +++ " .
"<a href=\"http://www.termindienst.at\">termindienst</a>" .
" +++ <a href=\"http://www.newsfox.com\">newsfox.com</a>" .
" und der jeweilige Aussender<br>
<a href=\"$pteURL/abo/\">&Auml;nderung Abo</a> f&uuml;r $towhom
</p>
<p class=\"impressum\">
<i><!-- necessary for mac/eudora -->
Medieninhaber und Herausgeber:<br>
presetext Nachrichtenagentur GmbH, Josefstädter Straße 44, A-1080 Wien
<br><br>
presetext ist eine Nachrichtenagentur für Meinungsbildner in den
Bereichen Hightech, Medien, Business und Leben. Die inhaltliche
Verantwortung für redaktionelle Meldungen (pte) liegt bei presetext,
für über presetext verbreitete Presseaussendungen (pts) beim jeweiligen
Aussender. Die Nachrichten werden auf den presetext-Länderplattformen
von http://www.presetext.com publiziert sowie den Abonnement-Wünschen und
der gewählten Zustellart entsprechend einzeln oder täglich als Newsletter
(presetext.digest) an die Abonnenten verschickt.
Weitere Informationen erhalten Sie bei unserem Redaktionsservice
unter Tel. +43-1-81140-300.
</i>
</p>
<hr noshade>;
    } elsif ($language eq "english") {
    $mbody .= "$uglyfooter
Editor and publisher: presetext Nachrichtenagentur GmbH<br>
Josefstädter Straße 44, 1080 Vienna, Austria
<hr noshade>;
    }

    if ($opt_p) {
    $mbody .= "<img src=\"${statisticsRedirURLbase}pt.mc?date=" . &UnixDate("today", "%d.%m.%Y") .
"&type=digest&md5=$mymd5&ptes=";
    foreach $p (@ptas_m, @ptas) { $mbody .= $pte2pteid{$p} . ","; }
    chop($mbody);
    $mbody .= "\" alt=\" \">;
    }

    $mbody =~ s#http://www.presetext.at# $pteURL#g;
    $mbody =~ s#(href\s*\s*\s*) (\\"|\')?(http://\[^\s\"'\>+)(\\"|\')?#q(target="_new"
href="http://www.nex-ware.net/cgi-bin/netnugget.cgi?siteid=goldbhPTEgoldbhDIGEST) . $ftoday . q(LINK)
. &linkvalue($3,$j++) . q(&zf=) . $mymd5 . q(&url=) . uri_escape($3) . q(&img=3) . $4#ige if
($evalstats);

    $mbody .= "\n</body>;
    $mbody .= "<body>$statTag</body>" if ($evalstats);
    $mbody .= "</html>\n";

    print "\n-----> language=$language and mimetype=$mimetype\n\n" if ($debug); #ac
    return (encode_qp(encode_entities($mbody, "\200-\377")));
} # if ($mimetype =~ /html/)
else # text/plain...
{
    print "\n-----> now creating body for text style mails\n\n" if ($debug); #ac
    if ($language eq "deutsch") {
    $mbody = "
++ presetext.austria +++ presetext.deutschland +++ presetext.schweiz ++
-----
";
    $mbody .= "
Schlagzeilen des Tages von presetext.$land
Nachrichten bei presetext vom $ftoday
-----
";

```

```

my $emailad = $mad->tell($ad_aut?'at':$ad_ger?'de':'ch','text',$rotate_ads);
if (length($emailad) && ($ad_aut || $ad_ger || $ad_sui)) { $mbody .= $emailad; }
} elsif ($language eq "english") {
$mbody = "
+++ newsfox.com      +++ newsfox.com      +++ newsfox.com      +++
-----
";
$mbody .= "
News of the day
Message digest by newsfox from $ftoday
-----
";
}

$i = 0;

$mbody .= "\n-----\n";
if ($stopstory && ($digest eq "GD"))
{
    $mbody .= $all_channels->{0}->[1]." +++\n-----\n";
    my $p = $stopstory->{dayofyyseq};

    $i++;
    $nr = sprintf("%2d. ", $i);
    $utitel = $UTITEL1{$p};
    $utitel .= "\n" if ($utitel);
    if (grep { $p eq $_ } @ptas_m) { # is a monitor message
        $mbody .= "\n" . wrap($nr, " ", "$TITEL{$p}\n$utitel($SENDER{$p}$p, $SENDER{$p}.monitor
"".join(" ", @{$KWs{$towhom}{$p}})."")\n");
    } else {
        $mbody .= "\n" . wrap($nr, " ", "$TITEL{$p}\n$utitel($SENDER{$p}$p)\n");
    }
    $mbody .= "\n-----\n";
}

foreach my $channel (@channels)
{
    next if (($channel == 0) && ($digest eq "GD"));
    if ($digest eq "GD") {
        $mbody .= $all_channels->{$channel}->[1]." +++\n-----\n";
    }

    foreach my $p (sort @ptas_m) {
        next if (($digest eq "GD") && ( ($channel{$p} != $channel) || ($stopstory && ($stopstory->{dayofyyseq} eq $p) ) ));
        $i++;
        $nr = sprintf("%2d. ", $i);
        $utitel = $UTITEL1{$p};
        $utitel .= "\n" if ($utitel);
        $mbody .= "\n" . wrap($nr, " ", "$TITEL{$p}\n$utitel($SENDER{$p}$p, $SENDER{$p}.monitor
"".join(" ", @{$KWs{$towhom}{$p}})."")\n");
    }
    foreach my $p (sort @ptas) {
        next if (($digest eq "GD") && ( ($channel{$p} != $channel) || ($stopstory && ($stopstory->{dayofyyseq} eq $p) ) ));
        $i++;
        $nr = sprintf("%2d. ", $i);
        $utitel = $UTITEL1{$p};
        $utitel .= "\n" if ($utitel);
        $mbody .= "\n" . wrap($nr, " ", "$TITEL{$p}\n$utitel($SENDER{$p}$p)\n");
    }
    $mbody .= "\n-----\n";
} # end foreach titelanzeige

$mbody .= "
Lesen Sie hier die Meldungen im Detail:
-----
" if ($language eq "deutsch");
$mbody .= "
Read details on the business messages below:
-----
" if ($language eq "english");

$i = 0;
$casi = 0;
if ($stopstory && ($digest eq "GD"))
{

```

```

$tbody .= $all_channels->{0}->[1]." +++\n-----
-----\n";
my $p = $stopstory->{dayofyyseq};

$i++;
$nr = sprintf("%2d. ", $i);
$tbody .= "\n" . sprintf("%2d. ", $i) . "$TEXT{$p}\n";
$tbody .= "\n-----\n";
}
foreach my $channel (@channels) {
next if (($channel == 0) && ($digest eq "GD"));
$csci = $i;
if ($digest eq "GD") { # nur bei Gesamtdigests; Ressortdigests sind ohnehin auf Mails
aufgeteilt
$tbody .= $all_channels->{$channel}->[1]." +++\n-----
-----\n";
}

foreach my $p (sort @ptas_m) {
next if (($digest eq "GD") && ( ($channel{$p} != $channel) || ($stopstory->{dayofyyseq} eq $p)
));
if ($i++ > $csci) {
$tbody .= " $uglyheader\n\n";
}
$tbody .= "\n" . sprintf("%2d. ", $i) . "$TEXT{$p}\n";
}
foreach my $p (sort @ptas) {
next if (($digest eq "GD") && ( ($channel{$p} != $channel) || ($stopstory->{dayofyyseq} eq $p)
));
if ($i++ > $csci) {
$tbody .= " $uglyheader\n\n";
}
$tbody .= "\n" . sprintf("%2d. ", $i) . "$TEXT{$p}\n";
}
$tbody .= "\n-----\n";

} # end foreach channel (Channel-Unterteilung der GesamtDigests)

if ($language eq "deutsch") {
$tbody .= "Änderung Abo für $towhom: $pteURL/abo
-----
(c) presstext Nachrichtenagentur GmbH
presstext.austria: http://www.presstext.at
presstext.deutschland: http://www.presstext.de
presstext.schweiz: http://www.presstext.ch
newsfox: http://www.newsfox.com
-----
Medieninhaber und Herausgeber:
presstext Nachrichtenagentur GmbH, Josefstädter Straße 44, A-1080 Wien

presstext ist eine Nachrichtenagentur für Meinungsbildner in den
Bereichen Hightech, Medien, Business und Leben. Die inhaltliche
Verantwortung für redaktionelle Meldungen (pte) liegt bei presstext,
für über presstext verbreitete Presseausendungen (pts) beim jeweiligen
Aussender. Die Nachrichten werden auf den presstext-Länderplattformen
von http://www.presstext.com publiziert sowie den Abonnement-Wünschen
und der gewählten Zustellart entsprechend einzeln oder täglich als
Newsletter (presstext.digest) an die Abonnenten verschickt. Weitere
Informationen erhalten Sie bei unserem Redaktionsservice
unter Tel. +43-1-81140-300.
-----
++ presstext.austria +++ presstext.deutschland +++ presstext.schweiz ++
";
} elsif ($language eq "english") {
# $tbody .= "Update Your personal subscription profile ($towhom):
$tbody .= "
-----
Editor and publisher: presstext Nachrichtenagentur GmbH
Josefstädter Straße 44, 1080 Vienna, Austria
-----
newsfox.com +++ newsfox.com +++ newsfox.com +++ newsfox.com +++ newsfox.com
";
}

$tbody =~ s#http://www.presstext.com#$pteURL#g;
# $tbody = MIME::QuotedPrint::encode($tbody);
return (encode_qp($tbody));
}
}

```

```
# end sub Digest

sub linkvalue {
    my $link = shift;
    my $counter = shift;

    # print "Read Link: $link, counter = $counter.\n";
    if ($link =~ /pte=\d{6}\d{3}/) {
        # print "detected pte string with id $1.\n";
        return "p$1";
    }
    return $counter;
}

sub get_stoplist {
    my $pteid = shift;

    # Stoplisten auswerten und returnen
    $db->select({
        fields => 'al.addresses as adr',
        table => 'address_lists al, ptas_stop_al ps',
        join => 'ps.alid = al.id',
        where => [
            { 'ps.ptaid' => $pteid },
            'AND',
            'stop_digest'
        ]
    });

    my %tmpstop = ();
    while (my $alist = $db->fetchrow_hashref) {
        foreach my $a (split( /\s*[,;\s]\s*/, $alist->{adr} )) {
            $tmpstop{$a} = 1;
        }
    }
    return \%tmpstop;
}
```