EXHIBIT Q

EXHIBIT
PENGAD 800-631-6989
Piccolo-7
ID
7/9/09 hF

EZ Reader User's Guide and Reference Manual

CHASE MANHATTAN BANK, N.A.
CUSTOMER KNOWLEDGE TEAM

CONFIDENTIAL

Table of Contents

Table of Figures

Table of Tables

Preface

Introduction and Release Notes

This document describes EZ Reader, currently in use by the ChaseDirect unit of Chase Manhattan Bank.

It describes EZ Reader version 1.0 requirements, design, operation, maintenance, and administration.

Document Organization

This document presents information about EZ Reader in four Parts: EZ Reader Overview, EZ Reader Design, EZ Reader Usage, and EZ Reader Maintenance.

Part One - EZ Reader Overview (p. 8) focuses on the busi iess purpose, scope and objectives of the system, EZ Reader's overall position in the systems architecture as well as high-level information about application operation and visible process flow.

Part Two - EZ Reader Design and Environment (p. 15) describes the EZ Reader design, architecture, and implementation, with sections on requirements, process flows, and displays.

Part Three - EZ Reader Usage (p. 21) provides necessary information on running and terminating EZ Reader in its two available modes. This section also provides information about monitoring an EZ Reader session.

Part Four - EZ Reader Maintenance (p. 26) describes in detail the steps necessary to maintance EZ Reader code, including the rulebase, casebase, and actionbase. Suggestions on techniques for more efficient maintenance are also included.

Appendices

Appendix A - Error/Status Messages describes application-related run time errors and API-related status messages. **Appendix B - EZ Reader Glossary** gives a list of words/phrases with their meanings pertinent to EZ Reader. **Appendix C - Future Releases of EZ Reader** lists suggested enhancements. **Appendix D - Standards and Miscellaneous** describes miscellaneous issues, including a list of references and people to contact for problem resolution.

Part One - EZ Reader Overview

Part One - EZ Reader Overview consists of two sections:

Executive Overview

A description of the business environment of EZ Reader, including the system's purpose and role in the business process.

EZ Reader Business Requirements Overview

A high-level description of the major business requirements addressed by EZ Reader.

Executive Overview

Business Purpose of EZ Reader

ChaseDirect, the remote banking arm of Chase, receives hundreds of electronic mail (email) messages per week from prospects and customers. The messages request information or services such as check reorders or copies of processed checks. Electronic mail from customers is received and processed by the staff of ChaseDirect Technical Support. The number of incoming electronic letters exceeds 50 per day and ChaseDirect expects that this volume will grow significantly this year and beyond.

Processing emails is costly because it involves a significant amount of manual intervention. As of July 1995, ChaseDirect Technical Support spent 10 work days per month to process incoming emails. Since emails can currently only be read and responded to by people, a growth in email volume suggests a corresponding increase in email processing support staff.

Business Benefits

EZ Reader will provide several significant benefits to ChaseDirect. It will:

- enable Chase to handle increasing email volumes without a proportional increase in staff.
- reduce future staff needs for processing emails, including people to interpret, route and answer emails.
- provide faster email responses to customers.

EZ Reader will be designed to work within Chase's strategic plan for intra-company electronic collaboration[1].

[1] the T.120 document conferencing protocol

EZ Reader Business Requirements

This section outlines specific requirements for functionality, ease of use, and security for EZ Reader.

Overall Business Requirements

EZ Reader's primary goal is to reduce the number of Internet emails that ChaseDirect Customer Service Representatives must process by acting as an intermediary intepreter, router, and automatic response generator for incoming emails.

Functionality Requirements

EZ Reader contains a knowledgebase of interpretation rules and a casebase of actual emails linked to appropriate responses. The application also contains functions that are necessary for interfacing with, i.e., retrieving from and replying to, Lotus Notes mail software.

Operation of EZ Reader

EZ Reader has two available processing modes: continuous and manual. For production purposes, continuous processing is required for uninterrupted email retrieval (i.e., 24 hours a day, 7 days a week), interpretation, and response. Manual processing is intended for development, testing, and maintenance purposes.

Continuous Processing

In general, EZ Reader will operate continuously in conjunction with Lotus Notes mail functions in ChaseDirect. Whenever a piece of electronic mail is received into the Lotus Notes environment in ChaseDirect, EZ Reader will automatically retrieve the email, process (i.e., interpret) it, and either automatically respond to it by placing it in a Lotus Notes "outbox" or forward it to the ChaseDirect "inbox" for human review and response.

EZ Reader will reduce, but not eliminate, manual processing of email messages by ChaseDirect. As the application expands, an increasing percentage of email will be processed without human intervention.

Initially, manual review of messages to which automatic responses are assigned by EZ Reader is needed to validate the

quality of system processing. For this reason, ChaseDirect will still need to take manual steps to read and upload those messages that are assigned automatic responses by EZ Reader.
Since EZ Reader will enable pre-sorting of mail into logical business process categories (automatic response or manaul review with different levels of priority), manual intervention will be streamlined.

Manual Processing

Manual processing is a mode of operation within EZ Reader that allows one to selectively choose the emails that are being processed (from a text file). This environment makes EZ Reader readily accessible for maintenance and testing purposes. The basic process flow of emails (retrieval, interpretation, response) is equivalent to continuous processing. Essentially, the main difference between the two processing modes is the location from where EZ Reader retrieves incoming email messages and where the output is directed to: from a Lotus Notes mailbox, or from a text file.

Ease of Use

EZ Reader must incorporate features that save time and effort for customer service representatives in their role of reading and responding to email messages.

Minimum operator intervention.

EZ Reader is an *embedded* application. This means that it runs stand-alone and is invisible to the user. Visually, it is a layer between the incoming Internet email and ChaseDirect's Customer Service Representatives (CSRs) in Rochester. Once EZ Reader is activated, it remains running without manual intervention. In this sense, there is no need to train Chase people on usage.

Reduction in man-hours.

All incoming emails are processed by EZ Reader. EZ Reader interprets all messages, determining that some messages can be responded to automatically. These will be filtered out and processed without manual intervention. The remaining messages will be referred to CSRs in Rochester.

Access to processing status

EZ Reader includes capabilities for managing and monitoring processing. EZ Reader displays status information during production and also writes to a Lotus Notes archive database for viewing, reporting, and archival purposes.

Performance

EZ Reader throughput, accuracy and timeliness need to be measured over time in order to validate performance and to predict the manual and system impacts of increasing volumes of email. EZ Reader will collect data needed to produce performance statistics.

In the first phase of EZ Reader implementation, two knowledgebase tests were performed. The test bed consisted of 3 weeks worth of real Internet emails, responded to by the ChaseDirect Technical Support staff, but also processed by EZ Reader.

Application response time performance.

Processing start time and completion time were recorded by the system during the knowledgebase tests. On a Pentium 75 with 32 MB of memory, EZ Reader throughput averaged 1 message per second.

Interface response time performance.

Because of the need to externally interface with Lotus Notes, email retrieval and automatic response time may add a few more seconds.

Overall response time performance.

EZ Reader response time requirements have not been set by Chase. The Customer Knowledge team has taken steps to optimize processing response time both in the foreground and the background.

EZ Reader should meet response time performance requirements without forcing hardware or memory upgrades beyond the standard ART**Enterprise*® configuration.

Response Accuracy

Testing of the knowledge-base portion of EZ Reader has shown accuracy of responses. The accuracy is measure in two ways: did EZ Reader classify the incoming email correctly (**"A"** = automatic response, **"R"** = referral, or **"D"** = detected) and if the classification is correct, is the actual recommended action correct, e.g., is it referred for the correct reason, is the automatic response appropriate for that email, etc.

In the first phase of EZ Reader implementation, two knowledgebase tests were performed. The test bed consisted of 3 weeks worth of real Internet emails, responded to by the ChaseDirect Technical Support staff, but also processed by EZ Reader. There were 190 incoming emails, for which EZ Reader responded to with an accuracy of approximately 94% (in the second test dated approximately December 13, 1995) with approximately 130 (or 68%) referrals. The breakdown of referrals is shown below:

50%	MS Net request forms
30%	Sign-Up Kit requests
5%	Referenced specific person within ChaseDirect
5%	Contained material defined as sensitive
5%	Contained multiple questions or were lengthy emails
5%	Other (e.g., foreign addresses)

Note that although the percentage of emails responded to automatically was under target (approximately 50% auto-responses desired), this was due to the business requirements at the time of the test. It is important to understand that the breakdown of automatic response vs. referrals is highly dependent upon the specifications set forth by the ChaseDirect business staff. As time passes, and ChaseDirect creates more canned responses, the percentage of automatic responses will rise. Also, there is an obvious rise in automatic responses whenever there is a large promotion (e.g., an ad in the New York Times which contains ChaseDirect's email address).

Security Issues

It is currently an established fact that the Internet is an unsecured environment. Emails could be read by any number of persons as it is routed from its source to its destination. For this reason, there are few ways to secure a piece of email from a customer to ChaseDirect's Internet domain. Once in ChaseDirect's Lotus email environment, however, passwords are required in order to retrieve email. Only those persons within ChaseDirect's Technical Support staff having the Notes passwords have access to the incoming emails.

The EZ Reader/Lotus Notes connection employs intermediate text files[2] for passing emails to and from the two Windows applications. The security of emails is ensured using this method because it overwrites the contents immediately after they have been read. For a detailed description of these intermediate text files and the status file, please refer to the Maintenance section of the document.

[2] Ideally, a clean interface via DLL calls is desired, but because of current limitations for simulated-32-bit ART*Enterprise® and the 16-bit VIM.DLL, there were enough complications to warrant a work-around solution involving separate C programs and external text files.

Part Two - EZ Reader Design and Environment

This section provides a high-level description of the technical environment, design objectives and constraints, architecture and components of EZ Reader.

EZ Reader Version 1.0 Environment

EZ Reader version 1.0 was developed using ART**Enterprise®* 2.0 for Windows, interfacing with Lotus Notes via the VIM protocol and Notes API[3].

The EZ Reader application is built using specialized software known as *knowledge-based* software.

An Integrated Knowledge-Based Application

This specialized software is used at other financial services institutions such as American Express, Dun & Bradstreet, Fannie Mae, GE Capital and Swiss Bank.

Knowledge-based software systems can interpret data from a wide variety of sources and display expert advice to a business transaction or to automate a decision step. Unlike conventional software that only displays or calculates data, *knowledge-based software* interprets data using *rules of thumb*.

Design

The EZ Reader application is a software program that will reduce the amount of manual intervention required to process incoming emails in ChaseDirect. EZ Reader will be implemented in multiple Phases.

Phase I of EZ Reader will:

- automatically classify each incoming email;

[3] VIM is the acronym for Vendor Independent Messaging and API stands for Application Programming Interface, which is necessary for two separate Windows-based applications to communicate, i.e., pass data, between each other.

- select a model response for each email from a library of replies;
- automatically set a priority level for each email;
- prepare an acknowledgement receipt to the sender, where appropriate; and
- route the original email, model response and acknowledgement receipt (if any) into an appropriate mailbox in Lotus Notes. From Lotus Notes. the emails may be automatically sent and/or reviewed.

Proposed enhancements to EZ Reader are described in Appendix C - Future Releases of EZ Reader.

Technical Objectives

The primary objectives to be met in the EZ Reader technical design are as follows:

Design for speed and ease of use

- ◆ EZ Reader is an embedded system which lies between Lotus Notes and the ChaseDirect Technical Support group. A very small amount of graphical interface is supplied, because very little interaction is required from the user.

Design to enable processing progress monitoring

- ◆ Application administrators need an efficient means for process monitoring and reporting. EZ Reader makes this information available through the archival Notes database, as well as on-line viewing of the current email being processed.

Design for flexibility, scalability, extensibility, ease of maintenance and ease of operation

- ◆ The knowledge-based software used to build EZ Reader inherently adheres to these specifications. This software has been used to deploy very large commercial applications that require these characteristics.

EZ Reader Architecture

EZ Reader is a minimally-interactive client/server application that interfaces with Lotus Notes email software.

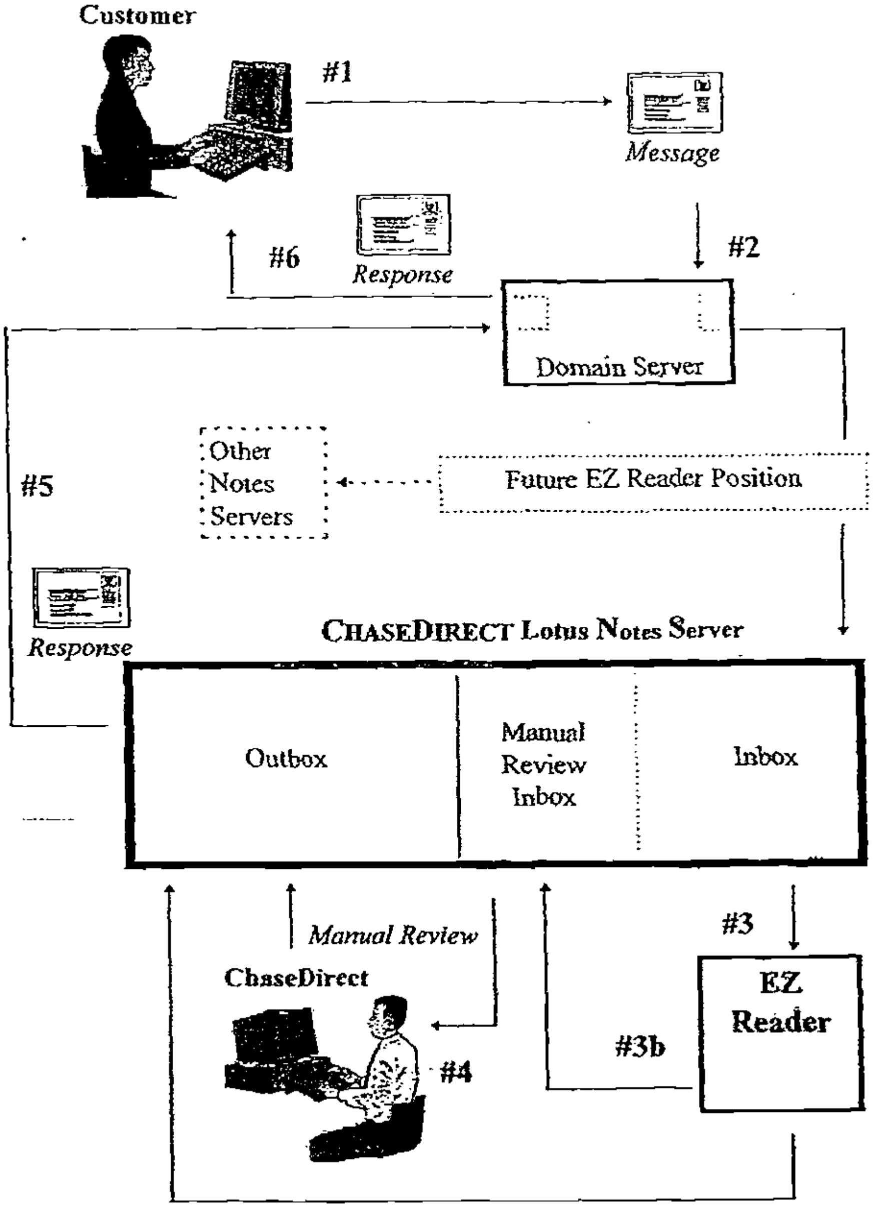


Figure 1: EZ Reader High-Level Architecture

As shown in Figure 1 above, the EZ Reader system is a component client/server system. Processing takes place as follows (the three major steps, email retrieval, interpretation, and response are denoted in boldface):

Process Flow Description

1. The customer sends an E-mail message to Chase Manhattan Bank's Internet address.

2. **RETRIEVAL:** Chase's corporate E-mail router passes the message to ChaseDirect's Lotus Notes mail management system. EZ Reader checks the Lotus Notes mail database for new mail throughout the day.

3. **INTERPRETATION:** As a new piece of mail comes in, EZ Reader retrieves the message and compares it to a library of actual customer messages, and categorizes the message. Categories are described in the section labeled "Key to Categories". EZ Reader recommends actions and responses based on the message's category and priority and routes the mail to one or more Lotus Notes mailboxes according to one of two action types:

   a) The mail message is responded to automatically by EZ Reader. Automatic responses are routed directly to the outbox.

   b) The mail message cannot be responded to automatically and is referred to ChaseDirect for human review and response by being placed in another inbox. EZ Reader assigns a priority to the message and suggests a response, based on message type. Priorities are described in the section labeled "Key to Categories - Referrals".

4. **RESPONSE:** ChaseDirect reviews and assigns responses to all messages referred by EZ Reader and places these responses in the outbox.

5. ChaseDirect uploads messages in the outbox to the Lotus Notes server for transmission back to the customer.

6. The customer receives ChaseDirect's E-mail reply.

Knowledge-base module

Interpretation

The knowledgebase portion of EZ Reader, written in the ART**Enterprise*® language, combines case-based analysis and rule-based reasoning to interpret incoming email messages. Rules are used to drive the flow of processing, but also are utilized in a pre-processing phase, to identify and tag certain characteristics of a message. A case-based retrieval is then performed, searching for the best matching case of the current email against the casebase. If any characteristics were tagged in pre-processing phase, they will contribute to the overall casebase score. Details on the knowledgebase can be found in the Maintenance section of this document.

EZ Reader/ Lotus Notes connection

Retrieval and response

The ART**Enterprise*® module of EZ Reader must still get its input data from somewhere and also have a place for its output. This is where the interface to Lotus Notes comes into play. The EZ Reader/Notes connection is accomplished using the VIM API[4] as well as Notes' own API. The Vendor Independent Messaging interface is an industry-standard application programming interface that provides the messaging (i.e., sending and retrieving mail) functionality provided by Notes or any full-featured messaging system. VIM allows programmers to develop mail-enabled applications that are portable across various email systems. A listing of the VIM and Notes API functions used by EZ Reader can also be found in the Maintenance section.

A Note about Nuisance Message Handling

There will undoubtedly be some messages that EZ Reader will not be able to interpret, e.g., nuisance messages.[5] All uninterpretable messages will be forwarded to a Lotus Notes mailbox that contains messages that are set aside for manual evaluation.

[4] Please refer to the EZ Reader Glossary in Appendix B - EZ Reader Glossary for further details.

[5] A reasonable goal for EZ Reader would be to interpret and process 50-80 percent of all messages.

Constraints

EZ Reader is designed to operate within a Windows 3.1 environment with an interface to Lotus Notes. EZ Reader's functionality depends upon the existence and continuous operability of programs to retrieve and deliver email messages to and from Lotus Notes. If EZ Reader loses its Notes connection, it will automatically self-abort.

Part Three - EZ Reader Usage

Part Three describes how to use EZ Reader, with an emphasis on how to initialize, monitor, and terminate an EZ Reader session.

Training

Chase people will not need any special training to use EZ Reader.

Input to EZ Reader is performed automatically between Lotus Notes and EZ Reader by the Windows VIM API functions of EZ Reader.

EZ Reader results are sent to targeted Lotus Notes mailboxes and displayed as email messages.

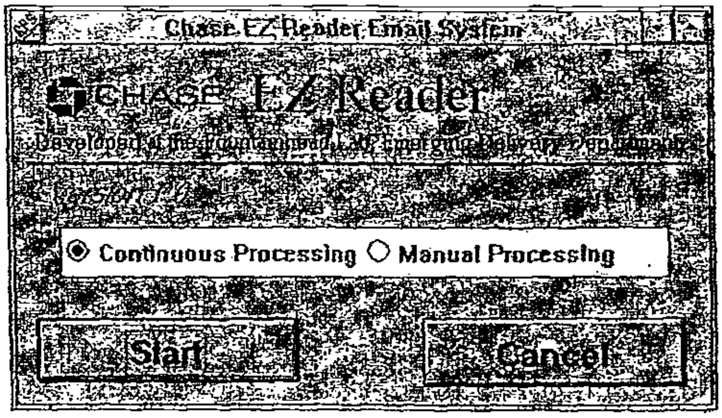
EZ Reader Interface

Because EZ Reader is an embedded system, there is practically no external interface for interacting with it. Once started, EZ Reader will display a status screen (ART**Enterprise*®'s Command Interpreter window) which shows each incoming email message along with EZ Reader's recommendations on how to best respond to it. This can be used to view the progress of EZ Reader, although emails may be processed rather quickly and subsequently, the output messages may scroll by too quickly to read. All processed emails are archived in a separate Notes database, which one may access at any time to review the incoming emails. This archive will also be used for statistical and reporting purposes.

Starting EZ Reader

EZ Reader is invoked by double-clicking on the EZ Reader icon in the program group **EZ Reader** in the Windows Program Manager.

Invoking EZ Reader brings up the EZ Reader Greeting Window which displays the Chase logo and other identifying information (see Figure 2).



◉ Continuous Processing ○ Manual Processing

Figure 2: EZ Reader Greeting Window

EZ Reader functions in one of two modes: *continuous processing* and *manual processing*. Ma.ual processing should be use l when modifying, testing, or demo-ing EZ Reader code whereas continuous processing is used for production.

Continuous Processing

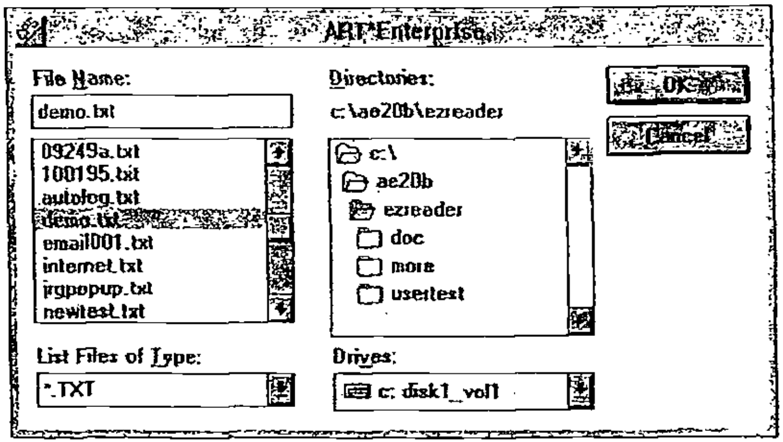
If the **Continuous Processing** radio button is selected and the **Start** button is clicked, then EZ Reader will begin to continuously retrieve emails directly from Lotus Notes and process those messages. The **Cancel** button, if pressed, will result in immediate termination before any EZ Reader processing even begins. EZ Reader is designed to run continuously without human intervention; it is not recommended to halt processing frequently. If it does becomes necessary to take EZ Reader off-line (for maintenance or upgrade purposes), then simply press the **Break** button at the top of the Command Interpreter window during continuous processing. This will break the ongoing communication with Lotus Notes until you continue or abort EZ Reader processing. At this point, you can take one of two actions:

- To continue processing, type :c in the bottom of the window.
- To abort EZ Reader, type **(halt)** to stop the rule-base engine from firing, then type **(ezr:abort)** which is a function that ends EZ Reader processing and performs clean-up routines. Then

when you are ready to place EZ Reader on-line again, follow the start-up procedures as described above.

Manual Processing

Manual processing requires input from an external text file (selected by the user from the File Selection Window (see Figure 3), with the file extension ".txt". This text file should contain one or more emails in a format which replicates that of emails downloaded directly from the Internet.[6] Contents of a test file in the EZ Reader directory. called **DEMO.TXT** follows in Figure 4.



File Name: demo.txt

09249a.txt
100195.txt
autolog.txt
demo.txt
email001.txt
internet.txt
jrgpopup.txt
newtest.txt

Directories: c:\ae20b\ezreader

c:\
ae20b
ezreader
doc
more
usertest

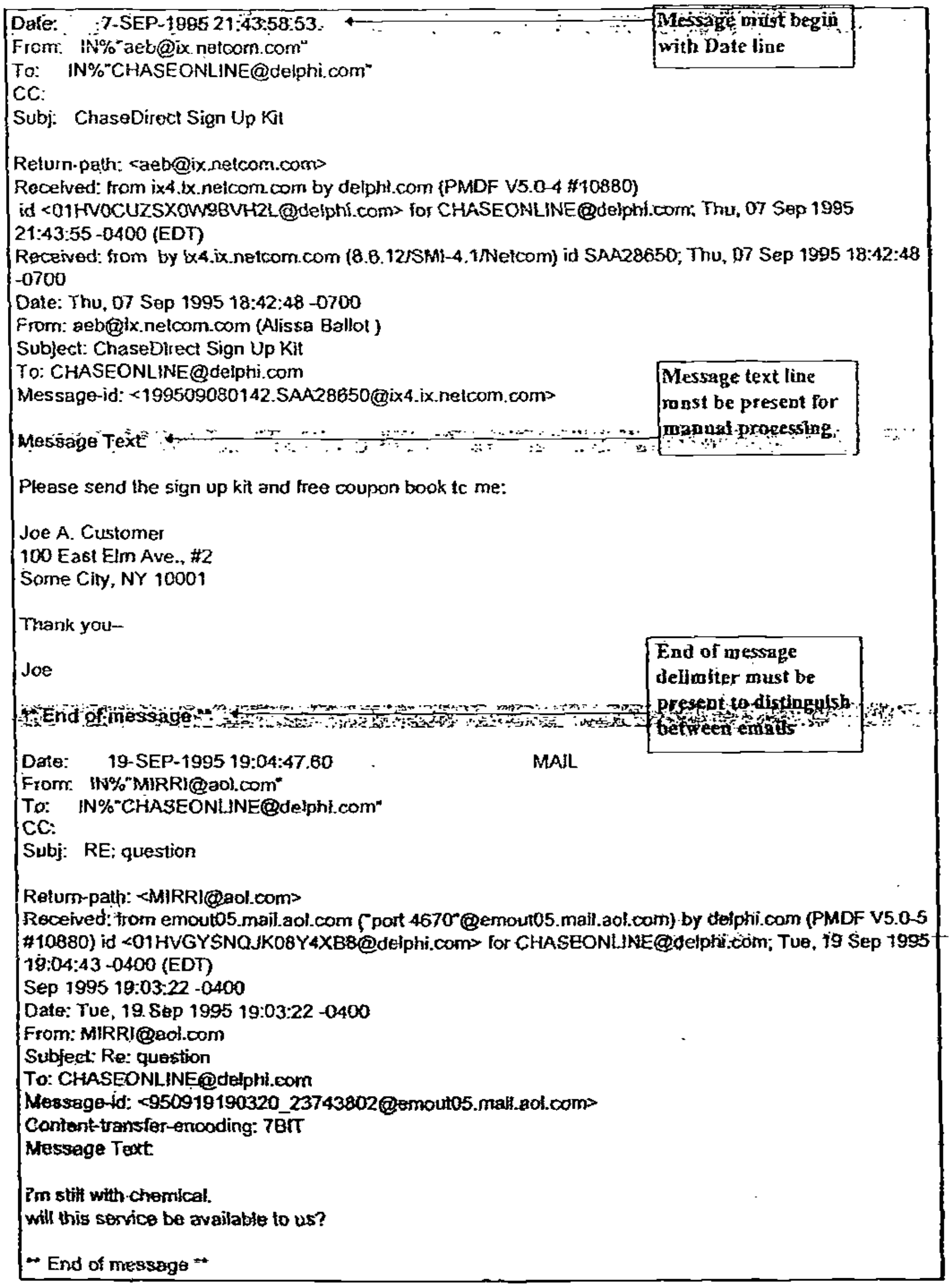
List Files of Type: *.TXT

Drives: c: disk1_vol1

Figure 3: Manual Processing File Selection Window

When creating your own test data, it is advised to use the samples in **DEMO.TXT** as a template. Simply tailor the body of an email message to reflect that which you wish to test. Then, to actually run EZ Reader against the test email(s) you created, simply run EZ Reader as usual, click the **Manual processing** radio button, then select the name of your file from the File Selection Window.

[6] For manual processing, EZ Reader looks for a line containing "**Message text:** " to determine the end of the email header. A single email begins with the "**Date:** " line and ends with the line "** **End of message** **". If you are creating your own test file, make sure these lines exist in the appropriate positions; otherwise, EZ Reader will not be able to discriminate between the header and the body of the message, and also between multiple messages in the same text file. This is taken care of in continuous processing by the Lotus Notes API.



```
Date:     7-SEP-1995 21:43:58.53
From:   IN%"aeb@ix.netcom.com"
To:     IN%"CHASEONLINE@delphi.com"
CC:
Subj:   ChaseDirect Sign Up Kit

Return-path: <aeb@ix.netcom.com>
Received: from ix4.ix.netcom.com by delphi.com (PMDF V5.0-4 #10880)
 id <01HV0CUZSX0W9BVH2L@delphi.com> for CHASEONLINE@delphi.com; Thu, 07 Sep 1995
21:43:55 -0400 (EDT)
Received: from  by ix4.ix.netcom.com (8.6.12/SMI-4.1/Netcom) id SAA28650; Thu, 07 Sep 1995 18:42:48
-0700
Date: Thu, 07 Sep 1995 18:42:48 -0700
From: aeb@ix.netcom.com (Alissa Ballot )
Subject: ChaseDirect Sign Up Kit
To: CHASEONLINE@delphi.com
Message-id: <199509080142.SAA28650@ix4.ix.netcom.com>

Message Text:

Please send the sign up kit and free coupon book to me:

Joe A. Customer
100 East Elm Ave., #2
Some City, NY 10001

Thank you--

Joe

** End of message **

Date:     19-SEP-1995 19:04:47.60                    MAIL
From:   IN%"MIRRI@aol.com"
To:     IN%"CHASEONLINE@delphi.com"
CC:
Subj:   RE: question

Return-path: <MIRRI@aol.com>
Received: from emout05.mail.aol.com ("port 4670"@emout05.mail.aol.com) by delphi.com (PMDF V5.0-5
#10880) id <01HVGYSNQJK08Y4XB8@delphi.com> for CHASEONLINE@delphi.com; Tue, 19 Sep 1995
19:04:43 -0400 (EDT)
Sep 1995 19:03:22 -0400
Date: Tue, 19 Sep 1995 19:03:22 -0400
From: MIRRI@aol.com
Subject: Re: question
To: CHASEONLINE@delphi.com
Message-id: <950919190320_23743802@emout05.mail.aol.com>
Content-transfer-encoding: 7BIT
Message Text:

I'm still with chemical.
will this service be available to us?

** End of message **
```

Message must begin with Date line

Message text line must be present for manual processing.

End of message delimiter must be present to distinguish between emails

Figure 4: Sample emails found in DEMO.TXT

Output from manual processing appears in the Command Interpreter window while EZ Reader is running. Although you can scroll back through the output, there is a limitation to the amount of data in the Command Interpreter buffer. For this reason, if your test file is quite large, or if you want to print a hardcopy of the output, use the dribble command to dump the output to a text file.

Type the following command before you type the **(run)** command to invoke processing:

```
(dribble "c:\\ae20b\\ezreader\\output.txt")
```

or whatever path[7] and filename you wish. Then, once EZ Reader has finished processing the text file of test emails, it will complete its own processing, leaving you back at the command prompt in the Command Interpreter. At this point, remember to close the dribble file by typing the **(dribble)** command. Otherwise, all other text that subsequently appears in the Command Interpreter will also get dumped to that open output file!

[7] Make sure to include a double back-slash when specifying a path. This is because the back-slash is ignored and will also "escape" the next character.

Part Four - EZ Reader Maintenance

Part Three contains a detailed description of the maintenance procedures for EZ Reader.

Maintenance

When designing and maintaining any knowledge-based system, one should keep in mind the following 3 levels:

1. Business stratum
2. Data stratum
3. Knowledge stratum

With respect to maintenance, the business layer is one of the driving forces behind any modifications to EZ Reader. EZ Reader must reflect current business requirements for responding to customers via the Internet.

The data stratum includes such things as identifying what data (and the source of this data) is needed by the business user, mapping the data to objects in ART**Enterprise*®, adding access methods for new data, addressing any issues related to object destruction or storage, etc. In the case of EZ Reader, the data comes from one uniform source: Lotus Notes. Lotus Notes, in turn, receives its data from the Internet (including the World Wide Web in future releases of EZ Reader) which means that the actual data source is invisible to EZ Reader. Consequently, the data layer does not play as important a role in this system as in a data-critical application such as Milestone Advisor. Changes to the Lotus Notes configuration, however, may result in updates to EZ Reader interface to Notes (i.e., the VIM API). In addition, changes in the business stratum may require updates to ART**Enterprise*®/EZ Reader objects.

Once the data connection (between EZ Reader and Lotus Notes) is in place, the most important piece of EZ Reader is the accurate interpretation of incoming emails. This is embodied in the knowledge stratum. Maintenance procedures for the knowledgebase may include modifying the rulebase, casebase, and actionbase.

Details of each of the 3 strata follow:

The following business-related events may result in EZ Reader maintenance:

Business Changes Requiring EZ Reader Maintenance

- A new or revised association between a type of message and its prepared response is required. For instance, if emails concerning a marketing promotion need to be answered differently than originally, then EZ Reader maintenance should be requested or performed. (Maintenance is not required when response text is changed or responses are developed that are not associated to messages received by EZ Reader.)
- A new type of action needs to be associated with a prepared response. For instance, if emails concerning a new marketing promotion need to be processed automatically, then EZ Reader maintenance should be requested or performed.
- A new informational action needs to be recommended by EZ Reader when it detects a particular type of incoming message, regardless of the response. For instance, if emails from foreign customers need to be flagged, then EZ Reader maintenance should be requested or performed.

In addition, changes in the environment of EZ Reader may also result in EZ Reader Maintenance:

Technical Changes Requiring EZ Reader Maintenance

Whenever a change in the technical environment around EZ Reader is planned, the application should be retested and assessed for needed changes. For example, if a new version of Lotus Notes will be installed, the application should be tested and assessed for changes before the production installation is done.

Maintenance Procedures

Until Phase II of the project is complete, a team effort is needed in order to maintain the complete EZ Reader business system. Our primary objective for the period of time surrounding application deployment will be a complete transition of the development and maintenance for EZ Reader to Chase employees who have been trained by experienced knowledge base consultants.

Under normal circumstances, requests for business changes should be able to be accommodated within a few days in Phase I of EZ Reader.

Skills Needed for EZ Reader Maintenance

The following types of expertise must be coordinated in order to perform EZ Reader maintenance.

- Business knowledge of rules for outgoing correspondence.
- Chase electronic mail system architecture and plans.
- Knowledge Acquisition for knowledge systems development.
- ART*Enterprise® rulebase and document integration application programming, testing and debugging.
- Rule-based systems concepts.
- Case-based reasoning concepts.
- Lotus Notes software testing and debugging.
- Microsoft C++ and Notes VIM API software testing and debugging.
- PC network setup and operation.

Modifying ART**Enterprise*® *and related EZ Reader Code*

> It is strongly recommended that one read and understand the ART*Enterprise®* documentation (especially for an understanding of rules and case-based reasoning) *before* attempting to make modifications to the EZ Reader code. At the very least one should make a copy of the relevant original files (e.g., ez-rules.art, ez-cases.art, ez-acts.art) before modifying them.

The EZ Reader ART**Enterprise*® code is divided into logically independent modules, which are each contained in a text file with the **.art** extension. In addition, EZ Reader calls several external C programs (executable binaries with the **.exe** file extension), which contain the API to Lotus Notes. These C programs make use of three external text files (having the extension .txt) for passing emails and status messages. These files are listed in Table I:

Table 1: EZ Reader file structure

Filename	Description
ez-acts.art	The set of actions that EZ Reader can recommend (the actionbase)
ez-cases.art	The set of cases against which the current email is compared (the casebase)
ez-cb.art	Contains functions and objects related to the casebase
ez-globs.art	Contains global variable and miscellaneous functions
ez-greet.art	Contains EZ Reader Greeting Window objects
ez-rules.art	The set of phase-processing rules and question rules (the rulebase)
filemsg.exe	VIM API module to retrieve the next unread email message from Lotus Notes
write-db.exe	Notes API module to write the response to a Lotus Notes database
markmsg.exe	VIM API module that marks the current email message as "read"
filemsg.out	Text file containing the retrieved email
sendmsg.out	Text file containing the original email appended with EZ Reader recommendations (the response)
statmsg.out	Text file containing status/error messages generated by the API modules during runtime

Before describing how to maintain the knowledge portion of EZ Reader (i.e., the rulebase, casebase, and actionbase), it is helpful to be familiar with the internal process flow of EZ Reader code, as diagrammed in Figure 5.

The basic flow of EZ Reader is similar for both continuous and manual processing. The flow of processing is driven by rules. Each rectangle in Figure 5 is in actuality a *phase-processing rule.* The remaining rules in the rulebase are *question rules*, which fire when EZ Reader is in the pre-processing phase, (the **ez-reader:mail:process:pre** rule). Question rules will be discussed in detail later; basically, they are used to discern information about the content of an incoming email message and to recommend actions accordingly.

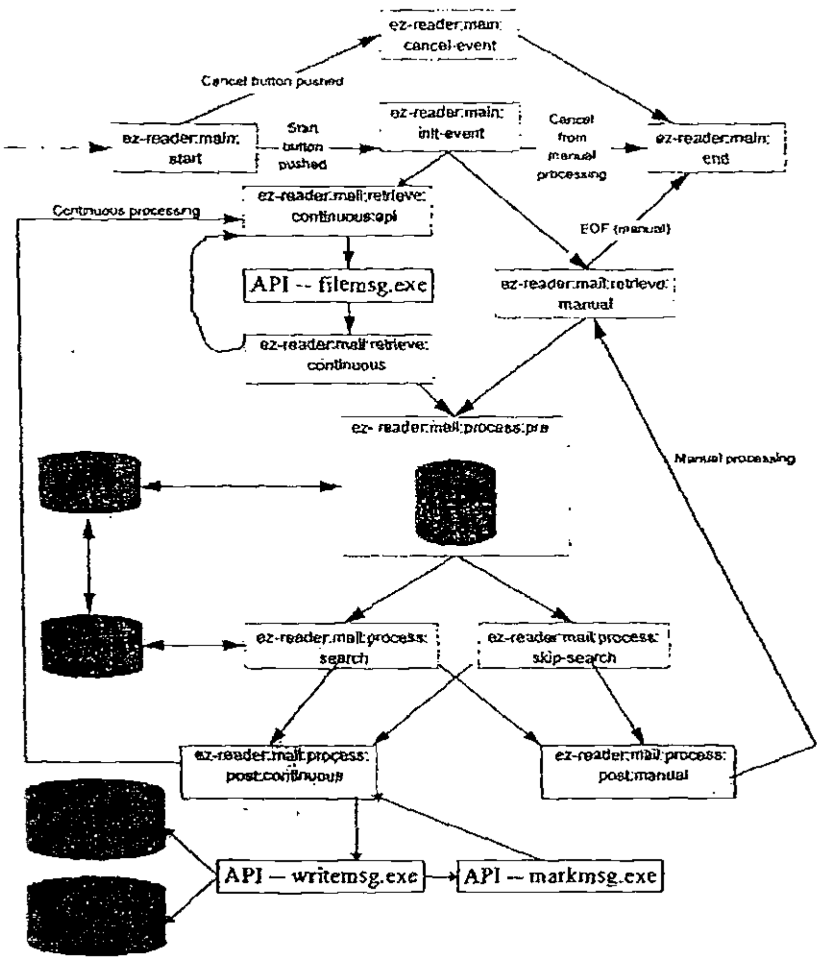


Figure 5: EZ Reader internal code process flow

The flow of phase-processing rules is described step-by-step as follows:

1. **EZ-READER:MAIN:START**: main initialization phase; displays the greeting window, opens the output log file, loads the casebase, etc.
   a) **EZ-READER:MAIN:INIT-EVENT**: if the **Start** button is pushed, this phase determines whether continuous processing or manual processing has been activated.

i) **EZ-READER:MAIL:RETRIEVE: CONTINUOUS:API**: this phase initiates a call to the C API module that checks for new messages.

ii) **EZ-READER:MAIL:RETRIEVE: CONTINUOUS**: if new messages are found, this phase retrieves the next unread email from the Lotus Notes *Chase Direct-In* box if EZ Reader is in continuous process mode. If EZ Reader is in manual processing mode, the next unread email message from the input text file is retrieved.

iii) **EZ-READER:MAIL:PROCESS:PRE**: contains question rules which flag certain characteristics of the current email (see the section below on ***Rule maintenance*** for more details). and determines whether or not a search on the casebase is necessary or not.

iv) **EZ-READER:MAIL:PROCESS:SEARCH**: if the pre-processing phase deems it necessary to search the casebase, then this phase calls the function which performs the search. If it is not necessary to search the casebase for the current email, then the next phase is **EZ-READER:MAIL: PROCESS:SKIP-SEARCH.**

v) **EZ-READER:MAIL:PROCESS:POST**: this phase determines the actual recommended actions, and writes the original message appended with the set of actions, category, and priority back to a Lotus Notes database.

vi) Repeat steps (i) thru (v). In continuous processing mode, this loop runs until interrupted. When there is no unread message in the *Chase Direct-In* box, EZ Reader will wait patiently until the next email arrives. For manual processing, this loop terminates when no more email messages remain in the input file.

b) **EZ-READER:MAIN:CANCEL-EVENT**: if the **Cancel** button is pushed in the Greeting Window, this phase will end all processing immediately and sets the phase to trigger **EZ-READER:MAIN:END**.

l. **EZ-READER:MAIN:END**: Closes the output log file, and performs general housekeeping tasks.

Rule maintenance

Both the phase-processing and question rules[8] used by EZ Reader (collectively known as the rulebase) reside in an **.art** file called **ez-rules.art** within the EZ Reader subdirectory. In EZ Reader, rules are used in two different ways: as *phase-processing rules* and as *question rules*. Eleven rules are related to the process flow of the system, which fire depending on certain window pushbutton events and fact assertions (*phases*). Each phase denotes a single state of processing, only one of which is relevant at a time. The eleven phase-processing rules along with their corresponding phases are detailed in Table 2.

Table 2: Phase-processing rules

Rule Name	Phase(s)/event(s) that triggers rule
ez-reader:main:start	none – always fires upon start
ez-reader:main:init-event	(gtk:event activate ergw:pb:start $?) – fires when **Start** button is clicked in Greeting Window
ez-reader:main:cancel-event	(gtk:even activate ergw:pb:cancel $?) – fires when **Cancel** button is clicked in Greeting Window
ez-reader:mail:retrieve:continuous:api	(phase ready-for-next-email), (continuous-process), (api)
ez-reader:mail:retrieve:continuous	(phase ready-for-next-email). (continuous-process), (no-api)
ez-reader:mail:retrieve:manual	(phase ready-for-next-email), (manual-process)
ez-reader:mail:process:pre	(phase ready-to-preprocess-email)
ez-reader:mail:process:skip-search	(phase ready-to-process-email) and (phase skip-search)
ez-reader:mail:process:search	(phase ready-to-process-email) and (not (phase skip-search))
ez-reader:mail:process:post:continuous	(phase ready-to-postprocess-email), (continuous-process)
ez-reader:mail:process:post:manual	(phase ready-to-postprocess-email), (manual-process)
ez-reader:main:end	(phase end-of-processing)

A phase-processing rule for EZ Reader takes the form:

```
(define-rule name-of-rule
  "Description of rule"
  (declare (salience ?salience:*salience of rule*))
  ?x <- (phase phase-for-rule-to-fire)
```

[8] For a detailed explanation of rule-based reasoning, including fact assertion, rule firing, and salience, please refer to the ART*Enterprise® online help called the ***Function and Class Reference Library*** (the FACLR), or the hardcopy reference manuals.

```
. . . any other statements that must be true in order
     for the rule to fire . . .
=>
   (retract ?x)  ; retract the current phase
. . . actions relevant to the current phase . . .
   ; set the next phase
   (assert (phase next-phase-in-processing))
)
```

Following this form, the rule for **ez-reader:mail:process:search** looks like:

```
(define-rule ez-reader:mail:process:search
   "Find best fitting matches from the casebase for the
        current mail message"
   (declare (salience ?salience:*very-high*))
   ?x <- (phase ready-to-process-email)
   (not (phase skip-search))
=>
   (retract ?x)
   (ercb:search)  ; calls the casebase search function
   (assert (phase ready-to-postprocess-email))  ; assert the next phase
)
```

The second kind of rule, a *question rule* in EZ Reader terminology, is used for tagging certain characteristics or (or answering certain questions about) the current email and specifying action(s) based on the answers. There are three subclasses of question rules:

1. Action-setting rules
   a) add a detected action
   b) add an automatic response action
   c) add a referral action
1. Attribute-setting rules
2. Action- and attribute-setting rules

Table 3 lists the current set of question rules (in alphabetical order) along a brief description and what type of rule it is:

Table 3: Question rules

Rule name	Question: Does the email/customer:	Type/action
q:address?	reference customer's address?	attribute- and action-setting: detected
q:blank?	contain an empty/blank body?	action-setting: automatic
q:cancel?	request cancellation of services?	action-setting: referral
q:change-of-address?	request a change of customer's address?	action-setting: referral
q:do-not-call-customer?	request specifically not to be called?	attribute- and action-setting: detected

q:fax?	contain a fax number?	action-setting: detected
q:fix-service?	request service or describe an error?	attribute-setting
q:investment?	contain investment references?	attribute-setting
q:foreign?	reference a foreign country?	attribute- and action-setting: detected
q:lengthy?	contain a long message?	attribute-setting
q:macintosh?	reference Macintosh/Apple products?	attribute-setting
q:merger?	reference the Chase/Chemical merger?	attribute-setting
q:multiple?	contain multiple questions?	attribute-setting
q:pay-on-line?	contain references to using Pay-On-Line?	attribute-setting
q:person-referenced?	reference a specific (Chase) person?	attribute- and action-setting: detected
q:phone?	contain a phone number?	action-setting: detected
q:replacement?	reference to a lost or stolen ATM/credit card	action-setting: referral
q:sensitive?	contain information defined as sensitive?	action-setting: referral, auto, and detected

Question rules are set to fire only during the pre-processing phase (see Figure 5) where the fact **(phase ready-to-preprocess-email)** must currently be asserted (refer to Table 2 for the list of phase-processing rules and their corresponding fact assertions). This could be easily changed if necessary. In general, the form of a question rule is:

```
(define-rule name-of-rule?
  "Description of rule"
  (declare (salience ?salience:*high*)) ; usually *high*
  (phase ready-to-preprocess-email) ; usually this phase
  (test
. . . any other relevant conditions here . . .
  )
=>
. . . rule processing here . . .
)
```

So, the question rule for handling emails referencing the merger (**q:merger?**) looks something like:

```
(define-rule q:merger?
  "Does the message reference the Chase/Chemical merger?"
  (declare (salience ?salience:*high*))
  (phase ready-to-preprocess-email)
  (test
    (ercb:in-mail-body? (create$
       "MERGER" "CHEMICAL" "CHEMBANK"
       "CHEM SELECT")))
=>
  (ercb:add-question merger?) ; calls attribute-setting function
)
```

By looking at the right hand side of the rule (lines after the => symbol), we see that **q:merger?** is an attribute-setting rule. This means that the **merger?** attribute of **class:case** is set to **"Yes"** for the present case (result of calling the attribute-setting function). Then, if a search against the casebase is performed, the **merger?** attribute will contribute to the resulting match score.

The rule for detecting sensitive information (**q:sensitive?**) is a bit more complicated. In this case, because ChaseDirect business rules require that all emails containing material defined as sensitive be referred, EZ Reader skips the casebase search and simply refers the email (also adding the corresponding detected action):

```
(define-rule q:sensitive?
   "Is sensitive material included (e.g. account #)?"
   (declare (salience ?salience:*high*))
   (phase ready-to-preprocess-email)
   (object present-case
      (description ?desc))
   (test
      (or
        (ercb:in-mail-body? (create$
           "AC#" "ACCT#" "ACCT #" "ACCT NO" "ACCT. #"
           "ACCT: NO." " PIN#" " PIN NO" " PIN NUMBER"
           "START UP PIN" "CLDB" " SSN" " SS#"
           "SOCIAL SECURITY NUMBER" "ACCOUNT NUMBER IS"
           "MY ACCOUNT:" "ACCOUNT#" "ACCOUNT #"
           "ACCOUNT. #" "ACCOUNT NO.:" "ACCOUNT NUMBER:"
           " PIC " "AUTHORIZATION CODE" "SECRET CODE"
           "PERSONAL IDENTIFICATION CODE"
           "PERSONAL IDENTIFICATION NUMBER"))
        (masked-member$ "999-9-999999" ?desc)  ;checking account
        (masked-member$ " 999999999 " ?desc)   ;savings account
        (masked-member$ "999-999999" ?desc)    ;savings account
        (masked-member$ "9-999-999999" ?desc)  ;time deposit
        (masked-member$ "999-99-9999" ?desc)   ;social security #
     ))
=>
   (ercb:add-question sensitive?)  ;not really necessary

   ; Set the referral action here
   (write ?*log* "Contains sensitive information; refer.")
   (bind ?*case-title* "Contains sensitive information; refer.")
   (bind ?*action-list* (create$ action:refer:service:priority:2))

   ; Also tag the email with a detected action
   (bind ?*detected-actions*
     (action:append ?*detected-actions*   (create$ action:detect:sensitive)))

   ; Skip the casebase search
```

```
        (assert (phase skip-search))
      )
```

Note that any question rule can fire in the pre-processing phase if the current email message passes that rule's criteria. So, in the q:sensitive? example above, in addition to sensitive information being detected in this email, if the customer also included their address, phone, and fax numbers, all of those corresponding rules will fire, resulting in a recommended action set containing all of the following

```
action:refer:service:priority:2
action:detect:sensitive
action:detect:address
action:detect:phone
action:detect:fax.
```

Case maintenance

The text file ez-cases.art contains all cases found in the EZ Reader casebase. Cases are instances of the class class:case which contains the following attributes:

```
(define-class class:case object
  (number-key) ; casebase requirement -- unique value
  (title)      ; descriptive text for the case
  (subject)    ; currently not used in casebase matching
  (description) ; body of email message
  (actions)    ; key pointer(s) to action db (multiple cardinality)
  (category)   ; used for outbox sorting
  (priority)   ; used for outbox sorting
 ; Note: only those attributes that contribute to the
 ;     case base match score are set to "No" and "Yes".
 ;     The other attributes are used for detection only.
  (address? "No")
  (blank?)
  (do-not-call-customer? "No")
  (cancel? "No")
  (fax?)
  (fix-service? "No")
  (foreign? "No")
  (investment? "No")
  (lengthy? "No")
  (macintosh? "No")
  (merger? "No")
  (multiple? "No")
  (pay-on-line? "No")
  (person-referenced? "No")
  (replacement? "No")
  (sensitive? "No")
  (phone?)
; following attributes not used, but could be for auditing purposes
  (date-of-creation)
  (time-of-creation)
  (date-last-modified)
  (time-last-modified)
  (author)
)
```

To add, modify, or delete cases in the casebase, there is no need to use ART**Enterprise*®; simply use a text editor. To add a case to the casebase, follow the **make-instance**[9] form (copy/paste from existing cases is an easy and less error-prone way to add a new case).

```
(make-instance class:case :name case-name    ;unique case name
   number-key  case-key                      ;unique number key
   title case-title
   subject email-subject   ;not currently used in casebase matching
   description pertinent-part-of-email-body
   actions (create$ some-action1 some-action2 . . .)
   relevant-attribute? "Yes"
)
```

A simple case, then, looks like:

```
(make-instance class:case :name case2
   number-key 2
   title "General ChaseDirect inquiry; Auto Respond."
   subject "chase direct"
   description "Please send information on managing an account
         from my PC .  My address is: "
   actions (create$ action:auto:chase-direct-std)
   address? "Yes"
)
```

There are a couple of general tips to follow when adding cases:

1. Do not include specific information, such as a customer's name, address, or phone number when translating an email message into a case. But do remember to include (and set to "Yes") the relevant **class:case** attributes that appear within the text. This is important in contributing to the casebase match score. For instance, if a customer's email contains his address, remove the actual address, but set the **address?** attribute to **"Yes"**. The following example (Table 4:

[9] We could have used the **define-instance** form, but this would have created a permanent entry of each case in the system, whereas EZ Reader dynamically loads the current casebase upon start-up.

Translating an email message into a caseTable 4) shows an original email message and its corresponding case in the casebase:

Orlginal email message	Corresponding case in casebase
Date: 19-SEP-1995 21.44:40.97 MAIL From: IN%"72053.2204@compuserve.com" "Thomas Hughes" To: IN%"ChaseOnLine@delphi.com" "ChaseDirect" CC· Subj: New Account Return-path: <72053.2204@compuserve.com> Received: from arl-img-6.compuserve.com ("port 4756"@arl-img-6.compuserve.com) by delphi.com (PMDF V5.0-5 #10880) id <01HVH4DYFVO08WW0P8@delphi.com> for chaseonline@delphi.com; Tue. 19 Sep 1995 21:44:40 -0400 (EDT) Received: by arl-img-6.compuserve.com (8.6.10/5.950515) id VAA27915; Tue. 19 Sep 1995 21:42:45 -0400 Date: Tue, 19 Sep 1995 21:37:30 -0400 (EDT) From: Thomas Hughes <72053.2204@compuserve.com> Subject: New Account To: ChaseDin'ct <ChaseOnLine@delphi.com> Message-Id: ·950920013730_72053.2204_GHB35-1@CompuServe.COM> Content-transfer-encoding: 7BIT Message Text: Dear Sirs. I recently saw your advertisement in The New York Times. I am Interested in learning whether or not your system would work for me. Currently I am using Quicken and Checkfree which is linked to a NOW account at a savings bank. For the most part, everything works pretty well. I especially like the fact that I can have Quicken send my payments to Checkfree without having to use any other software. Checkfree itself, however, could certainly use some improvement. I wouldn't mind losing the $9.95/month fee. either. I was debating switching to Citibank's system but they have their own software which I find too unwieldy. Please E-Mail me back and let me know the details of your system. The fact that you are giving Quicken away gave me hope that I might be able to use it with your system. Thanks. Tom Hughes 1854 Ranch Drive Syracuse. NY 13057 (505)284-1982 ** End of message **	(make-instance class:case :name case999 number-key 999 title "Lengthy inquiry from 72053.2204@compuserve.com: refer." subject "New Account" description " Dear Sirs. I recently saw your advertisement in The New York Times. I am interested in learning whether or not your system would work for me. Currently I am using Quicken and Checkfree which is linked to a NOW account at a savings bank. For the most part, everything works pretty well. I especially like the fact that I can have Quicken send my payments to Checkfree without having to use any other software. Checkfree itself. however. could certainly use some improvement. I wouldn't mind losing the $9.95/month fee. either. I was debating switching to Citibank's system but they have their own software which I find too unwieldy. Please E-Mail me back and let me know the details of your system. The fact that you are giving Quicken away gave me hope that I might be able to use it with your system. Thanks. " actions (create$ action:refer.sales:priority:3) address? "Yes" lengthy? "Yes")

Table 4: Translating an email message into a case

Also note in this example that the customer included his phone number, which EZ Reader will detect from the **q:phone?** rule. But because the **phone?** attribute is merely there for detected purposes, you would not want it to contribute to the match score. Therefore, it should not be included in the case.

2. Very lengthy email messages may not be appropriate candidates for new cases. The reason is because longer texts tend to contain more words that raise the casebase match score, whether the desired action is correct or not. It is best to use cases (or edit the text) that contain only the phrases that are pertinent to the corresponding actions.

When should you add cases to the casebase? The current casebase is appropriate for obtaining a high level of accuracy with respect to the types of email messages that are being sent in. If there is a new promotion, or you notice that many people are asking for information that is not currently addressed in the casebase, then it is recommended that cases be added to the casebase to handle these emails.

Modification of existing cases is only required if business requirements deem it necessary. If a new type of action is added to the actionbase, it may be wise to review the casebase to see if any of the cases would provide a more accurate response with respect to this new action. Another appropriate time for modifications is when a question rule in the rulebase is added, modified, or deleted. For example, suppose a new attribute, **current-customer?**, is added to **class:case** which has a corresponding rule that fires whenever a current customer is detected and incorporates that attribute (e.g. **q:current-customer?**) into the match scoring. It will then be necessary to update any cases in the casebase that contain references to current customers by adding the attribute-value pair **current-customer? "Yes"**.

It is generally not advised to delete existing cases from the casebase. In order to get good match scores (which translates into more accurate EZ Reader recommendations), the casebase should contain *at least* 200 cases. If there is a case that appears to get the highest match score often, but the response is incorrect for the emails being matched (but is correct for the case itself), this case is a good candidate for deletion. Most likely, this is a rather lengthy text that contains several common Chase/banking related phrases and keywords.

Action maintenance

The actionbase, which can be found in the text file **ez-acts.art**, is similar in structure to the casebase; it contains instances of actions belonging to the class **class:action** which consists of the following attributes:

```
(define-class class:action object
  (action:type)        ;; "A" for automatic response, "R" for referral, or
                       ;; "D" for detected (e.g., informational)
  (action:title)
  (action:priority)    ;; only applies to referrals (values 1 (highest) to 4)
  (action:category)  ;; "AUTO", "SALES", or "SERVICE"
  (action:text)        ;; replaced by "canned.nsf" Notes database of responses
  ;; following attributes not used, but could be for auditing purposes
  (date-of-creation)
  (time-of-creation)
  (date-last-modified)
  (time-last-modified)
  (author)
)
```

A case definition then takes on the following form:

```
(make-ir-stance class:action :name action name
   action:type {"A", "D", or "R"}
   action:title "title of action"
   action:priority {1, 2, 3, or 4}
   action:category {"AUTO", "SALES", or "SERVICE"}
   action:text "text of action (if any)"
)
```

The definition for the ChaseDirect standard automatic response action is:

```
(make-instance class:action :name action:auto:chase-direct-std   ←
   action:type "A"
   action:title "Automatic Response:  ChaseDirect Info Standard."
   action:text "Welcome to ChaseDirect.Thank you for your
                  interest in ChaseDirect . . . rest of canned response . . ."
   action:category "AUTO"
)
```

For each case in the casebase, all of its specified actions must be an action found in the actionbase and in the Notes repository (database) of canned responses; otherwise, an invalid response will appear if that case is selected by EZ Reader at runtime. The reverse does not hold true, though. You can create an action that never gets used. You may want to do this if you know future cases will use that action. The example below shows exactly where to specify an action name:

```
(make-instance class:case :name case999
   number-key 999
   title "ChaseDirect and merger inquiry; auto respond."
   subject "ChaseDirect"
```

CONFIDENTIAL

```
    description "I am interested in ChaseDirect.  Please send
                me more information about banking online.  I
                am also concerned about the merger between
                Chase and Chemical Banks.  How will this
                affect your products and services?"
    actions (create$ action:auto:merger action;auto:chase-direct-std)
    merger? "Yes"
  )
```

To insure the accuracy of action names referenced in a case, it may be helpful to mark&copy the action name from the actionbase, then paste it into the case, thereby reducing typos.

Additions, modifications, and deletions of actions are necessary whenever business changes require them. For instance, at one time, emails referencing investments were marked as a detected action only, then later, ChaseDirect created an automatic response for that type of email. In that case, the **action:detect:investment** was changed to **action:auto:investment** with appropriate action:type, action:title, and action:text changes. Remember that modifications can simply be viewed as an addition plus a deletion.

Action Categories and Priorities

There are three general types of actions EZ Reader recommends:

1. A=Automatic: no manual review necessary
2. R=Referral: needs manual review
3. D=Detected: EZ Reader found some information in the email currently being processed which matches a pre-specified keyword, phrase, or numbering scheme.

Furthermore, referred emails are classified into one of 2 categories with one of 4 priorities. These are shown in Table 5 below:

Table 5: Referral categories and priorities

	Category:	Category:
Priority	**Service**	**Sales**
1	Fraud/Lost/Stolen Cards	MSNet (current promotion)
2	Sensitive info. do not respond	Fulfillment kits
3	Misc. service (specific person referenced)	Misc. sales (foreign, multiple questions, lengthy)
4	Comments/FYI	None

In the actionbase, the general naming convention for referred actions is:

action:refer:CATEGORY{SERVICE,SALES}:PRIORITY:{1,2,3,4}

For example, the name of the action corresponding to lost ATM or credit cards is **action:refer:service:priority:1**. This categorization scheme will aid CSRs in quickly coming to an appropriate response for the current email.

The current set of actions is listed in Table 6 below:

Table 6: Action Codes

Action Code	Description
ACTION:AUTO:CHASE-DIRECT-MAC	standard brochure for Macintosh customers
ACTION:AUTO:CHASE-DIRECT-STD	standard brochure
ACTION:AUTO:INVESTMENT	information about investments
ACTION:AUTO:MERGER	information about the Chase/Chemical merger
ACTION:AUTO:SENSITIVE-WARNING	warning message for emails containing sensitive information
ACTION:AUTO:SIGN-UP-ACK	acknowledgement message for sign-up-kit requests
ACTION:DETECT:ADDRESS	a mailing address was included in the email
ACTION:DETECT:DO-NOT-CALL-CUSTOMER	customer requests NOT to be called by phone
ACTION:DETECT:FAX	a fax number was included in the email
ACTION:DETECT:FOREIGN	foreign customer
ACTION:DETECT:INVESTMENT	investment inquiry in email
ACTION:DETECT:MS-MONEY	email contains a reference to MS Money
ACTION:DETECT:MS-NETWORK	email came from MSNET
ACTION:DETECT:PERSON-REFERENCED	a specific ChaseDirect employee was referenced
ACTION:DETECT:PHONE	a phone number was included in the email
ACTION:DETECT:QUICKEN	email contains a reference to Quicken
ACTION:DETECT:SENSITIVE	email contains sensitive information
ACTION:DETECT:WIN95	email references Win95
ACTION:NONE	no response necessary
ACTION:REFER:PRIORITY:4	comments/thank yous
ACTION:REFER:SALES:PRIORITY:1	MSNet email forms
ACTION:REFER:SALES:PRIORITY:2	Fulfillment-kits
ACTION:REFER:SALES:PRIORITY:3	misc. sales (foreign, multiple questions, lengthy or no good recommendation)
ACTION:REFER:SERVICE:PRIORITY:1	lost/stolen ATM/credit card
ACTION:REFER:SERVICE:PRIORITY:2	email contains sensitive information
ACTION:REFER:SERVICE:PRIORITY:3	misc. service (specific person referenced, problem that needs resolution)
REFER:CANCEL-ACK	acknowledgement message for a request for cancellation of bank services
REFER:FOLLOW-UP-ACK	acknowledgement message telling sender that the email has been forwarded for further review

REFER:PAY-ON-LINE	customer has a question/problem with Pay Online service
REFER:QUICKEN-FOR-MAC	customer inquires about Quicken for Macintosh

It is important to remember that actions are also housed in the Notes database called EZ Reader Canned Responses, which can be found on the same server as EZ Reader (current MCB-NOT-01). The set of actions in **ez-acts.art** must be the same as the set of actions found in the canned responses database.

Note that the text of a response can be modified by the business users in Rochester; it is wording only and does not affect EZ Reader. On the other hand, if they require a new action or the deletion of an existing one, then the KB team is responsible for making these updates to the Notes database.

VIM/Notes API maintenance

Incoming email messages are routed from the Internet domain (e.g., chasedirect@delphi.com) to a Lotus Notes inbox (named *Chase Direct-In*), from which EZ Reader continuously retrieves unread messages one at a time. The ability for one Windows™ program to pass data to and from another is accomplished via an *application programming interface* (API). EZ Reader uses the *vendor independent messaging* (VIM) API provided by Lotus Notes as well as Notes' own API.

The EZ Reader interface to Lotus Notes is accomplished using the VIM API[10] as well as Notes' own API. The Vendor Independent Messaging interface is an industry-standard application programming interface that provides the messaging (i.e., sending and retrieving mail) functionality provided by Notes or any full-featured messaging system. VIM allows programmers to develop mail-enabled applications that are portable across various email systems.

The EZ Reader-to-Notes interface is modular in design. Four separate executable programs (written in C language) were created[11]: **filemsg, markmsg, sendmsg**[12], and **writemsg**. Filemsg

[10] Please refer to the EZ Reader Glossary in Appendix B - EZ Reader Glossary for further details.

[11] Ideally, a clean interface via DLL calls is desired, but because of current limitations for simulated-32-bit ART*Enterprise® and 16-bit VIM.DLL, there were enough complications to warrant a work-around solution involving

is responsible for retrieving emails from Lotus Notes, and writing them to an external text file (filemsg.out), from which EZ Reader reads them. After EZ Reader has processed the message, the writemsg module is called upon to perform the task of writing the outgoing message from another external text file (sendmsg.out) to the ChaseDirect inbox. It also writes the outgoing message to the EZ Reader archive database. Finally, markmsg is called when the email message has been successfully processed: it tags the message in the Notes inbox to "*read*".

Once EZ Reader has interpreted an email, it must somehow either reply back to the sender or forward it to ChaseDirect for manual review. Instead of performing a reply or forwarding mail-send command as provided by VIM, we opted for directly writing to the ChaseDirect inbox (a Notes database separate from the EZ Reader inbox). Because VIM does not provide functions for database access, we used the API bundled with Lotus Notes. The following sections list the VIM and Notes API functions used by EZ Reader.

STATMSG.OUT file

In addition to the 2 external text files used for message passing, the API modules also uses another text file, called statmsg.out. This file contains status and/or error messages generated at runtime. EZ Reader uses this file to monitor its connection to Lotus Notes. The most common messages are "NO MESSAGES" and "NO ERROR". If there is an error in any VIM or Notes API module, a message describing that error will be written to this file. Once the API module has finished running, EZ Reader will read from the statmsg.out file, and proceed according to the contents. For instance, EZ Reader calls filemsg.exe to retrieve the next email. For some reason, it cannot log onto a Lotus Notes Session. filemsg.exe will write the error message, e.g., "Error: cannot open VIM session", to statmsg.out and end its execution. EZ Reader will then read from statmsg.out and, discovering an error, will self-abort. As another example, suppose the filemsg module finds no new messages. It will write the message "NO

separate C programs and external text files. Details of this work-around are explained later in this section.

[12] Note that sendmsg is not currently used by EZ Reader; it performs an actual mail send, which is instead accomplished with a direct write to the database. We mention it here and leave the source code accessible, in case future phases require a mail send rather than a database write.

MESSAGES" to the statmsg file; EZ Reader will read the message, understanding that it should now pause a few minutes before trying to retrieve mail again.

VIM Functions

Table 7 thru Table 10 list the complete set of VIM functions employed by EZ Reader (by functional grouping), which is in actuality a subset of the entire group of functions defined by the VIM specification. The C modules (including the unused sendmsg) that call the functions are also listed.

Table 7: VIM Session Management Functions

VIM API module	Function	Description
filemsg, markmsg, sendmsg	VIMInitialize	Initialize VIM for EZ Reader processing
sendmsg	VIMQueryCapability	Return implementation features and limits
filemsg, markmsg, sendmsg	VIMGetDefaultSessionInfo	Return default values required for session login
filemsg, markmsg, sendmsg	VIMOpenSession	Open and log onto a session with VIM
filemsg, markmsg, sendmsg	VIMCloseSession	Close the session and free any associated resources
filemsg, markmsg, sendmsg	VIMTerminate	Terminate VIM for the current process
filemsg, markmsg, sendmsg	VIMStatusText	Return text associated with a status code

Table 8: VIM Message Container Functions

VIM API module	Function	Description
filemsg, markmsg	VIMOpenMessageContainer	Open a specified message container
filemsg, markmsg	VIMEnumerateMessages	Enumerate the messages in a message container
filemsg, markmsg	VIMCloseMessageContainer	Close the specified message container

Table 9: VIM Message Access and Attribute Functions

VIM API module	Functions	Description
filemsg, markmsg	VIMOpenMessage	Open a message for further access
markmsg	VIMMarkMessageAsRead	Mark message as read and send the return receipt if appropriate
filemsg	VIMGetMessageHeader	Get one or more message header attributes
filemsg	VIMEnumerateMessageItems	Get one or more multi-valued message header attributes
filemsg	VIMGetMessageItem	Read an item into a buffer or file.
filemsg	VIMEnumerateMessageRecipients	Enumerate the recipients of a message
filemsg, markmsg, sendmsg	VIMCloseMessage	Close the specified message

Table 10: VIM Message Creation and Submission Functions (alternative to writing to the Notes database)

VIM API module	Function	Description
sendmsg	VIMCreateMessage	Create a new message
sendmsg	VIMSetMessageHeader	Set the value of a message header attribute
sendmsg	VIMSetMessageRecipient	Add a recipient (to, cc, bcc) to a message
sendmsg	VIMSetMessageItem	Add any class of item to a message
sendmsg	VIMSendMessage	Submit a message for delivery

The path EZ Reader must take in order to retrieve a piece of email is outlined in Table 11 below (VIM functions called by the FILEMSG module). filemsg.exe will retrieve the first new message in the Notes inbox (if any). If a new message is retrieved, it will be written to the filemsg.out text file and the line "NO MESSAGES" will be written to the statmsg.out status/error file.

Table 11: FILEMSG functional flow

Called at the start:
VIMInitialize
VIMOpenSession
VIMOpenMessageContainer
Called to retrieve a new piece of email:
VIMEnumerateMessages
VIMOpenMessage
VIMGetMessageHeader
VIMEnumerateMessageRecipients
VIMEnumerateMessageItems
VIMGetMessageItem
VIMCloseMessage
Called at the end:
VIMCloseMessageContainer
VIMCloseSession
VIMTerminate

Once a single email message has been processed fully, i.e., retrieved by FILEMSG, interpreted by the EZ Reader knowledgebase portion, then written to the manual review database by the Notes API module WRITEMSG, the VIM module MARKMSG is called to mark the email as "read" within Lotus Notes. By performing this step last, the chance of lost or unresponded to emails is eliminated; i.e., if EZ Reader. For example, suppose EZ Reader just retrieved a new email message. Suddenly, the LAN connection is lost and EZ Reader aborts. When EZ Reader is switched back online, it will restart from the beginning. Although that piece of email was retrieved already, it wasn't marked as read by markmsg.exe, so we know that it has not yet been responded to. The filemsg program will retrieve that same piece of email as if it hadn't been retrieved yet.

Table 12 shows the functional flow of the VIM functions used by **MARKMSG**:

Table 12: MARKMSG functional flow

Called at the start:
VIMInitialize
VIMOpenSession
VIMOpenMessageContainer
Called to mark the current email as read:
VIMEnumerateMessages
VIMOpenMessage
VIMMarkMessageAsRead
VIMCloseMessage
Called at the end:
VIMCloseMessageContainer
VIMCloseSession
VIMTerminate

Lotus Notes API functions

As stated previously, VIM does not provide functionality for accessing Notes databases. For this reason, we must incorporate functions resident to Lotus Notes to accomplish the task of writing to a Notes database. In addition, the same functions are used for posting each email message to the archive database. Table 13 lists the Notes API functions used by EZ Reader. All Notes API functions are contained within the C module **WRITEMSG**.

Table 13: Notes API Mail and Database Access Functions

Function	Description
NotesInit	Initializes the Notes run time system
NSFDbOpen	Open a specific Notes database
NSFDbInfoGet	Returns the title and categories of the current database
NSFDbInfoParse	Get a specified piece of information from the database information buffer
NSFNoteCreate	Create a new data note
NSFItemSetText	Write a specific field to the note
NSFNoteUpdate	Takes a note handle as input, and updates the .NSF file on disk with all the fields

	written using that handle
NSFNoteClose	Close the specified data note
NSFDbClose	Close the specified database
NotesTerm	Terminates the Notes run time system

Table 14 shows the functional flow of the WRITEMSG Notes API module, which is called to write the original email, appended with EZ Reader's recommendations, to another Lotus Notes database (i.e., ChaseDirect's mailbox of emails requiring manual review). After EZ Reader has interpreted the current email, it will write the original message, along with its recommended actions, to the external text file called sendmsg.out. writemsg.exe reads the contents of sendmsg.out, then populates the fields of a database record, and adds the record to the appropriate Notes database.

Table 14: WRITEMSG 'unctional flow

Called at the start:
NotesInit
Called to respond to a piece of email:
NSFDbOpen
NSFDbInfoGet
NSFDbInfoParse
NSFNoteCreate
NSFItemSetText
NSFNoteUpdate
NSFNoteClose
NSFDbClose
Called at the end:
NotesTerm

Maintenance of the EZ Reader/Lotus Notes API should be minimal. It is not advised to arbitrarily modify this portion of the EZ Reader code. The functions are dependent upon the **VIM.DLL** file (Dynamic Link Library containing all of the VIM functions) and must follow the specifications provided by Lotus Notes. If the current version of Notes is upgraded, then it is wise to retest each of the VIM functions for compatibility.

EZ Reader Deployment

Deployment of source code is the process of transforming an application into a binary *image file* for distribution to users. Deployed programs load much faster than programs loaded through the ART**Enterprise*® Repository. In addition, by deploying an image, installation of full-blown ART**Enterprise*® onto a production machine becomes unnecessary[13].

EZ Reader source code resides in the following directory on both the development and production machines:

```
c:\ae20b\ezreader\
```

If modifications are made to the ART**Enterprise*®/EZ Reader code, it is necessary to re-deploy the image. A deployment environment customized for EZ Reader has been created on the development machine. Simply double-click on the **EZ Reader Deployment Environment** icon in the **EZ Reader Development** group in Windows Program Manager. A screen will appear, displaying many files being loaded into memory. Once the loading is done, the ⇒ prompt will appear at the bottom portion of the window. Now type

```
(deploy-ezreader)
```

which is a function that initializes some things, then calls the function to deploy the application. The result of successful deployment is the file **ezreader.ae** which can be found in the EZ Reader directory. This file must then be uploaded to the production machine into the EZ Reader directory of the same path.

Not all modifications require re-deployment of EZ Reader. If you are simply modifying or adding to either the casebase **ez-cases.art** or the actionbase **ez-acts.art**, then simply copy the updated file(s) from the development machine to the production machine. This is allowed because the actions and cases are loaded to the application at runtime and are not permanent objects within EZ Reader. So any updates made to either the actionbase or the casebase will be reflected the next time EZ Reader is run.

[13] Please refer to the ART**Enterprise*® Studio Guide, Chapter 5, for full details on how to deploy an application.

Change Control Procedures

Change control procedures follow the standard Chase protocol as outlined in the following steps:

1. The business users, i.e., ChaseDirect, submit a change request form for a modification to EZ Reader. Types of changes requiring form submission include correction of errors in EZ Reader's response and the addition or deletion of canned responses (a more detailed list of these can be at the beginning of the Maintenance section). Note: if a rate change is needed in any of the canned responses, ChaseDirect will be able to make those modifications themselves. They will have field modification access (Notes Group has Designer Access) to the EZ Reader Canned Responses Notes database residing on the server in Rochester (currently NYUROCHDRCTBNK).
   a) If the modification is an error in EZ Reader's response to a particular email message, then the entire message (including EZ Reader's recommendations) should be attached to the change request form.
   b) If the request is for a new response or new referral category/priority, then this should be specified in detail on the change request form.
2. This change request form (paper) is then routed to the KB Team, KB Team *will not* have access to the production database, who will
   a) implement the modification,
   b) copy the modified files to diskette (ez-cases.art, ez-acts.art, and/or the redeployed image ezreader.ae), and
   c) forward the diskette / template and change request form (with the modification date, person who implemented, etc. filled out on the form) to the Notes group. Update will be delivered via self extracting executible file. Binary file must be zipped due to size of file -example 3 Megs
3. The designated person in the Notes group will
   a) make a backup of the current EZ Reader application (EZ Reader is backed up on a daily basis)
   b) upload the modifications (file copy from the diskette to the EZ Reader subdirectory on the production machine's hard drive),

c) complete and file the change request form, and finally,

d) notify the parties involved that the modification has been fully completed.

ChaseDirectIn	(MCBNOT01)
Server Name	NYUROCHDRCTBNK

Troubleshooting Guide

Table 15 below lists a set of common EZ Reader/ ART*Enterprise® problems with recommended solutions.

Table 15: Troubleshooting Tips

Symptom	Diagnosis
Start EZ Reader with **(run)** but nothing happens	Type **(reset)** first to clear the rule agenda and perform other ART**Enterprise*® housekeeping tasks.
ART**Enterprise*® Command Interpreter window doesn't show all of the output	Because there is a limit to the amount of data the Command Interpreter buffer can hold, use the **(dribble)** command before running the application.
Insufficient memory or system lock up[14] during execution (and you have to reboot the machine)	Perform a **chkdsk /f** command from a DOS prompt. This is also recommended if you are getting many GPFs (Windows General Protection Fault error) during maintenance.

[14] System lock-up happens most frequently when you are working with APIs.

Appendices

Appendix A - Error/Status Messages

This section contains a guide to EZ Reader error messages. Error messages appear in ART**Enterprise*®'s Command Interpreter window, as well as in the **statmsg.out** external text file during API runtime.

For ART**Enterprise*® errors, please refer to the online or hardcopy documentation.

API error/status messages:

- "NO ERROR" : no error occurred, EZ Reader can proceed as usual.
- "NO MESSAGES" : generated during the **filemsg.exe** API module if there are no new messages in the Lotus Notes inbox.
- Any other VIM/Notes API errors : **statmsg.out** will contain the error number as well as a brief (one-line) description of the error. From there, you can refer to the VIM and Notes documentation.

Appendix B - EZ Reader Glossary

actionbase: The actionbase of EZ Reader is a set of ART* *Enterprise*® objects which are related to the casebase. It contains the set of all possible actions that EZ Reader can recommend for a given email. The casebase search function looks up the recommended action(s) in the actionbase to get the details about the action. The actionbase and casebase are two separate .art files, for modularity and ease of maintenance.

API: The Lotus Notes Application Programming Interface is a programming interface to mail functions. The functions and structures of the Notes API provide complete access to all types of Notes documents and databases. Since programmatic access to Notes mail is based entirely on Notes documents and Notes databases, the Notes API provides all the functionality necessary to send and receive messages in any context and to manipulate the messages within a Notes database.

casebase: A casebase is a database of information associated with specific conditions or situations which can be searched according to specific feature criteria, and retrieving those cases which meet the specified criteria. One of the main benefits of case-based reasoning (**CBR**) technology is that the cases retrieved from the casebase need not match the criteria exactly, which results in more powerful retrievals than a typical database. EZ Reader's casebase consists of hundreds of actual email messages and the reply most suited to the case. Each email message thus represents a case.

CSR: Chase Manhattan Bank's Customer Service Representative

DLL (**Dynamic Link Library**): a set of pre-compiled functions that reside in one location (a library), from which other external applications or functions may call any of the functions.

KB (knowledge-based) systems: specially designed software that uses *rules of thumb* to interpret data. These systems reflect the knowledge of professionals and enable the automation of operational decisions, display expert advice to a business transaction, etc. EZ Reader is a KB application.

message container: In Lotus Notes parlance, a message container is a database (repository) of email messages. In other words, a mailbox.

rule: A rule represents operational knowledge; i.e., rules do things -- they carry out actions. A rule is simply a *whenever-do* type of statement

rulebase: Also called a rule set, this is a collection of related rules that specialize in one aspect of an application. More generally, a rulebase simply a set of rules.

VIM: The Vendor Independent Messaging interface is an industry standard Application Programming Interface supported by Lotus Notes, Lotus ccMail, and other messaging systems which support mail-enabled program development (e.g., message send and receive) across a variety of operating platforms, including Windows 3.1 and NT, SunOS, etc. VIM is does not contain all the functionality of the Notes API.

Appendix C - Future Releases of EZ Reader

Enhancements are continuously being completed and the list varies depending on Chase direction. The current list of enhancements and fixes can be obtained at any time by consulting with the EZ Reader business sponsor.

Future releases of EZ Reader will:

- automate the CMS update process for emails for which automatic answers are selected by EZ Reader.
- automate the business maintenance functions of the application, i.e., enable ChaseDirect business users to change how EZ Reader interprets messages and links them to responses.
- enable EZ Reader throughput and output statistics to be accessed in a flexible and effective way for reporting purposes.

After quality is initially verified, we envision sending automatic responses directly to customers without human intervention. In order to deliver messages directly back to customer, we need to do a technical assessment of whether the delivery should take place directly from EZ Reader or be done through Lotus Notes.

Depending upon Chase business needs and requirements, EZ Reader can also:

- allow for incoming messages from other, related, Internet sources, such as the World Wide Web.
- enable EZ Reader output to be addressed directly to different Chase recipients through Lotus Notes.
- enable use of historical email and profile data to personalize EZ Reader processing.
- provide the ability to automatically process incoming messages in Spanish or other languages, providing a distinct marketing advantage.
- process incoming email for other areas of Chase.

Other future phases to be considered include incorporating machine learning, adding GUI screens for maintainence of the knowledge base (cases and actions, Notes DB of canned responses, etc.). Knowledge validation, etc.

Reusability

The framework and infrastructure that were developed for EZ Reader for ChaseDirect are an investment that can be capitalized on further. Much of the conceptual design and code may be reused for a similar system, albeit with different business rules, for such areas as the Personal Banking Center.

Appendix D - Standards and Miscellaneous

EZ Reader will follow existing production application standards as outlined below.

Administration

Change Management

After EZ Reader goes into production, requests for changes and responses to changes must be made through the change control procedures specified by Chase Technology Services. This specifies a routing protocol from ChaseDirect to the KB Team as a way to track and manage EZ Reader modifications.

EZ Reader Startup/Shutdown Procedures

EZ Reader depends on the simultaneous operation of several interrelated application systems. each with its own startup procedures.

To pause EZ Reader during processing, press the **Break** button at the top of the Command Interpreter status window. At this point, you can either continue with processing, or abort EZ Reader altogether. Use the latter when installing upgrades.

To continue processing after breaking, type :c in the bottom of the window.

To abort EZ Reader, type **(halt)** to stop the rule-base engine from firing, then type **(ezr:abort)** which is a function that ends EZ Reader processing and performs clean-up routines.

Backup/Recovery Procedures

All automatic emails processed by EZ Reader will be archived in the EZ Reader Archive Notes database. Anyone with access may review or search for emails. For referred emails, the Notes interface includes a macro that automatically archives the message appended with ChaseDirect's response (as well as who responded to it) to the archive database, after the ChaseDirect CSR sends the reply back to the sender.

The EZ Reader client machine is connected to the LAN at 1985 Marcus Avenue (MCB-1985-02 server). The LAN group performs scheduled back-ups of files residing on the LAN, including the Notes databases used by EZ Reader. Recovery of files can be requested from the LAN group as necessary.

The chance for lost emails or emails not responded to is eliminated by the placement of the markmsg.exe within the processing cycle. We want to leave the current email as unread until we know for sure that it has been responded to. That way, in case EZ Reader aborts before it has a chance to send a reply or to write to the manual review database, the next time EZ Reader is started, it will re-retrieve the current (previously unanswered) email because it wasn't marked as read.

Additional Resources

References and Resources

For further reference on any of the topics discussed in this manual, check the following resources:

- ART**Enterprise*® online documentation called the Function and Class Reference Library (or FACLR), especially the chapters on rules and case-based reasoning.
- ART*Enterprise® written documentation
- Lotus Notes Toolkit Collection CDRom
- Microsoft Visual C/C++ Development System and Tools CDRom
- The technical paper on EZ Reader: J. Hsu, A. Rice, A. Angotti, and R. Piccolo. *EZ Reader: Embedded AI for Automatic Electronic Mail Interpretation and Routing*. Submitted to IAAI '96.
- Other miscellaneous documentation (functional definition, cost-benefit spreadsheet, etc.) in the EZ Reader \DOC subdirectory.

Because EZ Reader is a component architecture, there are several groups of Chase people responsible for different portions. If problems with EZ Reader arise, please contact the following people (listed by responsibility) in Table 16:

Table 16: Contacts and Support Numbers

Responsibility	People	Telephone	Location
ChaseDirect	Brenda Doben, Brady Masterson	716.258.6174	ChaseDirect (Rochester)
Business functions/liaison	Rosanna Piccolo	212-252-5909	Chase Emerging Delivery
ART*Enterprise®	Julie Hsu, Amy Rice, Brightware Technical Support	800-416-7332 x 579, 800-416-7332 x 589, 800-572-3675	Brightware
Internet to Notes connection	Mike Mazza, Mary Flannery, Lisun Joao	718.242.2420, 718,242.6661	MetroTech
Internal Notes connection (and other Lotus Notes tech support)	Angel Vazquez, Bill Morse, John Courier (Notes API)	516-574-7664 718-242-8555 718-242-3582	1985 Marcus Ave., MetroTech, MetroTech