MS EMAIL AGENT 1.0

J. VIKRAMADITYA REDDY

Department of Computer Science & Engineering
VNR Vignana Jyothi Institute of Engineering & Technology
(Affiliated to J.N.T. University)
Hyderabad - 72

2005

MS Email Agent 1.0

A Project Report submitted in partial Fulfillment of the Requirements for The Award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & INFORMATION TECHNOLOGY

BY

Mr.J.VIKRAMADITYA REDDY (01071A1289)

Under The Guidance of

Mr.V.S.N.Raju

VALLURUPALLI NAGESHWARA RAO

VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

(affiliated to JNTU)

BACHUPALLY, HYDERABAD

TEL: 23042758, 23042759, 23042760 FAX: 040-23042761.

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

(affiliated to JNTU)

CERTIFICATE

This is to certify that the Project entitled "MS Email Agent 1.0" has been submitted by Mr.J.Vikramaditya Reddy (01071A1289) of IV/IV CSIT, VNR VJIET, in Computer Science Department of VNR Vignana Jyothi Institute of Engineering and Technology affiliated to Jawaharlal Nehru Technological University.

The results embodied in this project have not been submitted to any other university for the award of any degree or diploma.

1/4/2005

Mr.V.S.N.Raju
Internal Guide
Assistant Professor,
VNR VJIET,
Hyderabad.

Dr.K.R.M.Rao
Head of Department
Department of C.S.
VNR VJIET,
Hyderabad.

Microsoft India (R&D) Pvt. Ltd.
Block B, 4th Floor, Cyber Gateway
Hi-Tec City, Hyderabad - 500 081

Tel 91 - 40 - 5566 2000
Fax 91 - 40 - 5536 1910
http://www.microsoft.com/india

Microsoft

Date: 28th March 2005

CERTIFICATE

This is to certify that the Project titled "MS Email Agent 1.0" has been completed by Mr. Vikramaditya Reddy J in partial fulfillment for the requirements for the award of his B.Tech Degree. This report embodies the work done towards successful completion of the project.

Vikramaditya Reddy J has been employed by Microsoft India (R&D) Pvt. Ltd, as Intern under project guidance of Mr.Abhishek Mathur (Development Manager – Microsoft India (R&D) Pvt. Ltd).

Abhishek Mathur
Development Manager,
Microsoft India (R&D) Pvt. Ltd.

Declaration

I here by declare that this project entitled "MS Email Agent 1.0" has been submitted to the Department of Computer Science, VNR Vignana Jyothi Institute of Engineering and Technology, affiliated to Jawaharlal Nehru Technological University in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Information Technology.

This Project has not been submitted by me to any other University/Institute for the award of any degree/diploma.

Mr.J.Vikramaditya Reddy,
01071A1289,
VNR VJIET.

Acknowledgement

I would like to take this opportunity to state that the task of completing the project work entitled " **MS Email Agent 1.0**" would have been impossible and never taken shape without the inspiring encouragement, generous help, guidance and co-operation extended implicitly and explicitly to me by some individuals.

First and foremost I would like to express deep sense of Gratitude to **Mr.Abhishek Mathur**, Development Manager, Microsoft India (R&D) Pvt. Ltd. for having permitted me to carry out this project work. I humble express my sincere heartfelt thanks to him for his constant guidance, valuable help, timely co-operation with immense dedication, encouragement and providing all facilities to complete the project successfully. I woe to all staff and hardware department executives of Microsoft Corporation for the success of the project work.

I am grateful to Professor **K.R.M.Rao**, Head of Department, Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, for all the help he has extended. His constant motivation and concern has been invaluable.

I wish to express our deep sense of gratitude to **Mr. V.S.N.Raju**, Assistant Professor, Department of Computer Science, our internal guide, for all his important suggestions and advice. He has bestowed a lot of his time and attention towards the completion of our project. My special thanks go out to Mrs.Kiranmayi.C, project-in charge, for their support. I am grateful to the entire staff of the department of computer science for their assistance.

Finally, I would like to thank my Family and friends for their moral support and encouragement.

PROJECT ABSTRACT

ABSTRACT

Title: **"MS Email Agent"**

Language: VB.NET

Platform: Microsoft .NET

Environment: Windows xp / Windows 2003.

This project entitled "MS Email Agent 1.0" is a CRM tool. The main features of the MS Email Agent are to process the inbound email messages and create email logs against cases in the clarify system. It is intended to deploy this tool in EMEA (Europe, Middle East and Africa) region and North America.

The main objective of this tool is to address the attachment handling requirement not available in Email Clerk. This tool will perform all activities performed by the existing component of Clarify called Email Clerk and in addition will also have multiple attachment handling capabilities.

It's developed as a service polling a specific mailbox using WebDav http request. It notifies customers after successful processing of Inbound Emails in a given format. On successful processing it creates logs against cases in the clarify system. Processes and Saves Inbound Email Attachments to file shares and is accessible through the Clarify System. It provides User Interface to configure Options as currently exists in Email Clerk.

CONTENTS

Index

INTRODUCTION

Introduction

The main features for the MS Email Agent are to process the inbound email messages and create email logs against cases / Sub Cases (as currently done by Email Clerk) and in addition also handle attachments contained in these emails. It is intended to deploy this tool in EMEA region and North America.

1.1 Problem Statement

The existing customer relation management tool named Clarify from AmDoc's Inc, has a component called the Email Clark which would handle the email based request and interactions with the inbound messages. But this component did not have any feature to handle the attachments. Thus this is the most important task performed by the CR representative to handle requests which are in the form of attachments. This feature was lagging.

In order to enhance the current scenario we have decided to come out with a tool the "MS Email Agent 1.0", which would do all the features which were present in the Email Clark and would also, be enabled to handle the feature of handling attachments.

1.2 Goal

The main objective of this tool is to address the attachment handling requirement not available in Email Clerk. This tool will perform all activities performed by Email Clerk and in addition will also have multiple attachment handling capabilities.

1.3 Organization Profile

Microsoft is a World Leader and pioneer in computer software technology in the world. Services IT is a premier wing of Microsoft which runs IT for Microsoft.

Microsoft's Mission: To enable people and businesses throughout the world to realize their full potential

Microsoft Corporation develops, manufactures, licenses and supports a wide range of software products for various computing devices. The Company's software products include scalable operating systems for servers, personal computers (PCs) and intelligent devices; server applications for client/server environments; information worker productivity applications; business solutions applications; software development tools, and mobile and embedded devices. Microsoft provides consulting services and product support services and trains and certifies system integrators and developers. The Company sells the Xbox video game console, along with games and peripherals. Its online businesses include the MSN subscription and the MSN network of Internet products and services. The Company's seven product segments are: Client, Server and Tools, Information Worker, Microsoft Business Solutions, MSN, Mobile and Embedded Devices and Home and Entertainment. Microsoft Corporation develops, manufactures, licenses & supports a range of software products, including scalable operating systems, server applications, worker productivity applications and software development tools. For the 6 months ended 12/31/04, revenues rose 9% to $20.01B. Net income rose 44% to $5.99B. Results reflect continued improvements in overall IT spending and lower research and development costs.

REQUIREMENT ANALYSIS

Requirement Analysis

In this age of cutthroat competition, companies in the IT Sector are always on the lookout to outdo the other. Only the businesses with the right processes exist; the rest perish. In the initial phases of IT, Design and Development were the much-emphasized areas. However, as the pervasive nature of software grew, the focus slowly shifted from the products to the consumers. It has been realized that providing customer satisfaction and earning the customers' goodwill are major factors that ensures the longevity of a product, and resultantly, the company. Thus, most of the service-oriented companies today, like Microsoft, have re-evaluated their priorities, with customer services being given due importance and significant budgets.

If all the operations of IT Company are grouped, this can be done under three major headings:

1. Design & Development
2. Testing
3. Support

Each of the above has a uniquely important role to play in the overall IT process. Developers analyze the requirements – they evaluate multiple methods of solving a problem and finally coding the specifications. This process is extensively tested with several test cases, and corresponding changes are made to the code, so that it becomes nearly fail-proof. The fully tested product is finally launched, and the interaction with the customer begins. So begins the process of Customer Support.

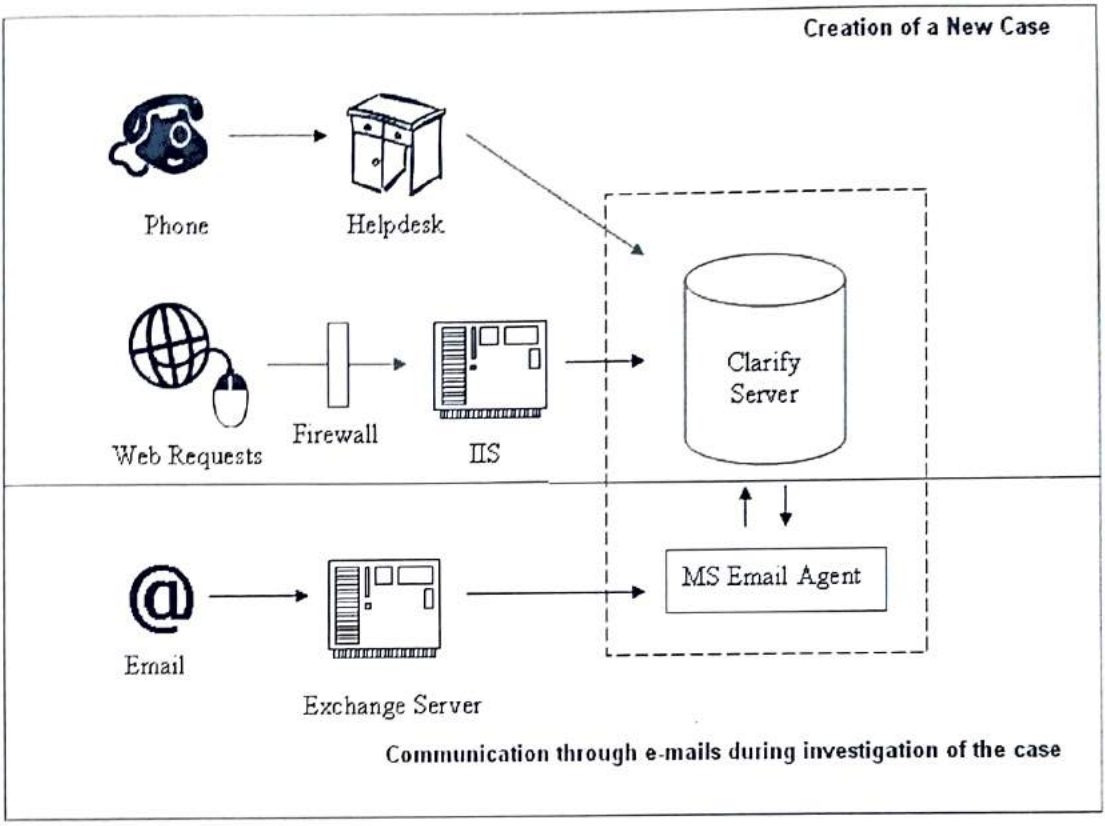
2.1 Support Organization at Microsoft

At Microsoft, the segment that deals and transacts with customers is called Services Operations and Technology. A further subdivision called Services IT is that part which provides information technology support for Microsoft's internal Product Support Services businesses. These internal businesses take care of the customers in the outside world. Primarily, there are three tiers of support, and this decentralization of support services is as follows:

- 1st Helpdesk: They take the phone calls from customers, and are the primary end points of contact with the customer
- Tier I: They take on the customer from the 1st Helpdesk, and deal with the issue of the customer. In the process, they use some CRM tools to track customer information, support entitlements and case histories. In short, record all the activities related to the customer. One such CRM tool is Clarify. Note that interaction with the end-user in the outside world ends here.
- 2nd Helpdesk: These personnel take on issues only from Tier I – i.e. they take on calls pertaining to Clarify and other related tools only.
- Tier II User Support: They troubleshoot basic server/network/connectivity issues; provide their Internal Customers with issue resolution and so on.
- Tier III Production Support: They deal primarily with server-side maintenance, along with performing operational tasks such as Data Base Administration and Infrastructure Planning

In this project, we are concerned only with the External Customers of the outside world, the 1st Helpdesk and Tier I Support.

2.2 Understanding the Process of Customer Support



The external customers are the end-users of Microsoft Products. Right from the time the Customer purchases a product, he/she may encounter some issues. These may be related to installation, product keys, contracts, usage issues and so on. The customer contacts the Technical Support of Microsoft – either through telephone or the Internet. The former shall be discussed, and the latter, called eSupport, is out of scope for this project.

The personnel at the first helpdesk take the customer's call and try to note the customer's issue. If they are entitled to solve it, they do so; or else, they escalate the issue to Tier I Support personnel. Product Support Service (PSS) Engineers form a major congregate at Tier I. Both the 1st Helpdesk and PSS Engineers use a CRM tool called Clarify.

2.3 Introducing: Clarify

Microsoft Corporation, over a period of time, has gained a huge customer base, with over three-fourths of the computer users around the globe using Microsoft Products. Thus, to manage and resolve customer issues is a formidable task. For this, specialized tools called CRM Tools are used.

Clarify is one such third-party CRM Tool, developed by Amdocs Corporation. It is a SQL Server based application used throughout Product Support Services for creating and tracking service requests (cases) submitted by Microsoft customers and for managing contracts. It consists of a client application that is installed locally on a workstation and requires connectivity to a central database.

Throughout the globe, there are several installations of Clarify – North America, United Kingdom, Japan, Hong Kong, Korea, Taiwan, China, Singapore, Latin America and Australia. Within the North America system alone, there are more than 3000 concurrent users of Clarify.

Clarify includes components to track customers, contracts, entitlements and customer support cases which have been submitted through a variety of means. Two common ways in which customers can submit a question to Microsoft are by telephone or by Web Response. Customers are entitled to support under Standard Warranty, Pay Per Incident (PPI) or by Contracts (Premier, Priority, Professional, Alliance). Clarify is used to validate the customer's entitlements and route the request to the appropriate support queue. Support Engineers accept cases from queues to identify the problem, research upon it, and finally, resolve it.

2.4 Emailing and Clarify:

Once the customer reports a problem, a Case is created against the customer's name. A corresponding Case ID is also generated, which is unique to each case created in Clarify. In all future correspondence with the customer, this Case ID is the unique reference number. All the interaction with the customer is recorded as part of the Case Information.

Sometimes, the support personnel may need some extra information from the customer to solve an issue. Thus, there is a need for future correspondence with the customer. The customer may need to send some important piece of information, which becomes crucial in resolving the issue. This is generally done through Email.

It may be noted that the purpose of Email within Clarify is different from the purpose of email in the outside world. In Clarify, only Inbound Emails are of primary importance, and need a different emailing system than a regular emailing system. The purpose is to

trap an inbound email, validate it, and append it to an appropriate case. We need a service that is constantly on the lookout for emails, and append them to appropriate cases.

Because an emailing service needs to be developed for the Clarify application, normal Web-based emailing principles are inapplicable here. Hence, a parallel cannot be drawn with respect to the functionality of this emailing system with commercial email applications like Yahoo or Hotmail.

2.5 The Existing System:

At the core of the Clarify system is the database, *Compass*. The other core components include:

- Email Clerk: It parses email messages from a Customer, and programmatically appends the message body to that customers existing Clarify Case
- Rule Manager: It triggers automated responses to certain programmed events or "rules", which have been defined by the PSS. In certain instances, it also prepares outbound email messages from Clarify/PSS Engineers to external customers.
- Mail Spooler: It is an internally developed tool that formats and sends email messages that have been prepared by the Rule Manager.

Normally, for Inbound Emails, Email Clerk was used. For outbound emails, Compass Rule Manager was used. Both of them create activity logs in the Clarify System.

Now, the presence of these services had several latent limitations, some of which are listed below:

- The presence of two services for emailing was cumbersome
- Attachments cannot be handled by Email Clerk
- Emails were not acknowledged by Email Clerk
- Emails were not secure

This earlier system, Email Clerk, was developed using C++. This system had no attachment-handling capability. Whenever the customer intends to send an email, he needs to send it to a common share. Then, they would have to be explicitly picked up and then attached, manually, to the corresponding cases. Now this is a very tedious job, considering that about half-a-million Clarify Cases are created per month. Also, because of the C++ based system, periodic memory refreshes are necessitated. The malloc() statement keeps allocating available RAM space, because of which RAM needs to be refreshed – this phenomenon is called Memory Leak.

2.6 Introducing: MS Email Agent

Keeping in view the inherent limitations, it was proposed that a new tool be developed to replace the functionality of Email Clerk. Hence, we have taken up this project, i.e. the development of MS Email Agent. MS Email Agent is to replace the current inbound email processing functionality of Email Clerk and enhance the inbound email processing by handling attachments.

One of the main objectives of MS Email Agent tool is to address the attachment handling requirement, which is not available with Email Clerk. This tool will perform all the activities performed by Email Clerk, and also, in addition, will have multiple attachment handling capabilities. This tool also processes digitally signed emails.

MS Email Agent is a Web-enabled application. It is different from normal emailing in the sense that the user does not have Direct Access/ Permissions to his/her mailbox on the mail server. Normally, in regular commercial email, an interface is provided which indirectly provides access to the mail box on the mail server.

But in this scenario, since there are hundreds and thousands of emails coming in, email monitoring and validation is required at the source end itself, and improper emails should not reach the destination. Hence, there is a need for a service that acts as a Check/Monitor to allow inbound emails – this is provided by MS Email Agent.

MS Email Agent shall be implemented as a VB .NET service on an appropriate Windows 2003 Server. It shall repeatedly poll the mailbox on the exchange server. It does not require that a MS Outlook is installed on the server. The client side outlook rules will not interfere in any manner with the email processing done by MS Email Agent.

For handling attachments, MS Email Agent shall obtain the file/attachment, save a copy of the attachment onto a local server and then upload it to the Clarify File Share. Eventually, the local copy is deleted. It may be noted that the Clarify File Share is a common file share where all the attachments reside. A link is created from the log of the case to access the particular attachment in the File Share.

Every Clarify case is assigned a unique 12 character Case ID beginning with letters SRX/SRQ/SRZ and so on. If it is stipulated that the Header/Subject/Body of the email must mention this Case – ID, then MS Email Agent shall automatically establish a relationship between the email text/attachment with a case.

It is intended that MS Email Agent shall be deployed in the Europe, Middle East and Africa (EMEA) Region and North America (NA) Region. MS Email Agent aims at eliminating the disadvantages of the earlier existing service Email Clerk. Its advantages include:

- Attachment handling capability
- Improved security through digitally signed emails

Using MS Email Agent, all attachments undergo a rigorous exchange server filtering process. After stringent checks for possible virus checks, spam and hacking attempts, the attachment is allowed by MS Email Agent into Clarify File Share. There is a list of file types that are generally considered as a threat, and these file types are not allowed. They are listed as follows:

*.ADE	*.COM	*.INF	*.MDB	*.PCD	*.URL
*.ADP	*.CPL	*.INS	*.MDE	*.PIF	*.VB
*.BAS	*.CRT	*.ISP	*.MSC	*.REG	*.VBE
*.BAT	*.EXE	*.JS	*.MSI	*.SCT	*.WSC
*.CHM	*.HLP	*.JSE	*.MSP	*.SHB	*.WSF
*.CMD	*.HTA	*.LNK	*.MST	*.SHS	*.WSH

The following are the various scenarios, which explain in detail how MS Email Agent will respond to the customer and the Support Analyst. Also the attachment handling process is elucidated:

1. Upon successful submission of an inbound email (with or without an attachment), a standard confirmation mail advising a successful log addition will be sent to the customer. An appropriate notification is sent to the Support Analyst.
2. As the Exchange Server removes certain attachment types, including the executables, not all attachments will be added and handled. This is explained hereunder:

   Case A) Attachment received from an external mail alias (like a Hotmail

Account)

The customer creates an email with a disallowed attachment (Eg. A .exe file) and sends it to the Support Analysts' mailbox. The customer receives a successfully sent mail notification. The customer's email log will be added to the case with a note saying that it has an attachment.

A .txt file is added to the case, stating that:

"FILE DELETED. MICROSOFT IT HAS REMOVED SUPPORT.EXE SINCE IT WAS FOUND TO MATCH THE FILE FILTER = <IN>*.EXE FILE FILTER."

Case B) Attachment received from an Internal Mail alias (i.e. Microsoft Account)

The Analyst creates an email with a disallowed attachment (Eg. A .exe file) and sends it to the Clarify File Share. The analyst receives a mail notification that advises that the log entry was made, but the attachment was deleted. The analysts email log will be added to the case.

Some more expectations from MS Email Agent:

- MSEA needs to handle incoming spam
- MSEA should take care not to spam a customer, if he is OOF (Out Of oFfice) and his Auto Reply is on. For instance, the SRX number in the incoming email may be wrong. MSEA generates a response saying "SRX Invalid ". Now, if the customers' Auto Reply feature is

active, MSEA receives an automated response, which obviously does not have the SRX number. So another email is generated by MSEA. This cyclical deadlock should be avoided.

- MSEA should be able to handle a large volume of emails, and alongside tackle new viruses

2.7 Feasibility Study

Feasibility is a high level capsule version of the entire system analysis and design process. The objective is to determine quickly and at minimum expense how to solve a problem. The purpose of feasibility study is not to solve the problem but to determine if the problem is worth solving. The system has been tested for feasibility in the following points:

Operational Feasibility

There is a need for the want of change in the current system as more number of cases is being raised and the customer market reach is increasing at the same time the workload is increasing day by day. And the proposed model would eliminate the usage of multiple tools to achieve the operationality which can be performed by a single tool. The user of this tool can operate without any difficulty because of the ease of usage of the tool.

Technical Feasibility

We have sufficient technology to develop the project. The system can expand after it is implemented. The proposed system provides adequate responses to enquir regardless of the number or location of the user. There are technical guarantees o accuracy, reliability, ease of access and data security.

Financial and economical feasibility:

A system would be a cost saver for the organization and would enable to replace the tool which was from another organization.Finacial benefits must be equal or exceed the costs. We can recover the investment for the proposed system within a short period.

DESIGN

3.0 Analysis Model

This project is developed using the "**Microsoft Solution Framework**" software development model.

3.1 Microsoft Solutions Framework (MSF)

Consistently delivering high-quality technology solutions on time and on budget is challenging for any business. The Microsoft Solutions Framework (MSF) provides people and process guidance – the proven practices of Microsoft – to help teams and organizations become more successful in delivering business-driven technology solutions to their customers. MSF is a deliberate and disciplined approach to technology projects based on a defined set of principles, models, disciplines, concepts, guidelines, and proven practices from Microsoft.

MSF provides an adaptable framework for successfully delivering information technology solutions faster, requiring fewer people, and involving less risk, while enabling higher quality results. MSF helps teams directly address the most common causes of technology project failure in order to improve success rates, solution quality, and business impact. Created to deal with the dynamic nature of technology projects and environments, MSF fosters the ability to adapt to continual change within the course of a project.

MSF is called a framework instead of a methodology for specific reasons. As opposed to a prescriptive methodology, MSF provides a flexible and scalable framework that can be adapted to meet the needs of any project (regardless of size or complexity) to plan, build, and deploy business-driven technology solutions. The MSF philosophy holds that there is no single structure or process that optimally applies to the requirements and

environments for all projects. It recognizes that, nonetheless, the need for guidance exists. As a framework, MSF provides this guidance without imposing so much prescriptive detail that its use is limited to a narrow range of project scenarios. MSF components can be applied individually or collectively to improve success rates for the following types of projects:

- Software development projects, including mobile, Web and e-commerce applications, Web services, mainframe, and n-tier.
- Infrastructure deployment projects, including desktop deployments, operating system upgrades, enterprise messaging deployments, and configuration and operations management systems deployments.
- Packaged application integration projects, including personal productivity suites, enterprise resource planning (ERP), and enterprise project management solutions.
- Any complex combination of the above.

3.1.1 Key MSF Terms

As a framework, MSF contains multiple components that can be used individually or adopted as an integrated whole. Collectively, they create a solid yet flexible approach to the successful execution of technology projects. The following list defines these components.

MSF foundational principles. The core principles upon which the framework is based. They express values and standards that are common to all elements of the framework.

MSF models. Schematic descriptions or "mental maps" of the organization of project teams and processes (Team Model and Process Model—two of the major defining components of the framework).

MSF disciplines. Areas of practice using a specific set of methods, terms, and approaches (Project Management, Risk Management, and Readiness Management—the other major defining components of the framework).

MSF key concepts. Ideas that support MSF principles and disciplines and are displayed through specific proven practices.

MSF proven practices. Practices that have been proven effective in technology projects under a variety of real-world conditions.

MSF recommendations. Optional but suggested practices and guidelines in the application of the models and discipline.

The MSF Process Model

The MSF Team Model

3.1.2 The MSF Process Model

Envisioning:

A broad description of the goals and constraints of the project will be created in this.

Planning:

Functional specification, design of the solution, work plans, etc. will be prepared. Software design is actually a multistep process that focuses on various attributes of the program such as data structure, software architecture, procedural details and interface characterization. The design process translates requirements into a representation of the software that can be accessed for quality coding.

Developing:

This includes creating the code for implementation and documenting the code. The coding step performs the task of translating the design into a machine readable format. If the design is performed in a detailed manner, coding can be accomplished mechanically.

Stabilizing:

This stage requires the project team to integrate, load and do a beta testing of the solution. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested.

Deploying:

Deployment of the solution technology and site components, stabilizing the deployment, transferring the project to operations and support, and obtaining the final customer approval for the project.

MSF guidance for these different project types focuses on managing the "people and process," as well as the technology elements that most projects encounter. Because the needs and practices of technology teams are constantly evolving, the materials gathered into MSF are continually changing and expanding to keep pace. Additionally, MSF interacts with Microsoft Operations Framework (MOF) to provide a smooth transition to the operational environment, which is a requirement for long-term project success.

Embracing MSF organizationally, however, is a demanding initiative that requires leadership support and careful planning. An effort of this nature may entail some change in organizational culture as well as individual habits. This makes it similar in many ways to the introduction of a new solution, so it is not surprising that many MSF techniques can be usefully applied to the implementation of MSF itself. Specifically, these would include a clear vision, representation of similar roles, versioned releases, risk and readiness management, and learning from experience. Microsoft recognizes the challenge this represents and has established many channels for providing training and assistance to organizations implementing MSF.

In summary, Microsoft Solutions Framework (MSF) provides proven practices for planning, building, and deploying a variety of technology solutions, combining aspects of software design and development, and building and deploying infrastructure into a single project life cycle for guiding technology solutions of all kinds.

Microsoft Solutions Framework (MSF) is a powerful tool that helps organizations address the key areas critical to technology project success—people and processes. Originating from real-life projects of Microsoft and its partners and customers, MSF provides guidance on the application of a defined set of principles, models, disciplines, concepts, and proven practices that have been shown to help prevent the primary causes of technology project failure.

Title	Description	Comments
MSEmailAgent replacement tool for Email Clerk	A new tool MSEmailAgent to be developed so to replace the current inbound email processing functionality of Email Clerk and enhance inbound email processing by handling attachments	

3.3 Risks

Risk Description	Probability ([Very] Likely, [Very] Unlikely)	Severity (High; Med.; Low)	Type (Functionality; Usability; Schedule; Development)
Any Open issues listed under	Unlikely	High	Development and Schedule
Availability of Dev and Test hardware resources	Unlikely	Med	Schedule
Defects in the end product due to problems faced during integration	Unlikely	High	Development and schedule.
Missed Deadline	unlikely	High	Schedule

3.4 Hardware & Software Requirement

Hardware Requirement:

The project is developed on the Microsoft Platform. Minimum computer configuration is recommended based on the Microsoft operating system used. The System Configuration on which the project is going to be developed is mentioned below:

Server Machine:

P IV 1.8 GHz.
2 GB RAM
120 GB HDD
52X CDROM Drive
10 / 100 Mbps NIC
4 / 8 GB Tape Drive

Client Machine:

P IV 1.8 GHz.
256 MB RAM
10 GB HDD
52X CDROM Drive
10 / 100 Mbps NIC

Software Requirements:

This system is developed to work on the Microsoft Platform which has .NET Framework 1.1 installed.

- .Net Framework
- Compass server
- IIS Server
- Exchange Server 2003 with WebDav
- Appropriate permissions on Servers

3.5 Design Principles and Methodologies

3.5.1 Introduction to Design

Of all the phases in the development of a software project or for that matter any kind of a system, the efficient and consistent designing of the system can be called one of the most important phase. System analysis and design is perhaps the most studied subject in the software field today. The system design or the architecture is incorporated in the requirements specification. The Design of a system is said to be efficient if the system design satisfies the requirement of the system. A design should be variable, complete and traceable. However the two most important properties of a good design are efficiency and simplicity. Efficiency of a system is concerned with the proper use of scarce resources of the system. The design must also be documented for later evaluations etc.

The basic principles of a system design are:

Problem Partitioning and Hierarchy:

The complexity of large problems and the limitations of human capabilities do not allow large problems to be traded as huge monoliths for solving larger problems. The basic principle is the time-tested principles of divide and conquers. Using these principles larger problems are divided to smaller problems. The solution is found for each sub problem and all the solutions are merged together to arrive at the solution of the larger problem.

Abstraction:

Abstraction is a very powerful concept and is used in all engineering disciplines. It is a tool that permits a designer to consider a component at an abstract level without worrying about the details of the implementation of the components of the system. The level of abstraction is to be incorporated into the design is decided by the system designer according to his/her understanding of the system.

Modularity:

The modularity of the system is the ability of dividing the whole system into subsystems, which are almost independent of each other. The modules/subsystems are to be defined in such a way that the number of modules is neither too many nor too small. Each subsystem must be self-contained in it. By bringing together all these modules we must be able to configure the whole system.

Top down and bottom up strategies:

A top down design approach starts with identifying major components of the system, decomposing them into their lower level components and iterating until the desired level of detail is achieved. The bottom up approach starts with designing the most basic components and proceeds to the higher level components that use these lower level components. One of these two design approaches is chosen based on the requirements of the system to be designed.

3.5.2 Design Methodology

A design Methodology is a systematic approach to creating a design by application of a set of techniques and guidelines. Most design methodologies focus on the system design. Most current design methodologies essentially offer a set of guidelines that can be used by the developer to design the system. These techniques are not formalized and do not reduce the design activity to sequence of steps that can be followed by the designer.

The design of the system produces the details that state how a system will mee the requirements identified during the system analysis. The design is a solutio transaction of requirements in two ways of meeting them. The design will determine th success of the system. The system design involves both logical and physical design.

The process of logical design starts with identifying various internal processi functions, decomposing high-level functions into sub functioned and establishi relationships and

Interconnections ... is also termed as logical architecture.

The basic design criteria, which serves as guidelines are as follows:

- A design should exhibit a Hierarchical organization that makes intelligent user control over components of software
- A design should be modular i.e., the software should be logically partitioned into components that perform specific functions and sub routines.
- A design should contain distinct and separable representation of data and procedure.
- A design should lead to modules that exhibit independent functional characteristics.
- A design should lead to interface that reduce the complexity of connections between modules and with external environment.
- A design should be derived for a repeatable method that id driven by the information during requirement analysis.

3.5.3 System Design

The design phase begins when the requirement specification document for the software to be developed is available. When the requirement specification activity is entirely in the problem domain design is the first step to moving from the problem domain towards the solution domain.

Design essentially bridge between requirement and the final solution for satisfying the requirements. The term design is used in two ways, used as a Verb, it represents the process of design when used as a noun; it represents the result of the process, which is the design for the system. The Goal of the design is to produce a module, or representation of a system. The produced module is called design of the system.

The design of the system is essentially blueprint of the system or plan for the system However we consider the system to be a set of components with clear behavior

Interconnections among the functions. This is also defined as logical architecture.

The basic design criteria, which serves as guidelines are as follows:

- A design should exhibit a Hierarchical organization that makes intelligent user control over components of software
- A design should be modular i.e., the software should be logically partitioned into components that perform specific functions and sub routines.
- A design should contain distinct and separable representation of data and procedure.
- A design should lead to modules that exhibit independent functional characteristics.
- A design should lead to interface that reduce the complexity of connections between modules and with external environment.
- A design should be derived for a repeatable method that id driven by the information during requirement analysis.

3.5.3 System Design

The design phase begins when the requirement specification document for the software to be developed is available. When the requirement specification activity is entirely in the problem domain design is the first step to moving from the problem domain towards the solution domain.

Design essentially bridge between requirement and the final solution for satisfying the requirements. The term design is used in two ways, used as a Verb, it represents the process of design when used as a noun; it represents the result of the process, which is the design for the system. The Goal of the design is to produce a module, or representation of a system. The produced module is called design of the system.

The design of the system is essentially blueprint of the system or plan for the system. However we consider the system to be a set of components with clear behavior or

services to its environment. A component of system can itself b considered a system with its own components. In software system a component is a software module.

The design process for a software system often has two levels. At the first level, the focus is on designing which modules are needed for the system, the specification of these modules and how these modules are interconnected. In the second level the internal design of the modules or how the specification of the modules can be specified is decided upon. This level is often called detailed design or logic design. Detailed design essentially expands system design to contain a more detailed description of the processing logic and data structure such that the design is sufficiently complete for coding.

3.6 Project Study

MSEmailAgent looks for specific tags in the Inbound Email and after successful validation, creates logs against Clarify Cases or Sub-cases.

3.6.1 Features of MSEmailAgent

The main feature of MSEmailAgent is its ability to process inbound email messages and create email logs against Cases / Sub Cases. Also important is its capability to handle and process attachments contained within these emails. It is intended to deploy this tool in EMEA region and North America. The number of attachments that can be processed from a single email is not restricted to one; multiple attachments can be processed at a time. MSEmailAgent also processes digitally signed emails

3.7 Prerequisites

1. The new MSEmailAgent will reside on an appropriate 2003 server (or existing Em
Clerk server).
2. WebDav feature should be enabled on the Exchange server, where the mailbox th
is being polled resides.
3. The Mailbox to be polled would be enabled for web access, this is required to ac
and read attachments using web client.
4. .Net Framework v1.1.4322 is installed on the server where MSEmailAgent will
deployed.
5. Two instances of the tool shall not be run against the same mailbox
6. Targeted to be deployed in the EMEA and NA regions.
7. Email Clerk will be retired on successful deployment of MSEmailAgent. (The
tools will not be run simultaneously)

3.8 Dependencies

The MSEmailAgent tool would be deployed as a windows service. The too
would make HTTP requests to poll a mailbox on the exchange server at regular in
and process the retrieved emails.

For this tool to function as expected the following is required.

1. IIS service should be running on the Exchange server
2. WebDav feature should be enabled on the Exchange server
3. The COMPASS server should be up and running for the service to cre
logs in the compass database.
4. The MSEmailAgent service needs to **be stopped before shutting down**
Server or the IIS service.
5. The new service would **be required** to run under an NT user account.

6. The NT user account used for running the service should have permi execute required sprocs on compass database.
7. The MSEmailAgent server should have permissions / access to be abl calls to Sprocs on the compass server.
8. The MSEmailAgent server should have permissions / access on the C share
9. The NT user account used for running the service should have permiss Clarify File share to upload attachments.
10. Email Clerk will be retired on successful deployment of MSEmail two tools will not be run simultaneously)
11. The Account used to change the configuration values for the tool and for running the service should have read write permissions on the r MSEmailAgent server.
12. Clearing of the log files from the log directory would be responsibil team.

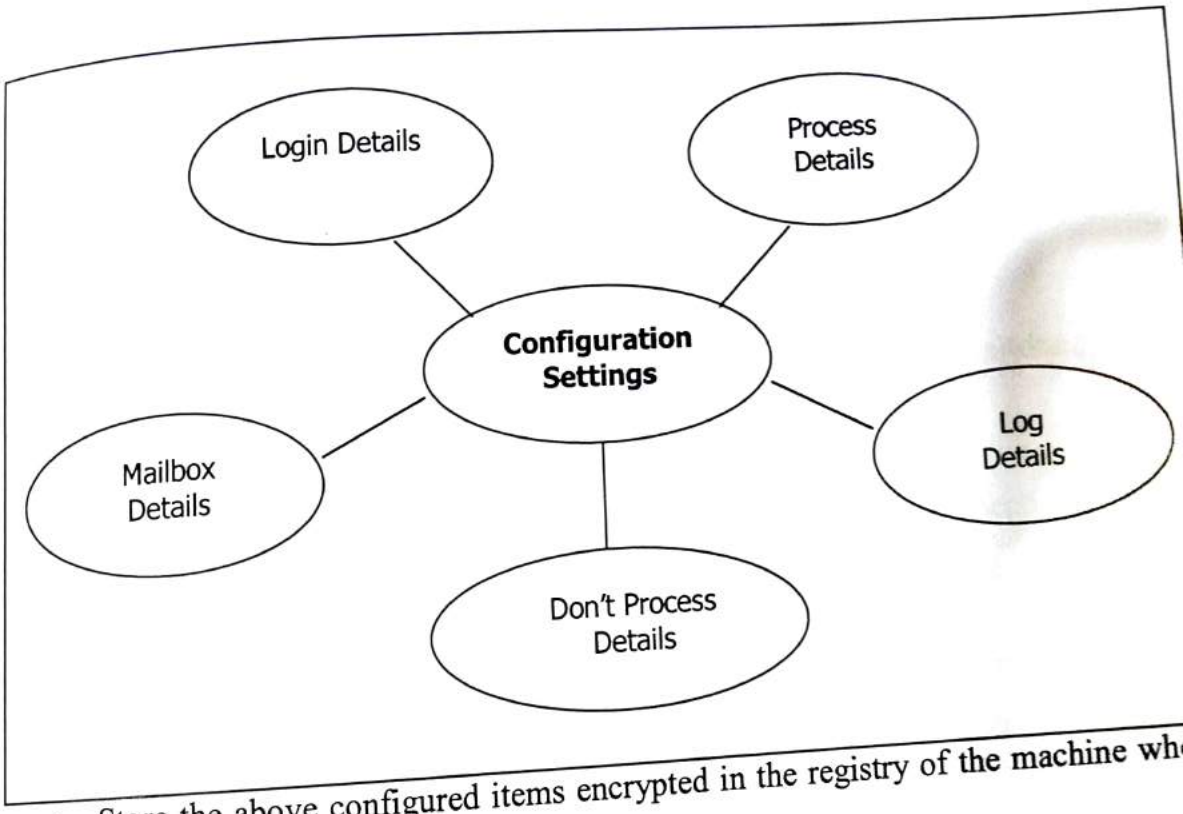
3.9 Expectations

- MSEmailAgent should be able to process Inbound Emails to the t more per day
- MSEmailAgent should be able to process Inbound Emails to the hour
- It should notify customers after successful processing of Inb given format
- Upon successful processing, it should create logs against ca System

- It should be able to process and save inbound email attachments to file shares, which are thus accessible through the Clarify System

- It should provide a User Interface to provide Configuration Options

3.10 Configuration Settings



- Store the above configured items encrypted in the registry of the machine wh MSEmailAgent runs

- Move emails which are not successfully processed to an ERROR folder in mailbox from where the emails can be reviewed at a later time.

- Provide option to delete or move (to SUCCESS folder) successfully proce Emails

- Three levels of process logging available (Basic/Medium/Verbose)

- All log files are created on the machine where MSEmailAgent runs and at a location as mentioned in the configuration screen
- Log files names include date stamp. Format: MSEmailAgent_YYYYMMDD_HHMISS.log
- A new text log file is created every day. Logs created by mails processed in a day are appended to the daily log.
- A new log is generated each time the service is restarted.
- The Start and Stop events of the service will be registered in the event log.
- All errors will be logged in the event log as well as the log file.
- No email alert will be sent from MSEmailAgent, external monitoring should handle all alerting
- The MSEmailAgent service would read a given number (as mentioned in the configuration screen) of emails in one batch.
- If a particular email for some reason is not moved or deleted as appropriate, the subject would be added to the "Don ' t Process Email with Subject " box and will need intervention from OPS. The same will also be logged in the event log.
- Any sender identified as a " Mail Bomber " (Sending more that " Spam Alert " number of emails in the last " Spam Reset " number of emails would be added to the " Don ' t Process Email From " box and will need intervention from OPS. The same will also be logged in the event log.
- All email sent out to customers are done so using web mail and would require a SMTP server to be specified in the Configuration screen.

After Email processing, two types of results might occur:

The Success email is sent to the customer in the following forma

- Subject: Re: <Sunject of processed email>
- Body: Your case email was accepted.

  Case_ID_NUM: SRl031114600083

The Error email is sent to the customer in the following format

- Subject: <Subject of Processed Email>
- Body: Your mailed case was rejected

  Kindly refer to the following format:

  CASE_ID_NUM: SRXxxxxxx6xxxxx

  MESSAGE: Enter the text of your message her

  Note: There should be a space after CASE_ID_

  Message: The Mailbox needs to have web-acce

DEVELOPMEN

4.0 Development Introduction

In order that the MS Email Agent system is developed, first, the general approach to email handling is discussed. User X has an email-ID with hotmail.com, an email service provider - for instance, xxxx@hotmail.com. When X sends an email from his email account, it is sent to the mail server (Simple Mail Transfer Protocol - SMTP Server) of the Internet Service Provider. X sends an email to yyyy@yahoo.com.

The Source SMTP Server (i.e. the Hotmail Server) communicates with the Domain Name Server Mail Server examines the email id, i.e. yyyy@yahoo.com, and identifies that the destination needs to be the mail server of Yahoo. It then resolves the IP address of the Destination, and decides the routing to the destination mail server. Finally, the mail goes to reside in the mailbox of User ymen, which exists on the Yahoo Mail Server.

For implementing MS Email Agent, we primarily use the following entities:

- A Windows 2003 Server in order to implement the MS Email Agent web service
- An Exchange Server where the mail box that MS Email Agent polls, resides
- An SMTP Server which communicates with the Exchange Server
- Clarify Database (Compass), where the stored procedures exist, and so do the email logs and attachment file shares

4.1 Programming Environment

4.1.1 Introduction to .NET

Microsoft announced its .NET initiative in June 2000 and brought to market at the beginning of 2002. The .NET is a layer on top of Windows and other operating systems.

- A key aspect of the .NET strategy is its independence from a specific language or platform. A .NET application can include any combination of .NET-compatible languages, such as C#, J#, Visual C++ .NET, Visual Basic .NET, Pascal, Python, Fortran, Eiffel, etc.
- A key component of the .NET architecture is Web services, which are applications that expose functionality to clients via the Internet. Web services extend the concept of software reuse by allowing programmers to concentrate on their specialties without having to implement every component of every application.

4.1.1.1The .NET Framework and Common Language Runtime

The .NET framework forms the core of the .NET technology. It consists of a runtime environment and an object-oriented class library that covers all areas of Windows and Web programming

- *Runtime environment*: this offers garbage collection, security, versioning and interoperability between programs written in different languages
- *Object-oriented class library*: this provides a rich set of functionality for graphical user interfaces, web interfaces, database connectivity, collections, threads and reflection.

Common language runtime (CLR): the runtime environment under which .NET programs are executed. A virtual machine performs the following

- *Compilation*: programs in .NET languages (C#, VB, C++ and J#) are translated into programs in **common intermediate language** (CIL).
- *Execution*: just before programs are run, the CIL programs are compiled (*just in time*) into the machine code.

Common type system (CTS): in order to be translated into the CIL, the .NET languages must use the same sort of data types. The CLR defines a common type system **(CTS)** that describes how classes, interfaces and primitive types are presented.

Common language specification (CLS): the minimal subset of the CTS that all languages must support if they want to make use of .NET's language interoperability

Garbage collector: it is responsible for reclaiming the storage of objects once they are no longer needed.

Verifier: it is used to check that the type rules of the CTS have not been violated when a program is loaded and translated into machine code.

Assembly: collection of classes and other resources such as images

- An assembly is stored either as an .exe file or as a .dll file
- An assembly contains metadata in addition to code. The metadata holds the full type information of the classes, fields, methods and other program elements.
- An assembly also contains a manifest thought of as a table of contents

Assemblies: the smallest components that can be individually deployed.

- Assemblies are used for version control
- Assemblies do not have to be recorded in the Windows registry. They are simp copied into the application directory.

The Base Class Library: contains the most important classes of .NET and can be use by all .NET languages. The base class library is divided into namespaces, each of whi deals with a particular functionality.

- **System.Collections**: contains classes that manage collections of objects such lists, sets, trees, dynamic and associative arrays and hash tables
- **System.IO**: contains classes for input and output
- **System.Threading**: provides classes for parallel programming
- **System.Net**: deals with network programming
- **System.Reflection**: allows access to metadata and runtime type information
- **System.Windows.Forms**: concerns with graphical user interfaces
- **System.XML**: contains classes for the creation and reading of data in (extensible Markup Language) format.

4.1.1.2 Who should use .NET?

You should learn .NET if:

- You like to use Microsoft technologies (eg. Windows, Windows Server)
- You like to use the latest and most powerful application devel technologies
- You want to build Windows, Web or Mobile Device applications
- You want to reduce development time
- You want to build applications that can communicate with each other Remoting or XML Web Services)
- You want to build enterprise level applications
- You want to build components that can be used for both Web and applications without modifying the code
- You want to use a technology that has enormous amounts of reso

4.1.3 Advantages of .NET

Powerful Development

When developing with the .NET Framework developers work with 100% object oriented languages, and can take advantage of all the features previously only seen in windows development, including debugging, tracing, reflection, compiled code (faster), and much more.

Extensive Code Library

The .NET Framework has an enormous amount of built-in functionality (over 4500 classes), allowing developers to focus more on developing their application than writing the tedious lines of code that perform the most common and mundane tasks. The built-in functionality includes, but is not limited to:

- Data Access - ADO.NET
- Web UI - Pages, controls, state, file uploading, etc.
- Windows UI - Forms, controls, GDI+, etc.
- Web Services - Communication between applications via the internet using XML Web Services or Remoting.
- Cryptography - MD4, MD5, SHA1, TripleDES, Base64, and many mory
- Security - Forms authentication, Windows authentication, Passport authentication
- Errors, performance, and logging
- Localization - Different text for different languages

Using the built-in functionality of the .NET Framework you can use less code and do more.

Language Interoperability

Due to the .NET common language runtime, your application can run components written in different .NET compatible languages all at once. This means that your team can consist of developers who are fluent in different languages and when the

components come together they will work seamlessly. The

has 25 supported languages, the following of which are built-in:

- C# - The most commonly used .NET language, with a syntax that is similar to C and Java, but as simple to learn as traditional VB (if not easier).
- VB.NET - Based on traditional VB, VB.NET uses a very similar syntax but has complete access to the functionality of the .NET framework
- JScript .NET - A .NET compatible language based on JScript.
- C++ .NET - Traditional C++ that can use managed extensions to take advantage of the memory management, and various other features in the .NET runtime
- Other languages are also in development, including Cobol .NET, Perl .NET, and more.

Painless Deployment

To install an ASP.NET web application you can simply copy all your files to the server and you're ready to go. No need to register your DLLs on the server like traditional components, they're registered automatically once uploaded. No need to worry about overwriting DLLs either, your old DLLs will be kept in the global assembly cache until no other components need them, therefore, no more 'DLL hell'.

Code Re-use

Like all good object oriented programming languages you can use classes/objects to cleanly store functionality that you can use across multiple applications. The code can either be compiled to a DLL and referenced by multiple applications, or just copied as a class and compiled with the rest of the application code. Your custom functionality classes can be called from your other code, or you can create a class in your application and inherit the common class (and all its functionality), then override some of the functions and properties to customize the way it works.

Easy Migration

If you're migrating from traditional ASP to ASP.NET there's no need to convert all your code straight away. Your ASP pages will run fine along side ASP.NET so you can migrate slowly, a section of your site at a time. Or just leave the existing ASP site as it is and wait until the next update to implement the ASP.NET version.

Web Controls

The .NET Framework has simplified many of the common techniques of displaying, validating, and managing data.

Some of the controls used for displaying records of data include:

- Data Grid - Using virtually any data source you can bind your data to a Data Grid control and have it automatically render a table-like layout displaying all your data. You can also customize the way it displays data to suit your application.
- Data List - Similar to the Data Grid but you specify the header, item, alternating item, and footer templates, then your data is displayed using those templates.
- Some of the controls used to display custom data or content include:
- Table - Rendering the same as a normal table this control can be programmatically modified and customized by your code, and can contain anything including text, other controls, images, data, etc.
- Label - Typically used to display simple text you can programmatically modify the style of the label which will be applied to the text when rendering.
- Some of the controls used for validating data include:
- RequiredFieldValidator - Assign this to a control on your page and it will ensure that the user enters data.
- RegularExpressionValidator - Assign this to a control to make sure the data inside it matches the specified validation expression.
- CompareValidator - Assign this to two controls to make sure they match, such as a password and password confirm box.

Mobile Device Support

Not only does .NET run on Windows, and in your browser, you can also create pages to view through your phone, PDA, or other WAP enabled device. The build-in mobile controls let you access all the same data and functionality it your ASP.NET application but render it in a format compatible with mobile devices.

Web Services

Web Services aren't a new concept, but using the .NET Framework you can turn any of your classes into a web service by adding a just a few extra lines of code, then using one of the built-in tools to generate the code required to interact with it. Once done you can communicate between your applications (windows and web) almost as if they're on the same machine.

Development Tools

The .NET Framework includes C# and VB.NET compilers so that you can create web and windows applications using as little as notepad.

If you want to improve your productivity you may like to use one of the various development IDEs available, including the following:

- Visual Studio .NET - Microsoft's own .NET development environment contains a myriad of ways to speed up your development, but with a hefty price.
- ASP.NET Web Matrix - A free IDE designed to get you developing ASP.NET applications quickly. It lacks a lot of the features of VS.NET but the price is a little more comforting.
- SharpDevelop - An open source application that is surprisingly similar to VS.NET but doesn't yet support ASP.NET.

4.1.2 Introduction to VB.NET Windows Forms

In VB.NET, Windows forms have changed completely. Now a Windows form is a representation of the System.Windows.Forms.Form class in the .NET Framework. This article offers an introduction to the new underlying technology by giving examples of how to do the basic tasks of writing a Windows application. This includes creating a form, adding controls, and responding to events. It also gives the object-oriented analysis of a WinForm, concluding with the most powerful new feature that enables code reuse: inheritance.

The new version of VB adds a new feature that entitles it to be called an object-oriented programming language: inheritance. Now, no one can argue that VB is not a purely OOP language. But what does this mean to VB programmers worldwide? Does it mean that every VB programmer must now learn OOP? Well, the answer is not that straightforward. The previous versions of VB have been known for their ease of use. Everyone remembers how easy it was to drag and drop controls and create a Windows application without any understanding of OOP. The half-OOP features in those versions were probably not often used by the majority of programmers. The ease of use continues with the current version; with Visual Studio.NET, creating a VB Windows application is as easy as it used to be. However, this article will show you why you should master OOP to really take advantage of the full power of VB.NET.

Visual Basic .NET has many new and improved language features — such as inheritance, interfaces, and overloading — that make it a powerful object-oriented programming language. As a Visual Basic developer, you can now create multithreaded, scalable applications using explicit multithreading. Other new language features in Visual Basic .NET include structured exception handling, custom attributes, and common language specification (CLS) compliance.

The CLS is a set of ... things as data types and how objects are exposed and interoperate. Visual Basic .NET adds several features that take advantage of the CLS. Any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET. And you, as a Visual Basic user, can access classes, components, and objects from other CLS-compliant programming languages without worrying about language-specific differences such as data types. CLS features used by Visual Basic .NET programs include assemblies, namespaces, and attributes.

Visual Basic .NET supports many new or improved object-oriented language features such as inheritance, overloading, the **Overrides** keyword, interfaces, shared members, and constructors. Also included are structured exception handling, delegates, and several new data types.

4.1.3 Introduction to Exchange Server 2003

Exchange Server 2003 is the latest version of Microsoft's industry, leading communications server. Exchange Server 2003 provides many new features and enhancements to improve reliability, manageability, and security. Exchange Server 2003 helps IT to deliver the service levels and capabilities demanded by end user while helping organizations to reduce their TCO through areas such as server and site consolidation. Although this new version of Exchange will run on Microsoft Windows® 2000 Server, Exchange Server 2003 is the first version of Exchange designed to run or and take advantage of Microsoft Windows Server™ 2003. This document provides a overview of features in Exchange 2003 in addition to features in Microsoft Offic Outlook® 2003, Microsoft Office Outlook Web Access and mobile device support usin Outlook Mobile Access and Exchange Server ActiveSync®.

4.1.3. Exchange Server 2003 delivers breakthrough advances for both IT professionals and business users.

- **Information workers** get their work done more quickly because Exchange 2003 and Outlook 2003 work together to deliver consistently efficient access to personal business information independent of network characteristics. Mobile users find it easier than ever to stay up-to-date with the information flowing into their office mailboxes no matter what device they choose to use.
- **Information technology (IT) managers** appreciate advances in administration, security, scalability, reliability, and built-in support for mobile computing that both lower the costs of managing an Exchange-based communications infrastructure and make it easier to deliver mission-critical messaging services. The ability to help IT managers to simplify their operations while meeting Service Level Agreements is one of Exchange Server 2003's top design goals.

Improvements in the way Exchange Server 2003 and Outlook 2003 work together dramatically improve the performance of Outlook over low or fluctuating bandwidth conditions. Checking e-mail over a dial-up connection or low-bandwidth mobile network, such as GPRS (general packet radio service) and 1xRTT (single carrier radio transmission technology) CDMA, feels much more like working in the office. Using the new Outlook 2003 user interface (UI) makes it faster and easier for information workers to optimize their e-mail.

Information workers rely on Outlook Web Access to reach their personal business information from any computer connected to the Internet. With Exchange Server 2003, Outlook Web Access now features a UI very similar to that found in Outlook 2003, providing users with a consistent experience whether they are connecting from their desktop computer or an Internet kiosk. In addition to improvements in the areas of security and performance, Outlook Web Access supports more Outlook functionality, including a spelling checker, task management, and antispam protection.

4.1.3.2 Client access functionality

Exchange Server offers integrated collaborative messaging features such as scheduling, contact, and task management capabilities. Exchange Server 2003 runs on Microsoft Windows Server 2003 and Microsoft Windows 2000 Server operating systems. Microsoft Office Outlook 2003 runs on Windows-based computers and communicates with the server running Exchange Server through the MAPI protocol that includes powerful messaging and rich collaboration capabilities. Exchange Server also accommodates other client access through its support for Post Office Protocol 3 (POP3) and Internet Message Access Protocol 4 (IMAP4) protocols as well as support for Simple Mail Transfer Protocol (SMTP). Microsoft Outlook Web Access, a service in Exchange Server, accommodates what are known as thin clients (Web browser–based access clients).

4.1.3.3 Mobile access functionality.

Exchange Server 2003 supports mobile devices such as Pocket PCs and Smart phones, thereby enabling the synchrony of Inbox, Calendar, and Contacts and Tasks lists.

Exchange Server supports the following types of clients:

- POP3 or IMAP Clients
- Outlook 2003 (MAPI Clients)
- Outlook Web Access (Browser Based Clients)
- Outlook Mobile Access (Mobile Browsers), and so on

The Exchange Server collaboration features aid in quick and efficient information sharing. Typical collaborative scenarios include maintaining shared address lists that everyone can view and edit, scheduling meetings that include people and conference rooms by viewing associated free or busy schedules, the ability to grant other people, such as administrators, access to personal mailboxes to people. Also, "rules" can be set up and managed for processing messages on Exchange Server. They give the flexibility to create auto-responses and automatic filing of incoming messages.

4.1.4 Introduction to WebDav

Exchange WebDAV notifications are a mechanism used performing event-based processing in a Web-based document storage environment. WebDAV Notifications allow a client application to perform actions based on changes in the state of data stored on the server, including creating, updating and moving documents, and processing new e-mail messages. Exchange WebDAV notifications help client applications display up-to-date information or inform the end user of document updates by another user.

With increased focus on Internet standards and network interoperability, WebDAV (Web Distributed Authoring and Versioning) has become an important communication protocol for the Web as an extension to HTTP 1.1. The WebDAV specification, which has been defined in detail in the RFC 2518 has been published by the Internet Engineering Task Force (IETF). Because of its inherent integration with Extensible Markup Language (XML), WebDAV not only has a large dependency on XML, but also has emerged as an excellent method for communicating XML data over the Web.

4.1.4.1 Why not HTTP alone?

HTTP 1.1 (Hypertext Transfer Protocol) has proven itself as a flexible, universal protocol for transferring data by virtue of the fact that the Web has become the bastion of the Internet. However, HTTP has some obvious shortcomings that have limited its adoption as a comprehensive Internet communication protocol: it works well for static documents intended for viewing, but does not provide the means to handle documents in a manner sophisticated enough to provide clients with rich authoring capabilities.

For example, when two authors make changes simultaneously to a document without consulting one another, the "lost update" problem occurs. Only the revisions made by the last person to upload the document back to the server will remain, and the changes made by the other author will be lost.

It was the goal of the IETF WebDAV working group to design a protocol that would provide functionality that could be required by any distributed authoring tool in a

standards-[illegible] [illegible] simple concepts of collaborative authoring tools.

- **Overwrite Protection.** HTTP 1.1 has no method of ensuring that clients can protect resources and make changes without fear of another client simultaneously editing them. With WebDAV, you can lock resources in a variety of ways to let other clients know you have an interest in the resource in question, or to prevent other clients from being able to access the resource.
- **Resource Management.** HTTP deals only with direct access to an individual resource. WebDAV provides a means of organizing data more efficiently. WebDAV introduces the notion of the *collection* (analogous to a file system folder), which can contain *resources*. Resource management via WebDAV includes the ability to create, move, copy, and delete collections, as well as the ability to do the same things to the resources or files within the collection.
- **Document Properties.** Different types of data have unique properties that help describe the data. For example, in an e-mail message, these properties might be the sender's name and time received. In a collaborative document, these properties might be the original author's name and the name of the last editor of the document. As the types of documents that people use diversify, the list of possible property types becomes infinite. XML is the type of extensible communication vehicle required by WebDAV.

4.1.4.2 Exchange WebDAV Notifications:

Exchange WebDAV notifications are a mechanism used performing event-base processing in a Web-based document storage environment. WebDAV Notifications allo a client application to perform actions based on changes in the state of data stored on t server, including creating, updating and moving documents, and processing new e-m messages. Exchange WebDAV notifications help client applications display up-to-d information or inform the end user of document updates by another user.

WebDAV notifications are typically used in remote client applications that already use WebDAV to communicate with the Exchange server. Typical applications include Web-based messaging clients and collaboration applications using Exchange public folders.

WebDAV returns information in text- and XML-streams that contain the item data, properties and error information. WebDAV is an extension of HTTP, so no connection state information is retained between transactions.

4.1.4.3 What does WebDAV support?

Applications that use WebDAV are typically Web-based, thin-client applications. However, traditional Windows GUI applications can be developed that use WebDAV to communicate with the Exchange server. In addition, WebDAV is frequently used as the communication mechanism between an application middle-tier and the Exchange server.

WebDAV is often ideal for remotely accessing Exchange. Because it communicates using the same ports that HTTP and HTTPS use, corporate firewalls and routers often require no special configuration.

Because WebDAV is a protocol, any programming tool and language that correctly send and receive HTTP requests and responses can be used to create applications that access Exchange using WebDAV.

4.1.4.4 Advantages of WebDAV:

WebDAV has two great advantages: It could be used to access a Mailbox on a different computer, and it requires only http port connectivity between client and server. WebDAV requires only http connectivity because it is an extension of the http protocol.

4.1.4.5 The Format of WebDAV Requests

HTTP 1.1 provides a set of methods that clients can use to communicate with servers and specifies the format of responses from servers back to the clients that have issued requests. WebDAV fully adopts all of the methods of this specification, extends some of

these methods, and introduces additional methods to provide the functionali
The methods used in WebDAV are:

- **Options**, **Head**, and **Trace**. Primarily used by applications for di
tracking server support and network behavior.
- **Get**. Retrieves documents.
- **Put** and **Post**. Submits documents to the server.
- **Delete**. Destroys resources or collections.
- **Mkcol**. Creates collections.
- **PropFind** and **PropPatch**. Retrieves and sets properties on bot
collections.
- **Copy** and **Move**. Manages both collections and resources within
namespace.
- **Lock** and **Unlock**. Overwrites protection.

The general structure of WebDAV requests follows the format of HTTP
of the following three components:

- **The method**. States the method (described previously) to b
client.
- **Headers**. Describe instructions about how the task is to be com
- **A body** (optional). Defines the data used in the instruc
instructions, about how the method is to be executed.

a crucial element in the

4.1.4.6 What XML Means to WebDAV

The method of communicating all of the information in HTTP was solely the responsibility of the headers in requests and responses. This imposes some limitations on transfers. It is difficult to apply header information to multiple resources in a request and to represent hierarchy.

Because of its inherent extensibility, XML was chosen to describe how these instructions are communicated. XML is crucial to the operation of WebDAV because it provides:

- A method of formatting instructions describing how data is to be handled.
- A method of formatting complex responses from the server.
- A method of communicating customized information about the collections and resources handled.
- A flexible vehicle for the data itself.

At a high level, a WebDAV instruction processor is really a set of logic that interpre WebDAV methods, followed by an XML parser that interprets the majority of t information communicated.

How does the use of XML in WebDAV turn this technology into such a powerful to First, XML provides a way of separating data from either the methods that act on data, or the way the data is presented. This enables straightforward and consis abstraction of data. For this abstracted data, WebDAV provides a method of consis unified transfer between all tiers in the network architecture over channels tha familiar to existing network architectures. This technology enables a much higher le interoperability between both Microsoft products and third-party applications.

Second, XML enhances WebDAV by providing a means for extensibility. XML clients to describe and set properties on a WebDAV server. These properties can t used to index, search, and process resources on the server. Because of the i extensibility of XML, the types of properties and uses for these properties are infir

4.1.4.7 Installing WebDAV

- First install IIS on the Exchange Server, then followed by W
- After installation, WebDAV should be enabled, by checkin in the IIS Manager.
- Then the next step is to have WebDAV installed on the cli

4.2 Project Process Module:

4.2.1 Working in Brief:

The journey of an inbound email starts from a customer. But before this, a case needs to be created. So the chronological sequence of events is described below:

Sequence 1: Customer Calls Helpdesk

- Customer has a problem – calls Helpdesk
- Helpdesk creates a Case, and gives the Case ID to the customer for future reference
- Customer is instructed to send emails to a particular ID, say abc@microsoft.com, and mention the Case ID in the subject line for any further correspondence.
- Here the telephone conversation ends and the emailing part begins.

Sequence 2: Customer Sends an Email

- The customer intends to send an email regarding his case – probably additional information like a screen snapshot, or some clarification, or an update.
- The customer sends an email to abc@microsoft.com from his personal mail ID, with the Case ID in the subject line.
- Here the Customer's role ends.

Sequence 3: Email Traversal from Source to Destination

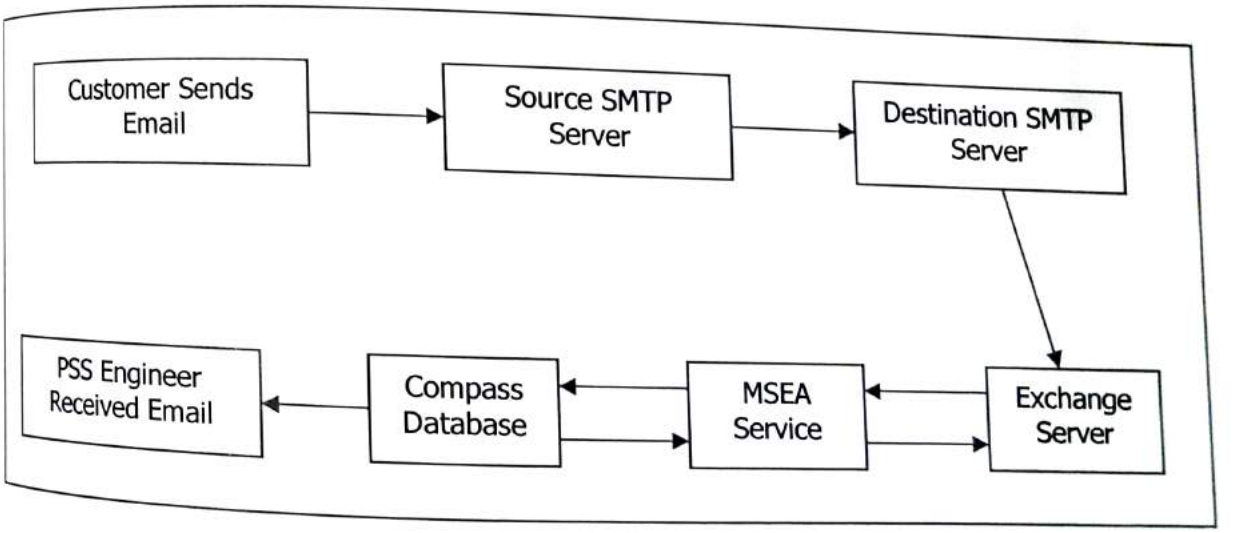
- The destination email ID is resolved and the Domain Name Server resolves the IP address of the destination email address, i.e. abc@microsoft.com. The email reaches the SMTP Mail Server of Microsoft.
- The email then traverses from the SMTP server to the Exchange server

Sequence 4: At the Exchange Server

- The mail is scanned, and mails with inappropriate extensions are filtered.
- Then the text of the email is checked. If it is a valid type, then it is allowed.
- Finally, the mail goes to reside in the Exchange Store, which is a temporary storage area. It is in fact a mail box which has web-access enabled.

Sequence 5: MS Email Agent Email Processing

- MS Email Agent polls the mailbox in the Exchange Server through HTTP Requests, picks up the email, and sends it back to the Windows 2003 Server where it resides, through a HTTP Response.
- Here it is validated, i.e. the appropriate Case ID is checked, and if it is valid, a stored procedure that creates an email log in Compass is called. After a successful log creation, if there are any attachments, they are uploaded into the Clarify File Share by through a stored procedure that creates an Attachment in Compass.



The Compass Server should be up and running for the MS Email Agent service to create email logs in the compass database. The stored procedures that are to be used, primarily two, also reside in the Compass Database. MS Email Agent is to open a

inserts the email logs/attachments.

accesses the stored procedures and

4.2.2 Workflow Introduction

Prior to this release all inbound Emails were processed by Email Clerk, which looks for specific tags in the Inbound Email and after successful validation creates logs against clarify cases or sub cases.

This CR was created to address the requirement of enhanced functionality of Inbound Email processing and handle Email attachments which is currently done with the help of attache32.dll in clarify GUI.

Also it is to be noted that, this CR is NOT intended to change the core Email Clerk processing functionality. Deployment of MSEmailAgent will eventually retire Email Clerk in EMEA and NA regions.

4.2.2.1 Assumptions and Approach

Below is a high-level description of how the CR1028 will be implemented.

1. MSEmailAgent is a new tool to replace Email Clerk
2. Will be deployed as a service polling a specific mailbox using WebDav http request.
3. Should be able to process Inbound Emails to the tune of 10,000 or more in a given day
4. Should be able to process Inbound Emails to the tune of 1000 per hour
5. Should notify customers after successful processing of Inbound Emails in a given format
6. On successful processing should create logs against cases in the clarify system

7. Process and Save Inbound Email Attachments to file shares and is accessible through the Clarify System
8. Provide User Interface to configure Options as currently exists in Email Clerk.
9. Store the above configured items encrypted in the registry of the machine where MSEmailAgent runs
10. Move emails which are not successfully processed to an ERROR folder in the mailbox from where the emails can be reviewed at a later time.
11. Provide option to delete or move (to SUCCESS folder) successfully processed Emails
12. Three levels of process logging available (Basic/Medium/Verbose)
13. All log files are created on the machine where MSEmailAgent runs and at a location as mentioned in the configuration screen
14. Log files names include date stamp. Format: MSEmailAgent_YYYYMMDD_HHMISS.log
15. A new text log file is created every day. Logs created by mails processed in a day are appended to the daily log.
16. A new log is generated each time the service is restarted.
17. The Start and Stop events of the service will be registered in the event log.
18. All errors will be logged in the event log as well as the log file.
19. No email alert will be sent from MSEmailAgent, external monitoring should handle all alerting
20. The MSEmailAgent service would read a given number (as mentioned in the configuration screen) of emails in one batch.
21. If a particular email for some reason is not moved or deleted as appropriate, the subject would be added to the "Don't Process Email with Subject" box and will need intervention from OPS. The same will also be logged in the event log.
22. Any sender identified as a "Mail Bomber" (Sending more that "Spam Alert" number of emails in the last "Spam Reset" number of emails would be added to the "Don't

Process Email From" box and [illegible] the name will also be logged in the event log.

23. All email sent out to customers are done so using web mail and would require a SMTP server to be specified in the Configuration screen.

4.2.3 Workflow

Configuration Settings

1. On successful start of MSEA Config application, the configured values stored in registry are read using Trustee.dll.
2. If no values are stored in registry then default values are set to Database name, Port Number, Folder Name, Logging Level, Process Count and Sleep time in sec, Spam Reset, Spam Alert, Max Log size and the user can have the same values or modify them.
3. User needs to enter values for all the mandatory fields in the MSEA Config application.
4. On clicking Ok Button or Apply Button the values entered in the UI are validated and then stored in registry using Trustee.dll.

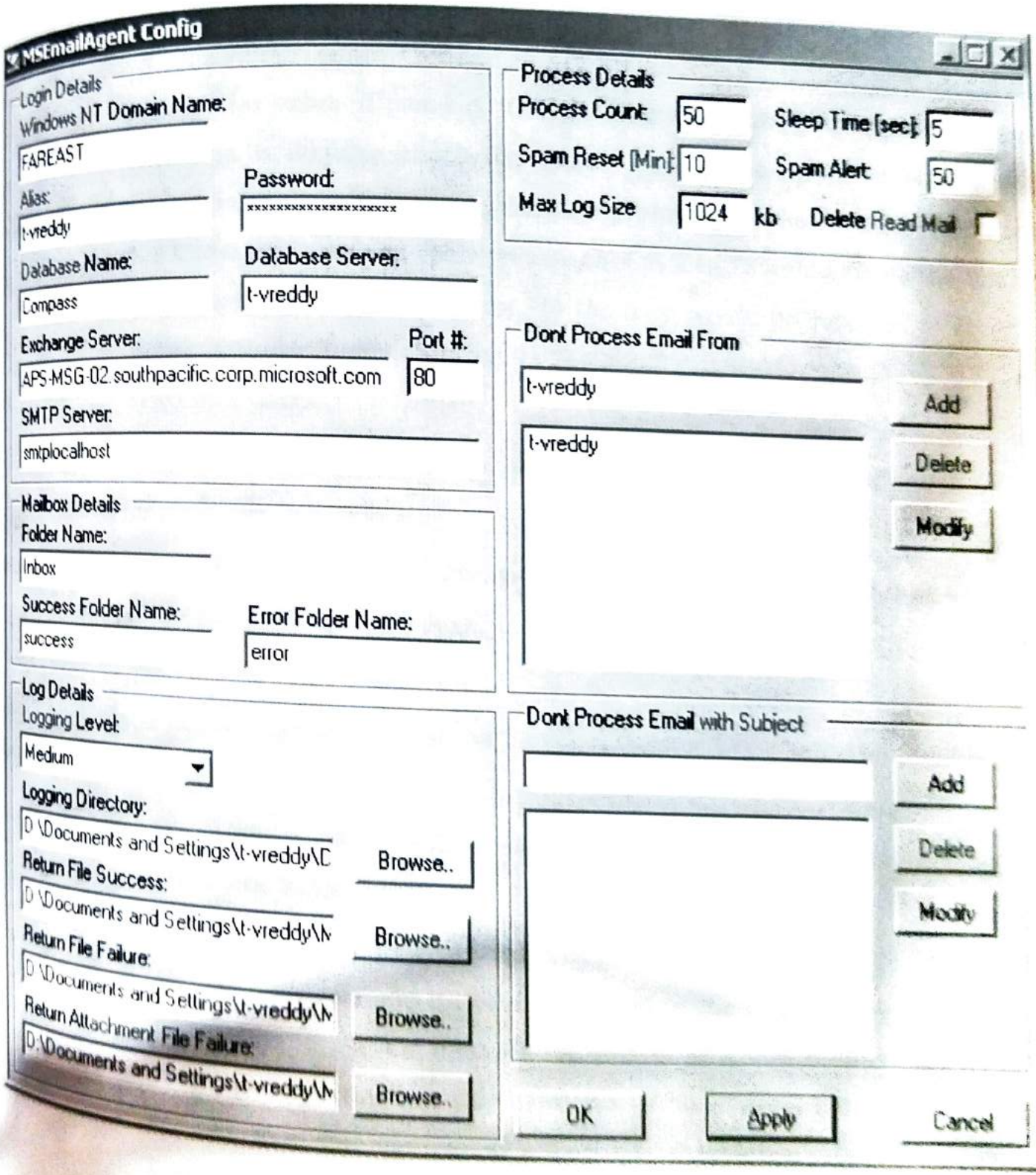
The Login Details section

The Login Details section contains NT Domain Name, Alias, Password, Database Name, Database Server, Exchange Server, Port Number and SMTP Server details.

i. User needs to enter a valid Windows NT Domain Name. Eg: Fareast
ii. User needs to enter a valid Alias. Eg: t-vreddy
iii. User needs to enter a valid Password. Eg: *****
iv. Default value "Compass" is set to the Database name. The user can modify the value if needed.
v. User needs to enter a valid Database Server name. Eg: tmp22
vi. User needs to enter a valid Exchange Server name. Eg: APS-MSG-01.southpacific.corp.microsoft.com

vii. Default value "80" is set to the Port Number. The user can modify the value if needed.

viii. User needs to enter a valid SMTP Server name. Eg: 65.52.10.82



The Mailbox Details section

The Mailbox Details section contains Folder Name, Success Folder Name and the Error Folder Name details.

- Default value "Inbox" is set to the Folder name. The user can modify the value if needed. On clicking Ok Button or Apply Button validation is done to check the Folder name. If the Folder name is not present in the Mailbox the application prompts the user whether to create this folder and user can select yes or no. On Yes, it would create the folder name given by the user and on No the user would be required to enter a different folder name. Similar is the functionality for the SUCCESS & ERROR folders.

- In the Folder Name, subfolder creation is not allowed. (E.g. Inbox/Mails is invalid)

- In the Success Folder and Error Folder Name creation only one sub folder creation is allowed. (E.g. Logs/Success is valid)

The Log Details section

The Log Details section contains Logging Level, Logging Directory and Return File details.

- User needs to select the Logging Level. The available Logging Levels are Basic, Medium and Verbose.

- User needs to enter the Logging Directory path. The User can either enter directly in the text box or select the Logging Directory using the browse button. On clicking Ok Button or Apply Button validation is done to check the Logging Director. If the Logging Directory is not present, then the application asks the user whether you need to create Logging Directory and user can say yes or no. On Yes, it will create the

Logging Directory given by the user and on No the user can enter a different Logging Directory.

- User needs to enter the Return File Success path. The User can either enter directly in the text box or select the Return File Success using the browse button. On clicking Ok Button or Apply Button validation is done to check the Return File Success. If Return File Success is not present, then the application pops up the appropriate error message. Then the user needs to enter the valid Return File Success path.

- User needs to enter the Return File Failure path. The User can either enter directly in the text box or select the Return File Failure using the browse button. On clicking Ok Button or Apply Button validation is done to check the Return File Failure. If Return File Failure is not present, then the application pops up the appropriate error message. Then the user needs to enter the valid Return File Failure path.

- User needs to enter the Return Attachment File Failure path. The User can either enter directly in the text box or select the Return Attachment File Failure using the browse button. On clicking Ok Button or Apply Button validation is done to check the Return Attachment File Failure. If Return Attachment File Failure is not present, then the application pops up the appropriate error message. Then the user needs to enter the valid Return Attachment File Failure path.

The Process Details section

The Process Details section contains Process Count, Sleep Time in seconds, Spam Reset, Spam Alert, Max Log size in kb and Delete Read Mail details.

- Default value "50" is set to the Process Count. The user can modify the value if needed. Only numeric value must be entered.

- Default value "5" is set to the Sleep Time in seconds. The user can modify the value if needed. Only numeric value must be entered.

- Default value "500" is set to the Spam Reset. The user can modify the value if needed. Only numeric value must be entered.
- Default value "50" is set to the Spam Alert. The user can modify the value if needed. Only numeric value must be entered.
- Default value "1024" is set to the Max Log size in kb. The user can modify the value if needed. Only numeric value must be entered.
- If the Delete Read Mail is checked, then the Success Folder name is cleared and then disabled.

The Don't Process Email From section

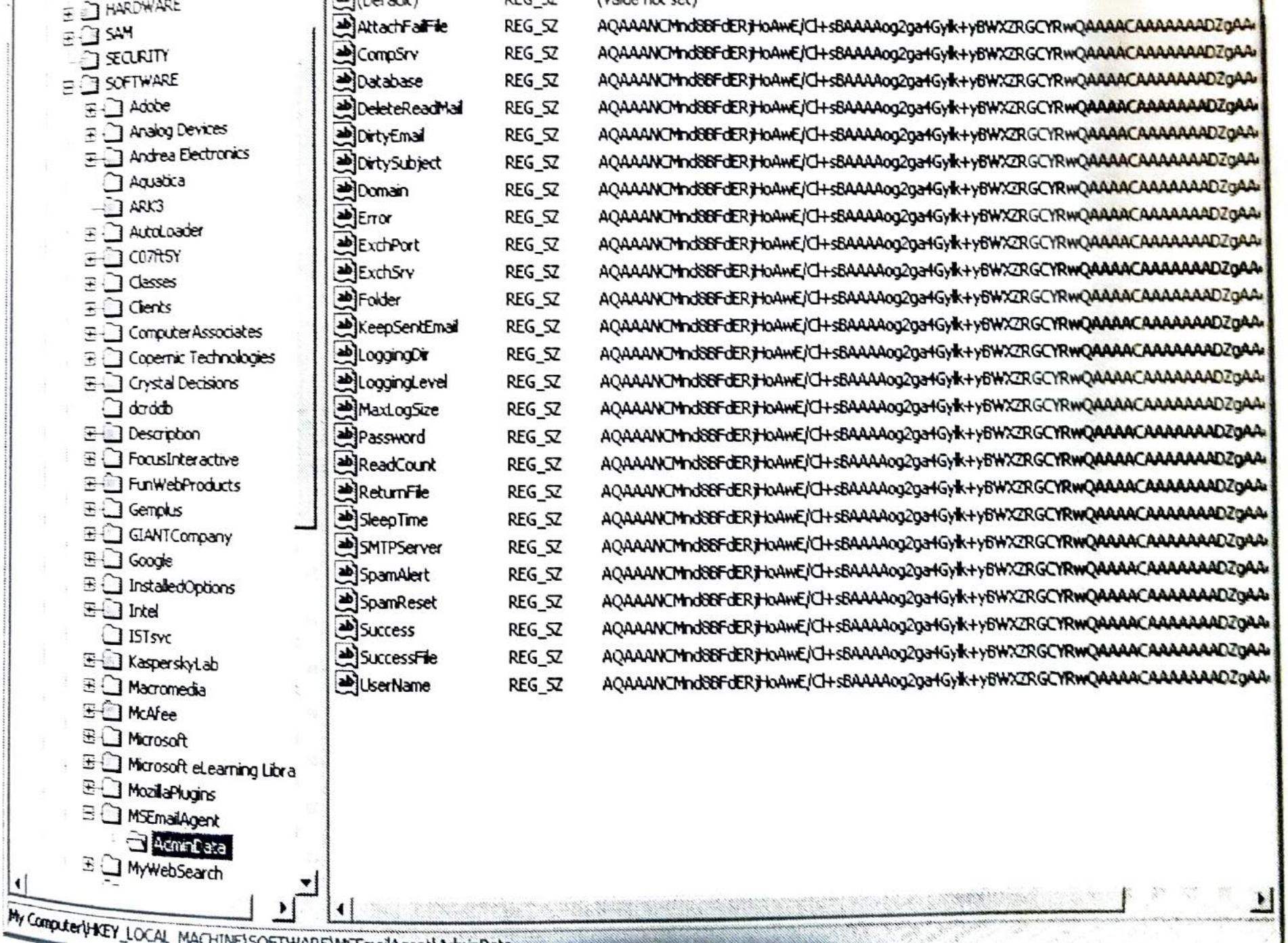
In the Don't Process Email From section the user can enter the email address that need not be processed by the Email Agent service.

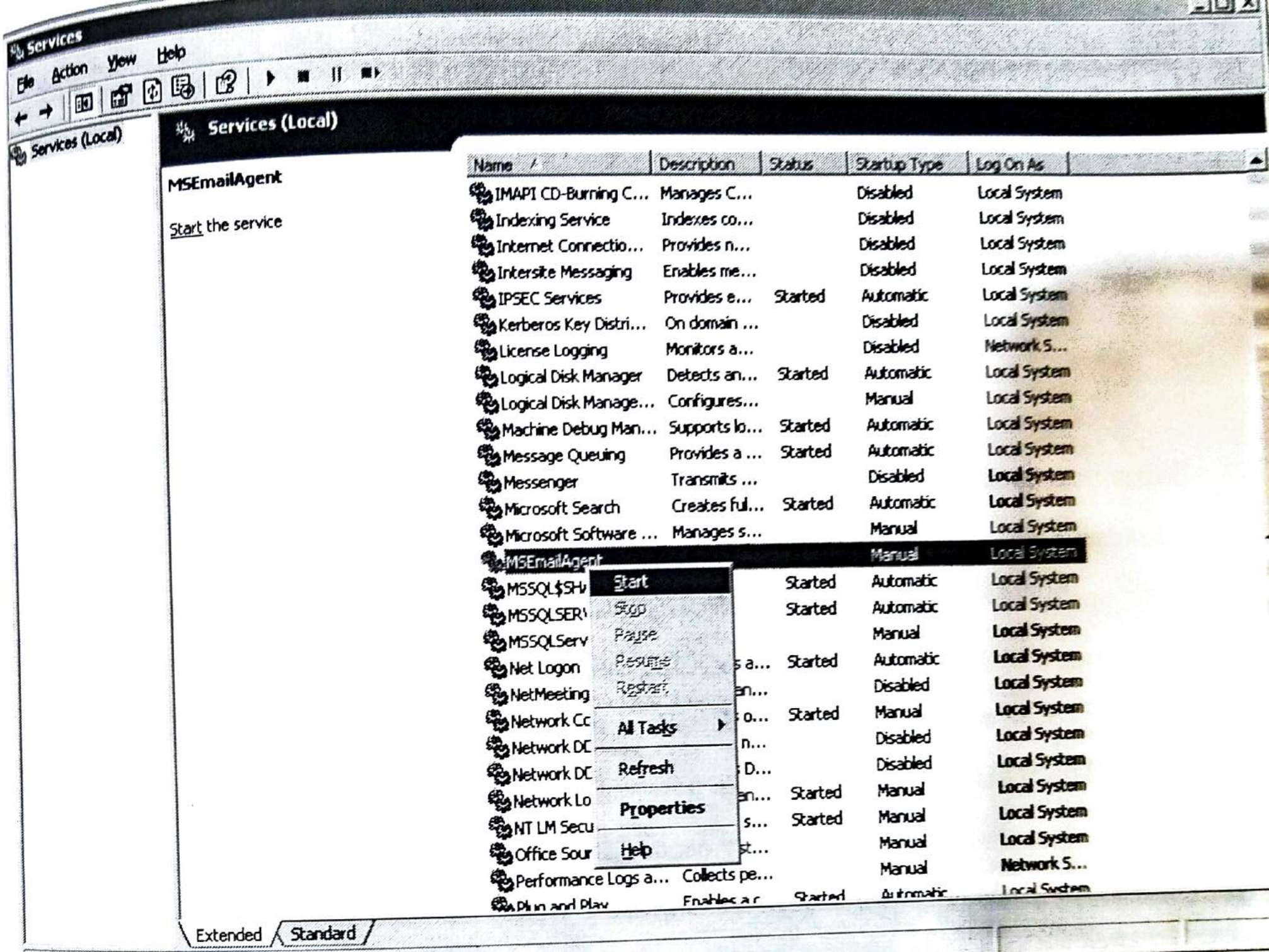
- To Add an email address, type the email address in the text box and press the add button.
- To Modify an existing email address, select the email address that needs to be modified from the given List and then type the new email address in the text box and press the modify button.
- To delete an existing email address, select the email address that needs to be deleted from the given List and then press the delete button.

The Don't Process Email with Subject section

In the Don't Process Email with Subject section the user can enter the subject names that are not needed to be processed by the Email Agent service.

- To Add a subject, type the subject in the text box and press the add button.
- To Modify an existing subject, select the subject name that needs to be modified from the given List and then type the new subject name in the text box and press the modify button.


HARDWARE
SAM
SECURITY
SOFTWARE
Adobe
Analog Devices
Andrea Electronics
Aquabica
ARK3
AutoLoader
C07ft5Y
Classes
Clients
ComputerAssociates
Copernic Technologies
Crystal Decisions
dcrddb
Description
FocusInteractive
FunWebProducts
Gemplus
GIANTCompany
Google
InstalledOptions
Intel
ISTsvc
KasperskyLab
Macromedia
McAfee
Microsoft
Microsoft eLearning Libra
MozillaPlugins
MSEmailAgent
AdminData
MyWebSearch
(Default) REG_SZ (value not set)
AttachFailFile REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
CompSrv REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
Database REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
DeleteReadMail REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
DirtyEmail REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
DirtySubject REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
Domain REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
Error REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
ExchPort REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
ExchSrv REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
Folder REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
KeepSentEmail REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
LoggingDir REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
LoggingLevel REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
MaxLogSize REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
Password REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
ReadCount REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
ReturnFile REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
SleepTime REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
SMTPServer REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
SpamAlert REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
SpamReset REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
Success REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
SuccessFile REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
UserName REG_SZ AQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAog2ga4Gylk+yBWXZRGCYRwQAAAACAAAAAAADZgAA
My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\MSEmailAgent\AdminData


Services
File Action View Help
Services (Local)
Services (Local)
MSEmailAgent
Start the service
Name
Description
Status
Startup Type
Log On As
IMAPI CD-Burning C... Manages C... Disabled Local System
Indexing Service Indexes co... Disabled Local System
Internet Connectio... Provides n... Disabled Local System
Intersite Messaging Enables me... Disabled Local System
IPSEC Services Provides e... Started Automatic Local System
Kerberos Key Distri... On domain ... Disabled Local System
License Logging Monitors a... Disabled Network S...
Logical Disk Manager Detects an... Started Automatic Local System
Logical Disk Manage... Configures... Manual Local System
Machine Debug Man... Supports lo... Started Automatic Local System
Message Queuing Provides a ... Started Automatic Local System
Messenger Transmits ... Disabled Local System
Microsoft Search Creates ful... Started Automatic Local System
Microsoft Software ... Manages s... Manual Local System
MSEmailAgent Manual Local System
MSSQL$SH Started Automatic Local System
MSSQLSER Started Automatic Local System
MSSQLServ Manual Local System
Net Logon Started Automatic Local System
NetMeeting Disabled Local System
Network Cc Started Manual Local System
Network DC Disabled Local System
Network DC Disabled Local System
Network Lo Started Manual Local System
NT LM Secu Started Manual Local System
Office Sour Manual Local System
Performance Logs a... Collects pe... Manual Network S...
Plug and Play Enables a c... Started Automatic Local System
Start
Stop
Pause
Resume
Restart
All Tasks
Refresh
Properties
Help
Extended
Standard

4.2.4 Differences from Existing EmailClerk System

1. MSEA does not require that a MS OUTLOOK is installed on the server.
2. MSEA polls the mailbox on the exchange server.
3. Client side outlook rules will not impact the MSEA processing. All such rule should be move to server side or be implemented using the "Don't process" features provided on the MSEA configuration utility.
4. The Success email is sent to the customer in the following format
   - **Subject** - Re: <Subject of processed email>
   - **Body** - Your Case email was accepted.

     Case_ID_NUM: SRL031114600083
5. The error email is sent to the customer in the following format
   - **Subject** - Re: <Subject of processed email>
   - **Body** - Your mailed case was rejected.

     Please refer to the following format:

     CASE_ID_NUM: SRXxxxxxx6xxxxx

     MESSAGE: Enter the text of your message here.

   NOTE: There should be a space after CASE_ID_NUM: & MESSAGE:
6. The Mailbox will need to have web access enabled
7. MSEA processes about 20 email a minute. (Might vary depending on the size of the email and attachments)
8. The MSEA configuration screen is different from the "Rule Manager" and "Logging" screens and cannot be integrated with them.
9. MSEA saves attachment to the local server before it uploads it to the Clarify File Share and eventually deletes the local copy.

4.3 Email Processing Workflow

MS Email Agent is implemented essentially as a Window Service. It issues requests to the exchange server, validates the responses, and makes sproc calls to submit the email data to the clarify email logs.

The entire email processing workflow can be categorized into four phases:

- Initialize Phase
- Request-Response Phase
- Message Read & Append Phase
- Attachment Validation and Upload Phase

Phase I: Initialize Phase

In this phase, MS Email Agent service is started. The first activity is to read and validate the configuration values. Then a link (i.e. a Uniform Resource Identifier) is created to the mailbox that the MS Email Agent service polls.

Phase II: Request – Response Phase

The mails in the mailbox are retrieved through the issue of a HTTP Web Request. A corresponding Response sends the data back to the MS Email Agent service. In case an error occurs during processing, it is logged and the service sleeps for a fixed amount of time before restarting.

Phase III: Message Read & Append Phase

Once data in the proper format is received from the Exchange Server, MS Email Agent service opens an SQL Connection. Then, programmatically, logs are appended into the database, with the help of a counter variable I. Email count is a variable which is initialized with the value of the number of unread messages. A loop is created, wherein the variable I keeps growing in steps of 1, till it reaches the value of Email count, and then the connection is closed.

For a valid value of I, i.e. while I is less than the value of Email count, that particular email message is read, and its subject and body are obtained. If this is an error-free process, it is a success, and the email log is created in the Compass Database against the Case. In case there is a failure, there can be two reasons:

1. Subject/Body are in improper format/unrecognized format
2. Subject/Body are in proper format, but the Case ID mentioned does not exist

In the former case, nothing is done, but for logging the error and moving the email to the error folder. In the latter case, because the Case ID is invalid, the customer is notified that the Case ID is invalid, and needs to resend the email with the appropriate Case ID.

If the message is without any error, and the Case ID is valid, a stored procedure is called, to create an Email Log in Compass Database.

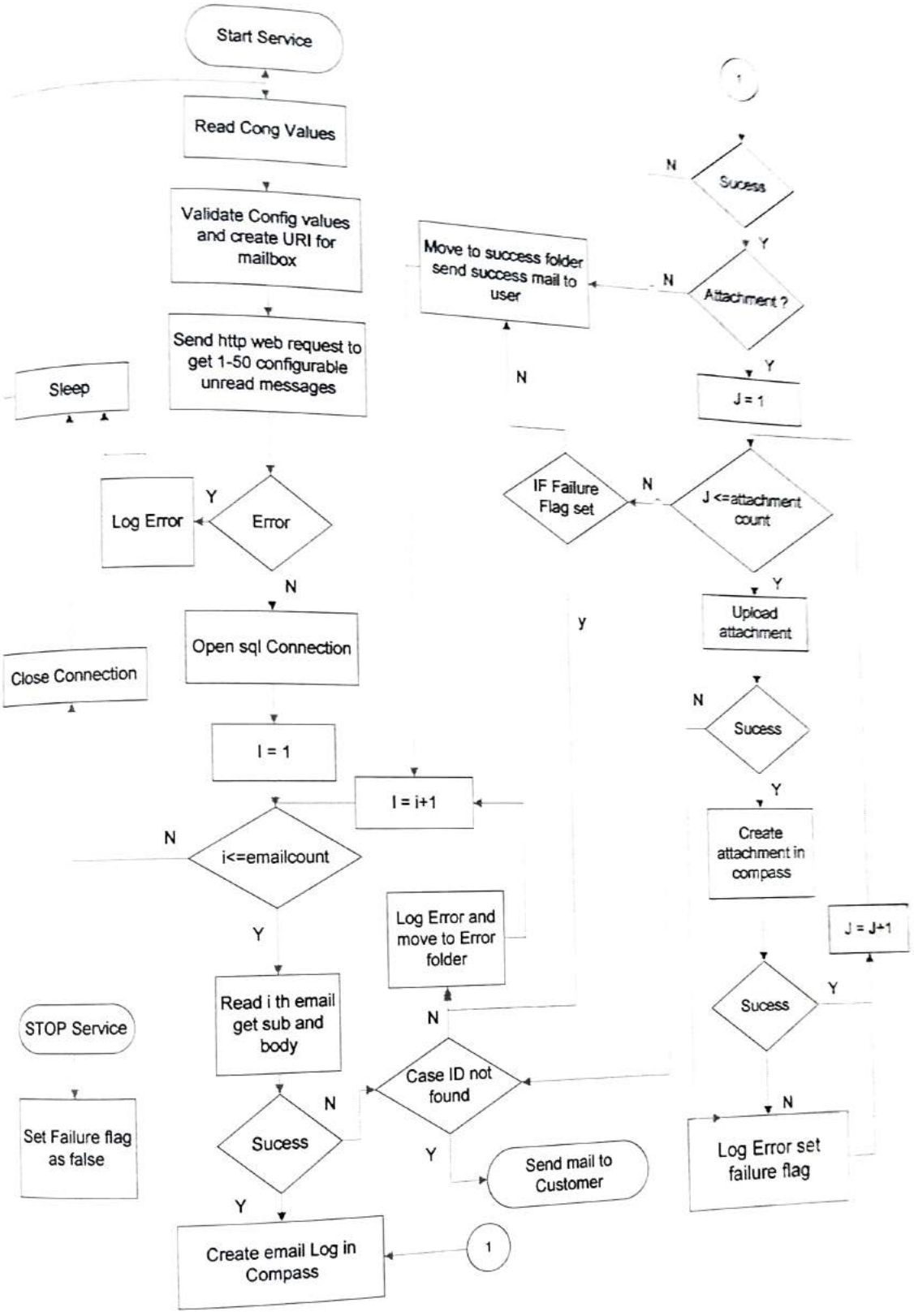
Phase IV: Attachment Validation & Upload Phase

Once the message processing is successful and the email log is created, then the presence of an attachment is checked. A variable called Attachment count keeps track of the number of attachments that have arrived with a particular email. A temporary variable, J, runs through increments of 1, beginning from 1 till a maximum value of Attachment count. As long as J value is less than the Attachment count, it implies there is a valid attachment, and it may be uploaded.

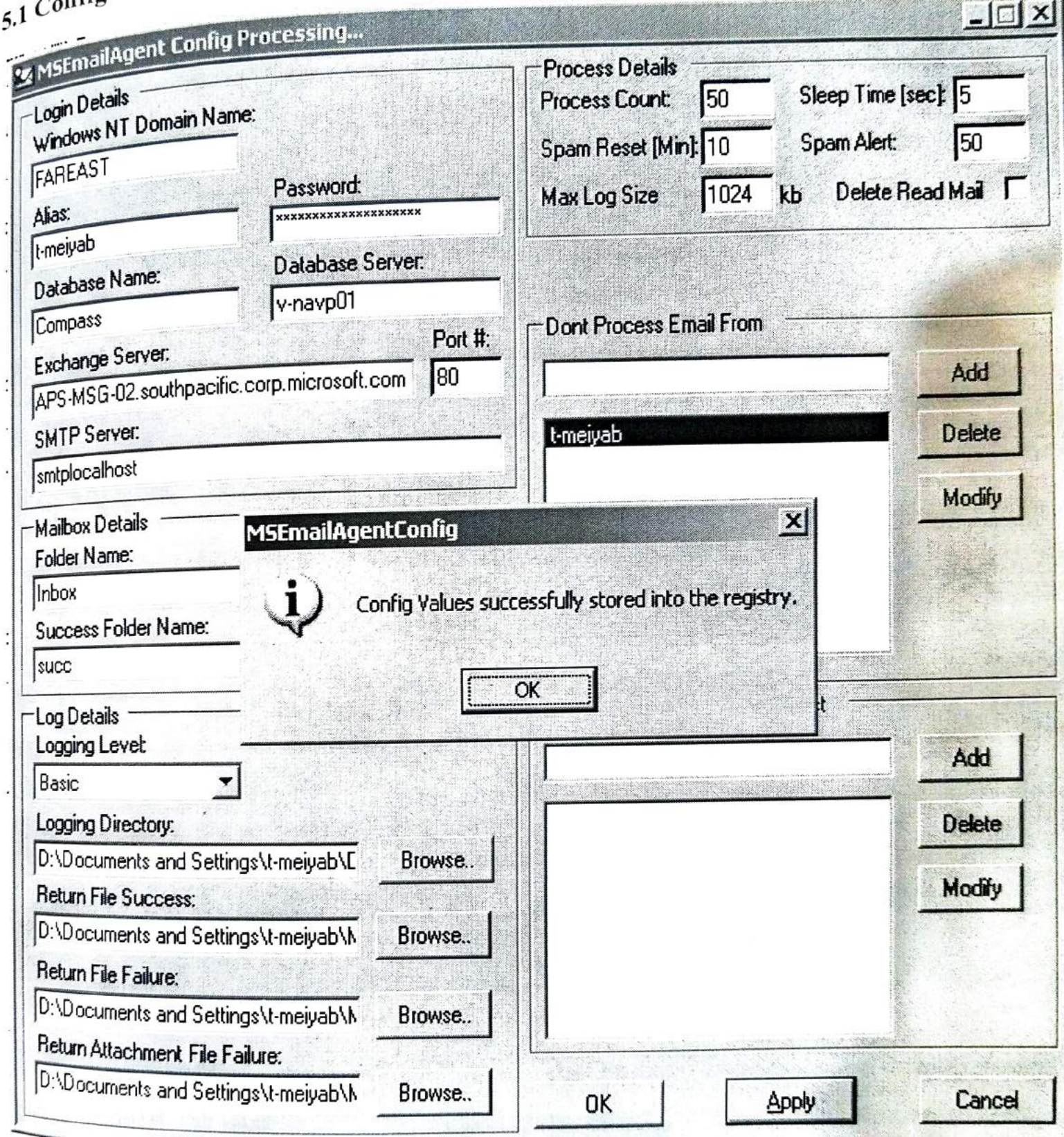
If the Attachment is successfully uploaded, then it is sent to the Clarify File Share, where a repository of other attachments exists. If that upload is also successful, then the variable J is incremented by 1. This increment follows that the file has finally been successfully processed. In case there is an error in the file upload, or the file contents are not readable for any reason, an error is logged, and the failure flag is set.

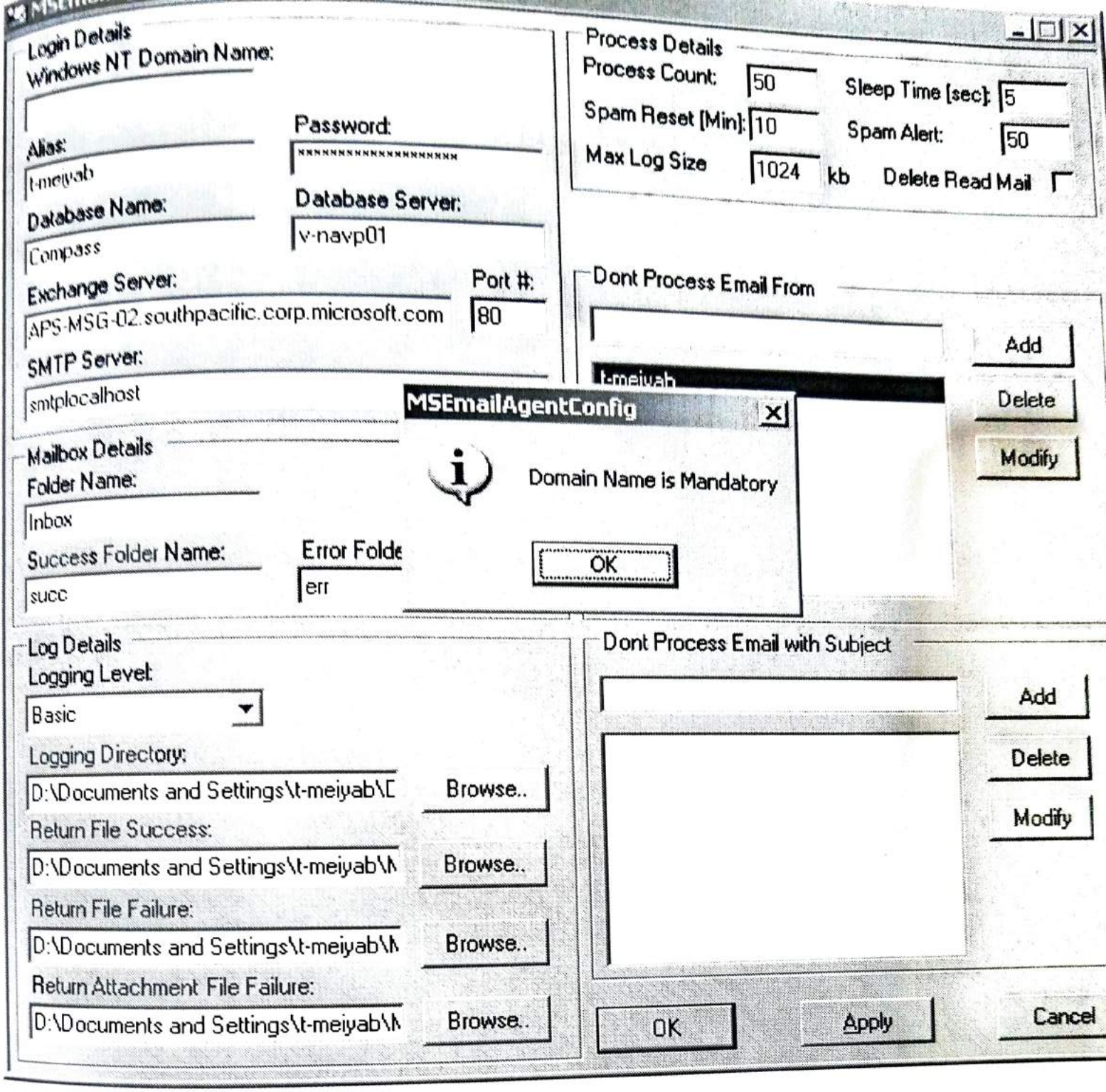
Thus, this process occurs at repeated intervals of time, when MS Email Agent polls the Exchange Server at specific intervals. Two important stored procedures are used. Sp_CreateMSEmailLog Stored Procedure is used to create an email log in Compass. The other, Sp_CreateMSInboundAttachment Stored Procedure is used to create an attachment in Compass.

4.3.1 Email Processing Workflow Diagram

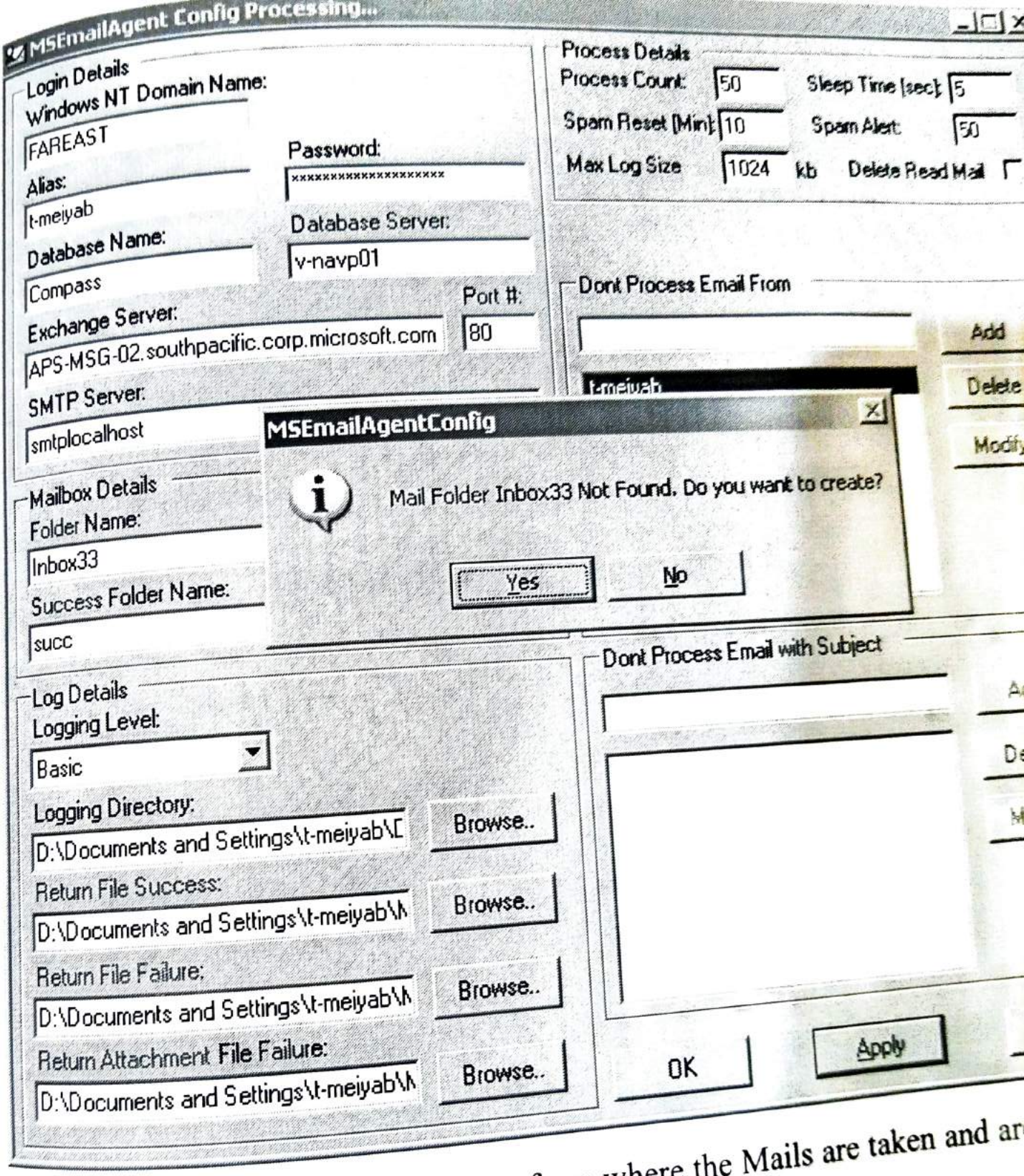


SCREENSHOTS


MSEmailAgent Config Processing...
Login Details
Windows NT Domain Name:
FAREAST
Alias:
t-meiyab
Password:
Database Name:
Compass
Database Server:
v-navp01
Exchange Server:
APS-MSG-02.southpacific.corp.microsoft.com
Port #:
80
SMTP Server:
smtplocalhost
Process Details
Process Count:
50
Sleep Time [sec]:
5
Spam Reset [Min]:
10
Spam Alert:
50
Max Log Size
1024
kb
Delete Read Mail
Dont Process Email From
t-meiyab
Add
Delete
Modify
Mailbox Details
Folder Name:
Inbox
Success Folder Name:
succ
MSEmailAgentConfig
Config Values successfully stored into the registry.
OK
Log Details
Logging Level:
Basic
Logging Directory:
D:\Documents and Settings\t-meiyab\D
Browse..
Return File Success:
D:\Documents and Settings\t-meiyab\M
Browse..
Return File Failure:
D:\Documents and Settings\t-meiyab\M
Browse..
Return Attachment File Failure:
D:\Documents and Settings\t-meiyab\M
Browse..
Add
Delete
Modify
OK
Apply
Cancel



If the Domain is not mentioned we see that an error box would be thrown and would intimate the user to enter the region from where the user is authenticated to lo The region or the domain has various purposes to serve. This would also be linked to active directories which would enable the exchange server to authorize the user.



The Mail Folder is the location from where the Mails are taken and ar accordingly. Here when the user enters a wrong or not existing mail folder a the exchange server would ask for creating a new account and can be handl also various other limitations which are placed on the exchange server base and the creation limitations of the required folder accounts.

TESTING

6.0 Testing

Testing involves operation of a system or application under controlled conditions and evaluating the results. Testing is the one step in software engineering process that could be viewed as destructive rather constructive. Testing requires that the developer discarded preconceived notions of the "Correctness" of the software just developed and overcome a conflict of interest that occurs when errors are uncovered.

If the testing is conducted unsuccessfully, it uncovers errors in the software. As a secondary benefit, testing demonstrate that software function appear to be working according to specification, that performance requirements appear to have been met. In addition, data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole.

Software Quality Assurance involves the entire software development PROCESS - monitoring and improving the process, making sure that any agreed-upon standards and procedures are followed, and ensuring that problems are found and dealt with. It is oriented to 'prevention'

Testing cannot show the absence of defects, it can only show that software defects are present.

6.0.1 Objective:

The objective of testing is:
Testing is a major consideration in software development and maintenance today.
Testing is a process of executing a program with the intent of finding an error. A good test case is one of that has a high probability with the intent of finding an undiscovered error. A successful test is one that uncovers as undiscovered.

has been tested successfully using few of the following
strategies and errors that were encountered were corrected and again the part of the
program or the procedure or function is put to testing until all the errors were removed.

6.0.2 Testing Performed:

Testing is critical aspect that is done to ensure that the software according to the expectations. Testing is a process that is done with intent to find an error that is not yet discovered. There are two types of testing techniques followed in designing and development of the information system. They are:

Black Box Testing:

The system is tested as a whole without considering the internal processes and their inputs and outputs. The testing results conclude that the system is taking the inputs correctly as specified and giving the correct outputs for the inputs.

White Box Testing:

It is a technique that is done to verify internal processes, their inputs and outputs. Here, a test case is designed so as to check for all the possible cases. The system is tested using the test cases designed. Each data entry form executed separately and the results are compared with the present system results. Each field is validated to ensure that the data being captured is with in the valid range. It is found that all the functions are working satisfactorily according to the specifications. The controls are checked. A separate test case is designed to check the control, which passes control to all the forms generated, and the controls are found to be working satisfactorily.

Basic Path Testing:

This enables the test designer to derive a logical complexity measure
and use this measure as a guide for defining a basic

User Interface testing:

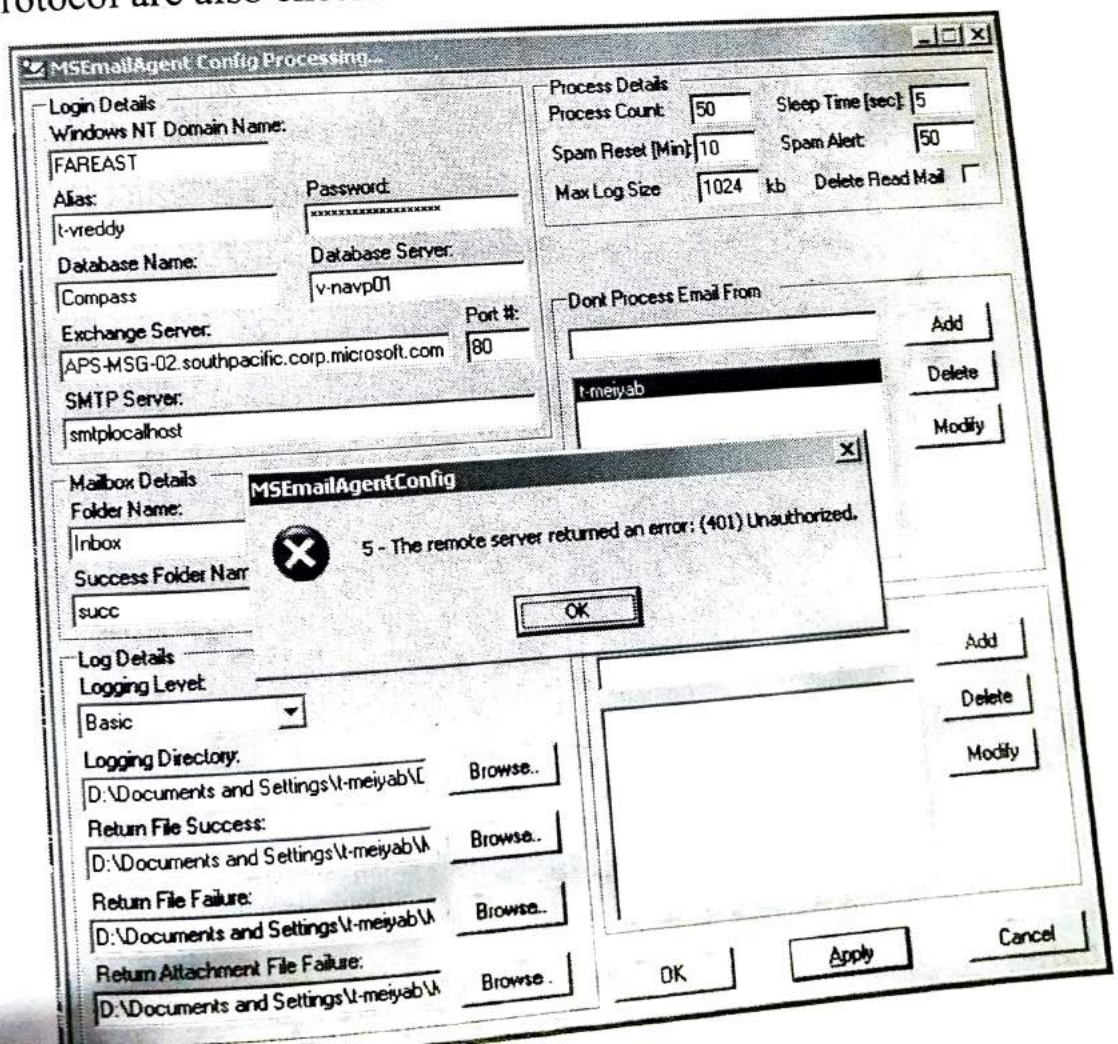
Software testing is one element of a broader topic that is often referred to as verification and validation. Verification refers to a set of activities that ensure that software correctly implements a specific function. Validation refers to a different set of activities that ensure the software that has been built, is traceable to customer requirement.

Unit Testing:

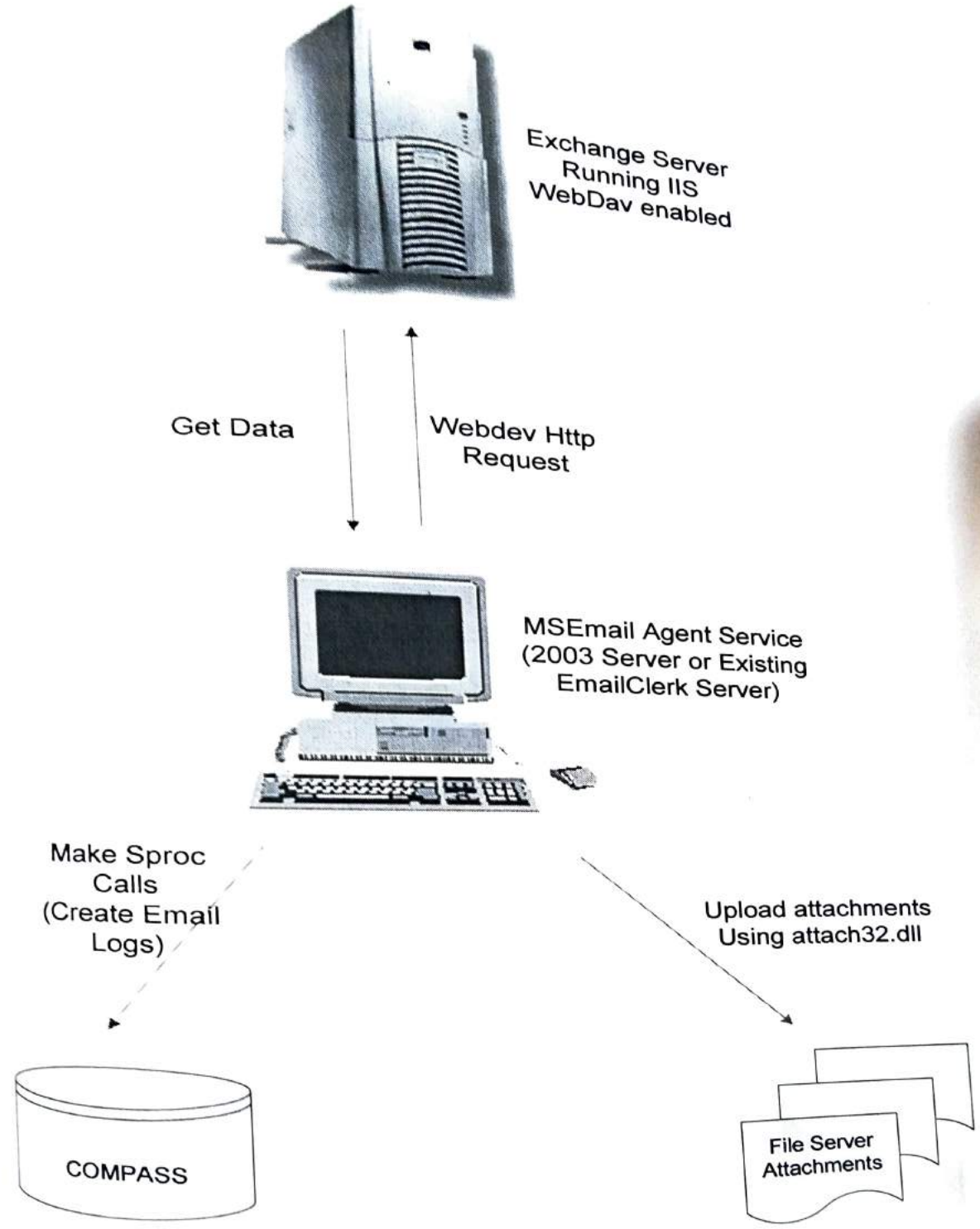
All units, which were developed for the sent and received part, are tested. If any invalid is seen then an error message gets fired.

Condition Testing:

Here we performed the user credentials were valid or not. If the user's password does not match with the original password it would through errors. Role based security was implemented in this model. We check that whether all the required path locations are entered before extracting mails from the exchange server. The conditions required for the webdav protocol are also checked.



3 Deployment Diagram



CONCLUSION

8. Expandability

An integrated CRM application can be developed with all the functionalities for managing the customer requests.

9. Conclusion

This product has taken care of the process, which are tedious and time consuming. The product can be customized to meet all the requirements of the organization and also the support engineers. This can be scalable and also can be made a part of the bigger customer relationship management tool which handles all the features of web, phone and email interactions.

This tool had incorporated the functionalities of three tools in the Email Clerk and made them as one and also greatly solved the attachment handling feature which was not provided by the Email Clerk.

10. Bibliography

The following resources have provided the necessary help in the development of the project and also in carrying out the documentation.

- Francesco Balena, Microsoft Press, "**Programming Microsoft VisualBasic.NET 2003**".
- Juval Lowy, O' Reilly, "**Programming .NET Components**".
- Brian Patterson, William Sempf, Richard Conway, Robin Dewson, Wrox Press , "**Visual Basic .NET Windows Services Handbook**"

The following websites also greatly contributed to this project and documentation.

- [Microsoft02-1] Microsoft Corporation. "XML Web Services Overview." *.NET Framework Developer's Guide*. Available from the MSDN Library at: http://msdn.microsoft.com/library/default.asp?url=/library/enus/cpgu de/html/cpconwebservicesoverview.asp.

- [PnP02] *patterns & practices*, Microsoft Corporation. "Application Architecture for .NET: Designing Applications and Services." *MSL Library*. Available at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/distapp.asp.

- Http://www.msdn.com

- Http://www.microsoft.com/msf