DB Mail™
for Oracle, Microsoft SQL Server, Sybase ASE, and IBM DB2

Introduction

E-business is fundamentally changing the way companies operate, going far beyond buying and selling over the Internet, or e-commerce. E-business enables businesses to leverage the use of technology to gain an advantage in the marketplace. With this technology, businesses can expand their customer base, provide faster, more efficient services, and develop more personalized services. In addition, the overhead associated with implementing business processes can be minimized or reduced by leveraging e-business technology to streamline interactions with customers, suppliers and business partners. Most modern e-business solutions are database-driven. Powerful relational databases systems play a central role in storing e-business data and automating e-business services. However, most commonly used database systems do not come inbuilt with all the features required for a successful e-business implementation. A lot of customization, tweaking and programming is required to fully integrate and automate these systems.

DB Mail provides an easy and efficient method for e-business enabling many database systems. With minimal effort, it allows you to automate such common e-business functions as sending electronic messages including e-mail and e-fax messages, pager and mobile phone messages (SMS messages), interruptible network messages and administrative alerts, and voice messages. And best of all, all these messages can be sent directly from your databases. Any application capable of connecting to your databases can also use DB Mail services with minimal programming.

DB Mail is also very easy to install, configure and use. Using DB Mail developers can easily add messaging functions to their homegrown applications. Database Administrators can automate alert functions for various database events and monitoring processes. For example, an application can send an alert whenever the inventory drops below a previously defined critical level, or an office management application can send an automated message to all users reminding them to submit attendance timesheets.

Sample Scenarios

1. On-line order processing application

A web-based Order Processing system accepts user orders and inserts every accepted order into an "order" table stored in the back-end Order Processing database. The INSERT operation fires a database trigger created for the INSERT operation on the "order" table. The trigger generates invoice data in a ready for faxing format and writes a message to the local message queue table. The database returns control back to the web portion of the application which then displays a confirmation message on the user's screen. This entire processing takes only a fraction of a second, as the application does not have to wait for the relatively slow fax operation to complete before getting the workflow control back and displaying the order confirmation.

On the other side of the local DB Mail queue, the asynchronous DB Mail daemon picks up the message and transfers it to the central message queue where it is validated and processed by the DB Mail message processor. The DB Mail message processor then delivers the message.

2. Supply chain warehouse application

A Manufacturing system supplies finished goods to warehouses. Requests for finished goods may be fulfilled by the Manufacturer by supplying from internal stock or, if the required quantity is not available, by scheduling a production run. Since there could be a considerable time delay between receiving the order and informing the warehouse of shipment of goods, an asynchronous processing model is used. This allows a warehouse to proceed on other business, and allows the Manufacturer to callback to the Warehouse once the order has been fulfilled. In the event the order has been fulfilled the warehouse application calls DB Mail stored procedure that in turn writes callback message to the message queue.

As before, on the other side of the local DB Mail queue, the asynchronous DB Mail daemon picks up the message and transfers it to the central message queue where it is validated and processed by the DB Mail message processor. The DB Mail message processor then delivers the message.

3. Database administrative applications

An Oracle administrative application implemented as a database stored procedure runs periodically in an unattended mode on the database server. This procedure checks critical database metrics and invokes an administrative alert whenever a metric falls below certain threshold. Oracle's DBMS_JOB package is used to schedule and run the job. In order to send the alert the stored procedure calls SEND_ALERT function from DB_MAIL package. For critical conditions, the procedure also sends a pager message to the seniors DBAs by calling SEND_PAGE function. Both SEND_ALERT and SEND_PAGE functions write corresponding messages to the message queue. These messages are then picked up by DB Mail database daemon and get delivered to system administrators as interruptible administrative alerts that are displayed on the administrator's workstations and also as alphanumeric pager messages.

A Microsoft SQL Server application implemented as a database stored procedure runs periodically in an unattended mode in the MASTER database. This procedure checks all databases residing on the SQL Server instance for LOG FULL conditions. If a LOG FULL condition has been detected the procedure calls the SENPOPUPMESSAGE stored procedure which in turn writes new message to DB Mail message queue. This message is then picked up by DB Mail database daemon and gets broadcasted to all users notifying them that a database become unavailable and user applications accessing that database may become frozen. The message also tells users that their application will resume normal processing after database administrators truncate the filled log.

4. Appointment-reminder application

An automated application running every morning in a healthcare provider office checks in the local database for appointments scheduled on the same day. For every found appointment it then retrieves patient's name and phone number. It then calls DB Mail's SEND_VOICE function from the DB_MAIL package in order to send a pre-recorded message reminding the patient about the appointment time. The actual appointment time is pronounced using a computer generated voice. This entire processing takes only a few seconds, as the application does not have to wait for the relatively slow phone dialing operation to complete before getting the workflow control back.

On the other side of the local DB Mail queue, the asynchronous DB Mail daemon picks up all submitted messages and transfers them to the central message queue where they are validated and processed by the DB Mail message processor. The DB Mail message processor then communicates to the configured VoMS server which delivers the submitted messages to all patients.

Supported messaging methods and protocols

DB Mail is a complete out-of-the-box messaging solution, which is easy to customize and does not require installation of any additional software or services on the database server and network other then included DB Mail components. DB Mail uses only standard database communication network protocols, which allow it to support any Oracle, DB2, Sybase and Microsoft SQL Server database systems running on any platform.

Sending electronic mail

DB Mail supports several email protocols and interfaces, including:

- Windows Messaging Application Interface (MAPI)
- Simple Mail Transfer Protocol (SMTP)
- Lotus Notes (HTAPI)

All supported protocols allow sending plain text messages with and without attachments. In addition, DB Mail's SMTP protocol implementation supports sending rich-text format messages, HTML messages and XML messages. Such message formats provide greater control over email message appearance. They can also include in-line images, tables, formats and other graphical elements. Up to 255 external files or contents of database BLOB (Binary Large Object) columns can be attached to a single email message.

Sending SMS messages to pagers and cell phones

DB Mail uses Simple Network Paging Protocol (SNPP) to send SMS messages through the Internet. DB Mail implements Simple Network Paging Protocol Version 2 as specified in RFC 1861. In the case of a

numeric message the number of digits received by the recipients is usually limited to 8 to 12. In the case of an alphanumeric message, the number of characters received by the recipients depends on how they set up their server preferences and usually is limited to 240 characters.

Sending interruptible network popup messages

DB Mail uses Windows network messaging which allows sending arbitrary messages to a registered message alias such as network computer name or user name. The message appears on the destination computer as an interruptible popup message box. The user is required to click the OK button displayed on the message box in order to close the message and continue working. Because the size of the message box is somewhat limited to the computer screen, brief network messages are usually used to send important notifications or alerts.

DB Mail uses NetMessage protocol supported on Windows NT/2000/XP/2003 systems.

Sending interruptible system alerts

Sending interruptible system alerts is similar to sending network popup messages described in the previous topic. The only real difference is that network popup messages are sent to named recipients while administrative alerts are sent to a group of alert subscribers, which usually includes network administrators and other technical personal. Obviously the second method is more appropriate for sending administrative messages.

DB Mail alert functions rely on Windows NT LAN Manager protocol and more specifically on Windows NT Alerter service available on NT/2000/XP/2003 systems. Users running Windows 95/98/Me systems are unable to receive administrative alerts.

Sending electronic faxes

DB Mail supports sending outbound electronic faxes. DB Mail uses Microsoft Windows ® Fax API available in Windows 2000 and later to create and transmit faxes. Here is how this is done.

You use DB Mail stored procedures in your database to dynamically create text, HTML or XML documents. DB Mail then opens each document and converts it to a FAX-compatible TIFF image. After this conversion takes place, a cover page is optionally added and the fax is handed off to the digital modem. The modem dials the recipient fax telephone number, listens for a fax signal, and sends the fax to the recipient's fax machine. After the fax has been sent successfully, DB Mail optionally sends a confirmation to the sender via e-mail. If the fax transmission fails after three or more attempts, a failure notification is optionally sent to the sender via e-mail. This failure notification indicates whether the fax failed because of a busy signal, no answer, or the telephone number dialed was not a fax number.

The actual modem and fax transmission can be performed on the computer running DB Mail Console or on any other Windows 2000 or later computer connected to the same network and featuring internal or external Class II digital fax-modem. Multi-modem digital adapters can be also used for high performance fax processing. Such multi-modem adapters are capable of sending several fax messages simultaneously.

Note: DB Mail implements advanced message queuing, which enables true high-performance asynchronous database processing, as the performance is not affected by the performance of the fax processing. Database applications simply call DB Mail interface functions that quickly write messages to the message queue and immediately free applications making it unnecessary for the applications to wait for the slower fax processing.

Sending voice messages

DB Mail supports sending pre-recorded and dynamically generated voice messages using advanced text-to-speech technologies. For dynamic text-to-speech messages DB Mail uses Speech API available in Windows 2000 and later to generate dynamic sound files. It then uses the VoMS (SoftTree Technologies' Voice Message Server™) software to deliver the messages over a regular phone line as an automated phone call. Below is the description of this processing workflow.

You use DB Mail stored procedures in your database to send voice messages. Messages can contain references to pre-recorded sound files or simply contain text that you want to say upon successful phone call. Messages can be also assembled from multiple parts containing both sound files and text segments. For every sent message DB Mail message queue processor picks up the message description, voice and text parts and submits them to the VoMS server that can be running on the same or different computer on your network. The VoMS server in turn converts all text segments to intermediate sound files and merges all referenced sound files into one single message. The resulting message is then handed off to the VoMS message processing engine that uses either a local modem or Intel Dialogic board to dial the recipient's phone number, listen for a human voice answer, and then speak the message. If the processing fails, for example, there is no modem or phone line available or and answering human voice has not been detected after three or more attempts, a failure notification is optionally sent to the message sender via e-mail. This failure notification indicates whether the phone call failed because of a busy signal, no answer, call hang-up before completion, or the dialed number is not in service.

The actual phone call could be made from the computer running DB Mail Server or any other Windows 2000 or later computer connected to the same network and running VoMS software. The computer must have a sound card and internal or external voice modem or Intel Dialogic phone board.

Note: DB Mail implements advanced message queuing, which enables true high-performance asynchronous database processing, as the performance is not affected by the performance of the voice call processing. Database applications simply call DB Mail interface functions that quickly write messages to the message queue and immediately free applications making it unnecessary for the applications to wait for the slower voice call processing.

Message Processing

Applications that use asynchronous operations (where more than a single operation can execute simultaneously) often have better availability, reliability, scalability, and usability than applications using only synchronous operations (where operations execute sequentially). DB Mail is designed with the goal of supporting asynchronous message processing, which allows database applications to write messages to a message queue and be immediately freed to serve other user requests.

After messages are written to the message queue, asynchronous DB Mail database daemon processes pick up queued messages and transfer them to the central message queue where messages are processed according to their priorities and timing.

Because DB Mail is designed as a multi-threaded application it can be configured to simultaneously serve multiple databases servers. DB Mail runs a separate daemon process for every configured database connection. In case one of the connected databases goes down, it does not stop other daemon processes connected to other databases. Each daemon process checks periodically for messages written to message pipe in the connected database and transfers new message to the central message queue. DB Mail also runs another asynchronous process called Message Queue Processor whose purpose is process messages in the central message queue. The Message Queue Processor verifies message data and delivers messages to the recipients. The actual delivery method varies for different message types.

The following diagram demonstrates message-processing workflow.
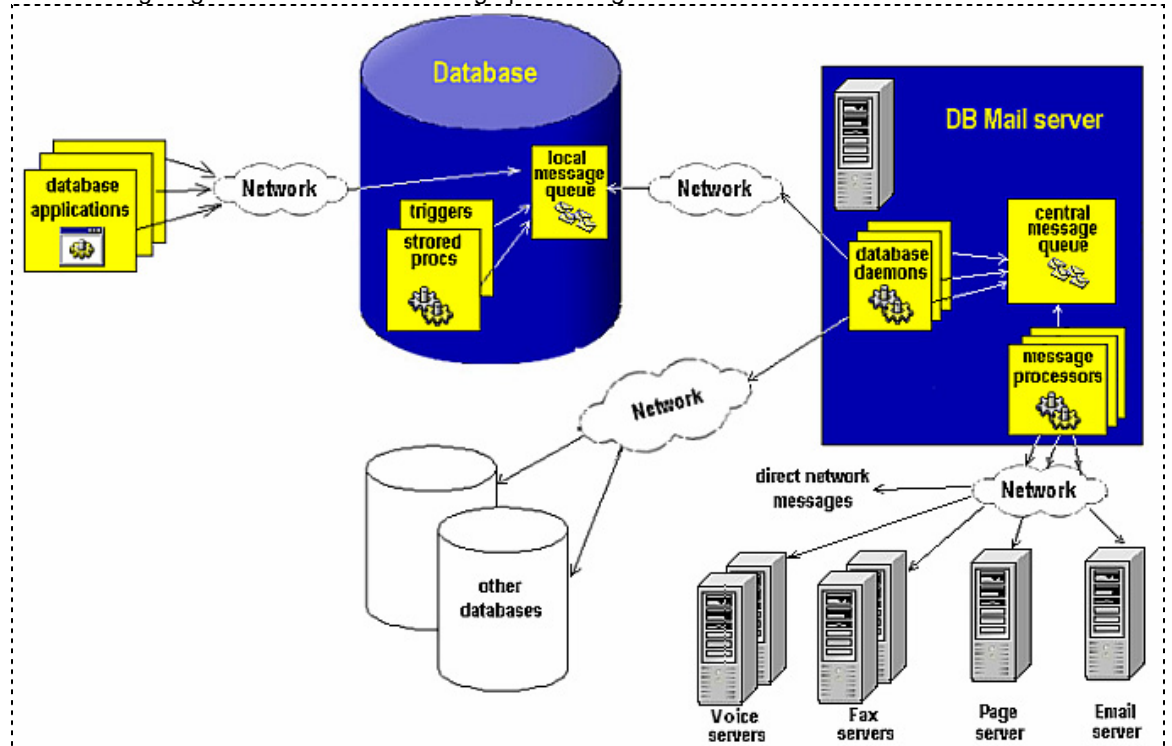
Database applications

There are essentially 3 types of database applications:

- Internal applications that use stored code run entirely within the space of the database server. An example of internal application could be a standalone database stored procedure designed to periodically produce and then email some business report. Such procedures can be scheduled using one of the available database job scheduling methods.
- External applications have their code stored completely outside of the database. An example of such application could be a Visual Basic program (or other type of interactive or automatic program) that is run on a user's workstation. When executing, this program connects to the database, performs some data manipulations and then disconnects from the database.
- Mixed applications have some portions of their code stored and run in the database space while other parts of their code are stored and run outside the database. For example a Visual Basic application described in the previous paragraph could insert new records into an order entry table. The database portion of this application could be implemented as a database trigger for the order entry table, which is executed by the database in the event of new records being inserted into the table. Such trigger can perform some business data validation and email sales managers in case inventory level falls below a certain level. The same trigger could also create customer invoices for the inserted orders and automatically fax them to customers.

Database server

Database management systems containing local DB Mail queue and messaging interface procedures.

Local message queue

Database local message queue is used to temporarily store pending messages. In most systems database queue is implemented as a regular database table. Database applications normally write to the message queue using DB Mail interface functions and procedures.

Database daemons

Daemon processes run on the DB Mail server computer. Daemon processors monitor database-side message queues and transfer new messages to the DB Mail server.

Central message queue

Central message queue is the file based message queue where database daemons drop transferred messages.

Message processors

Message Processors are special processes that run on the DB Mail server computer. They are responsible for processing queued messages and delivering messages to specified recipients.

Email server

The email server such as Microsoft Exchange or Lotus Notes is used to send electronic mail. DB Mail server contains internal email client component, which it uses to connect and communicate to your corporate email server and send email messages. The connection and communication method differs for different email systems. DB Mail supports 3 most popular email interfaces: SMTP, Windows MAPI and Lotus Notes.

Page server

Page server is used to send SMS messages and alpha-numeric pages to cell phones and pagers using Simple Network Paging Protocol (SNPP). DB Mail server contains an internal SNPP client component, which is used to send SMS messages. It implements Simple Network Paging Protocol Version 2 as specified in RFC 1861. In order to deliver messages SNPP component connects and communicates to a SNPP server through the Internet. Most telephone service carriers provide SNPP servers to its users free of charge.

Fax server

Fax server is used to transmit electronic faxes. DB Mail server contains internal file-to-FAX converter and fax client components, which are used to print attached files to a Microsoft FAX print driver. The FAX driver converts files to fax-compatible TIFF images. The resulting TIFF images are then sent to a Windows 2000 or Windows XP computer connected to a fax-modem hooked to a phone line. Because all Windows 2000 and Windows XP computers feature built-in fax server software, any Windows 2000 or Windows XP computer can be used as a fax server.

Note: Fax transmissions are relatively slow operations. DB Mail provides built-in support for fax processing scaling out strategy (scaling out is the strategy that increases the capacity of an infrastructure tier to handle load by adding servers, thereby increasing the aggregate capacity of those servers). DB Mail server can be configured to work with multiple fax servers concurrently in order to increase the overall system throughput. As your fax processing volume grows you can add additional fax servers as needed and configure DB Mail server accordingly.

Voice server

Voice Message Server (VoMS) is used to make phone calls and send sound messages using either pre-recorded sound files, dynamically synthesized voice messages (using text-to-speech functions) or combination of both. DB Mail server contains internal VoMS client, which is used to send sound messages, text-to-speech, and mixed messages to a Windows NT, Windows 2000 or Windows XP computer connected to a standard voice-modem or Intel Dialogic phone board hooked to a phone line. The computer must be also running VoMS server software. A limited edition of VoMS is provided with each DB Mail license.

Note: Phone calls and voice messaging are relatively slow operations. DB Mail provides built-in support for voice processing scaling out strategy (scaling out is the strategy that increases the capacity of an infrastructure tier to handle load by adding servers, thereby increasing the aggregate capacity of those servers). DB Mail server can be configured to work with multiple voice servers concurrently in order to increase the overall system throughput. As your call processing volume grows you can add additional voice servers as needed and configure DB Mail server accordingly.

Direct network messages

DB Mail currently supports two direct network messaging protocols and methods: sending network popup messages and sending administrative alerts. Both methods rely on the standard Windows API and can be used in Windows-based networks to deliver interruptible popup messages displayed on end-user computers.

DB Mail™
for Oracle, Microsoft SQL Server, Sybase ASE, and IBM DB2

Key Benefits

➤ Provides a robust messaging gateway that allows internal and external database applications to easily create and send various massages including e-mails, e-faxes, network popup messages, page and cell phone messages, and voice messages.

➤ Uses only standard database communication network protocols, which allow it to support any Oracle, DB2, Sybase and Microsoft SQL Server database systems running on any platform.

➤ Supports asynchronous message processing, which allows database applications to write messages to a message queue and be immediately freed to serve other user requests. Applications that use asynchronous operations (where more than a single operation can execute simultaneously) often have better availability, reliability, scalability, and usability than applications using only synchronous operations (where operations execute sequentially).

➤ Ensures quick return on investment and low cost of ownership. Utilizes widely available common hardware and software components that are inexpensive and affordable for businesses of all sizes.

➤ Supports efficient scaling out methods for time-consuming fax and voice message processing types. These methods allow increasing the capacity of an infrastructure tier to handle load by adding additional servers, increasing the aggregate capacity of the entire system.

➤ Provides unified messaging interface for different database systems thus allowing developers to create true cross-platform applications utilizing identical messaging functions.

➤ Features centralized graphical interface that shortens system setup and tuning curve and is easy to use.

➤ Provides robust error handling options including options for self-tuning and self-healing.

➤ Provides complete message processing audit trail that keeps track of messages being sent, in case troubleshooting needs to occur.

➤ Supports both regular email messages and messages with attachments. Up to 255 files or database blobs can be attached to a single message. Also supports messages with different content types including but not limited to plain text, HTML, XML.

➤ Full compatibility with all host Operation Systems on which the supported Database Management Systems can run, including but not limited to Windows NT, UNIX, Linux, VMS, OS/390, z/OS.

Additional Information

For additional information about DB Mail and to download a free evaluation version please visit http://www.softtreetech.com/dbmail/

About SoftTree Technologies, Inc.

Founded in 1997, SoftTree Technologies provides advanced software, services, solutions and technologies for business computing. Today we are focused on developing powerful tools for database management, monitoring, auditing, tuning, and coding. We are also developing enterprise strength software task scheduling and automation tools.

Our software is carefully designed for mission critical applications. That is why many organizations all around the world have chosen our products. Among them are 3M, AT&T, IBM, JP Morgan Chase, Hewlett- Packard, Shell, Sony, USA Army, NASA and many others.

Trademarks

DB Mail, DB Tools for Oracle, DB Audit, 24x7 Automation Suite and 24x7 Scheduler, VoMS are trademarks of SoftTree Technologies, Inc. in the United States and other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.