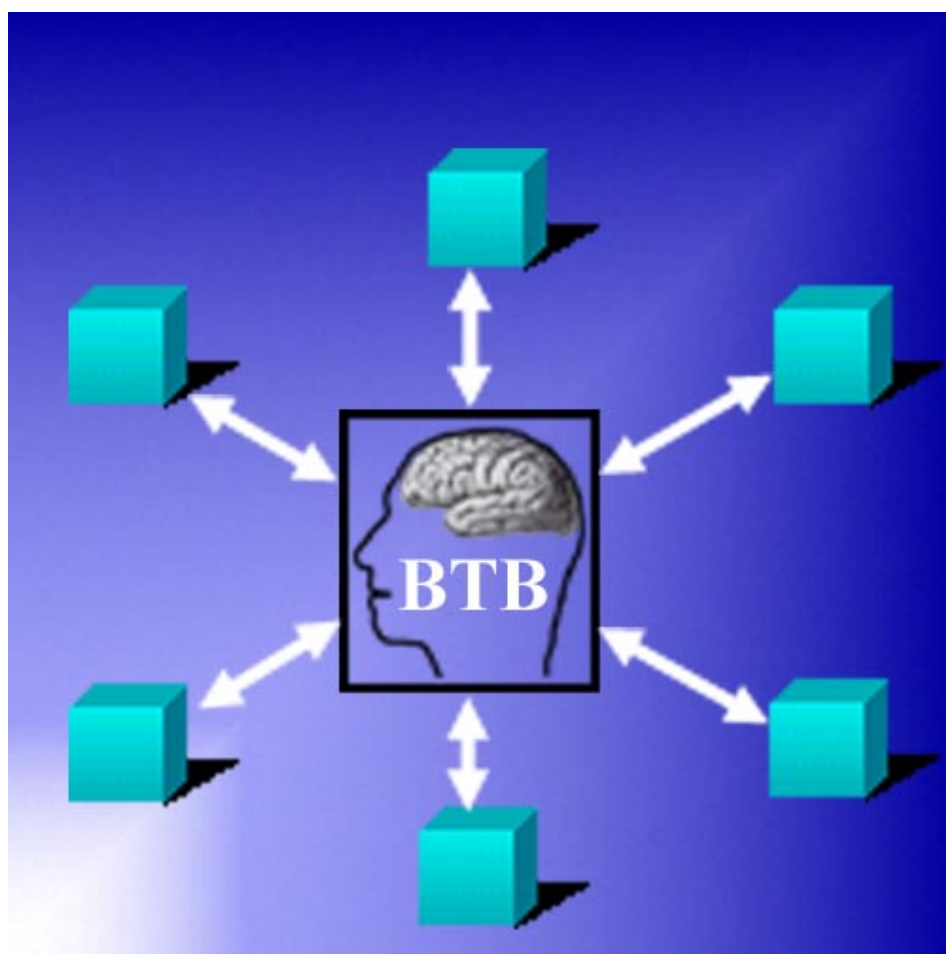


# Facilities Guide

*Version 1 Release 0*





Business Transaction Broker  
for z/OS



# Facilities Guide

*Version 1 Release 0*

**Note!**

Before using this information and the product it supports, be sure to read the general information under “Notices” on page vii.

**Second Edition (September 2002)**

This edition applies to Version 1, Release 0 of Business Transaction Broker for z/OS (product number 5799-GPR) and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Download publications via internet at the following address:

<http://www.ibm.com/software/ad/imb>

Publications are not stocked at the address below.

Please address your comments to:

IBM Global Services  
Sortemosevej 21  
DK-3450 Allerød  
Denmark

☎ + (45) 45 23 30 00  
Fax + (45) 45 23 68 01  
E-mail [SPOC@dk.ibm.com](mailto:SPOC@dk.ibm.com)

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1992, 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Figures</b>	v
<b>Notices</b>	vii
Trademarks and service marks	vii
<b>Preface</b>	ix
About this book	ix
Who should read this book	ix
Conventions and terminology used in this book	ix
Note about version and release numbering	ix
Bibliography	x
Books from related libraries	x
How to send your comments	xi
<b>Summary of changes</b>	xiii
<b>Chapter 1. Introducing Business Transaction Broker</b>	1
BTB architecture	2
BTB services	3
Introducing the MailRoom	6
Document formats	6
Auditing and logging	8
National Language Support (NLS)	8
<b>Chapter 2. Administering resources in BTB</b>	9
Understanding resources	10
Trading Partner registrations	13
<b>Chapter 3. Understanding the MailRoom</b>	17
Terminology used in the MailRoom	20
<b>Chapter 4. Functionality of the MailRoom components</b>	27
Source scenarios	27
The kernel component	30
Destination scenarios	32
Automated proactive monitoring	35
MailRoom acknowledgment settings	35
<b>Chapter 5. MailRoom Audit Facilities</b>	37
Audit logging and reconciliation	37
<b>Index</b>	39



---

## Figures

1.	BTB architecture and services . . . . .	3
2.	Administration responsibilities . . . . .	9
3.	Services . . . . .	11
4.	Business Programming Interfaces . . . . .	11
5.	Back-end definitions . . . . .	12
6.	Trading Partners . . . . .	14
7.	BTB MailRoom component overview . . . . .	18





---

## Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

---

## Trademarks and service marks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM	MQSeries
AIX	MVS/ESA
CICS	OS/2
DB2	OS/390
ExpEDIt	QMF
IBM	RACF
IBM Global Network	VTAM
IMS	WebSphere

The following terms, used in this publication, are trademarks of other companies:

Mercator	Mercator Software, Inc.
SAP	SAP AG

Windows and Windows NT are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.



---

## Preface

---

### About this book

This book provides an introduction to the facilities of Business Transaction Broker for z/OS (BTB) middleware and shows how they can be used to implement a client/server processing environment, as well as how to facilitate a re-engineering process towards such an environment.

IBM Business Transaction Broker for z/OS helps address brokering requirements and helps enable enterprises to implement brokering solutions with confidence. While IBM Business Transaction Broker for z/OS is designed to facilitate Application-to-Application (A2A) brokering for interchange between internal applications within a customer's enterprise, the dominant strength of this product is revealed and appreciated when implementing Business-to-Business (B2B) brokering for the external interchange outside of the customer's enterprises.

To better emphasize its B2B brokering capabilities, the IBM program product formerly known as Intelligent Message Broker for z/OS has now been renamed Business Transaction Broker for z/OS (BTB).

---

### Who should read this book

This book is for system planners, architects, and developers building extended enterprise solutions. It describes how the BTB middleware facilities can be used:

- In developing client/server solutions, including inter-organizational systems (IOS) across different platforms
- To establish advanced enterprise-wide EDI solutions
- As a systems integration tool, for example involving a merger of companies or departments which have different solutions and platforms
- As a productivity tool set for CSP-DB2 application development

---

### Conventions and terminology used in this book

In this book, Business Transaction Broker for z/OS is referred to as "Business Transaction Broker" or simply "BTB" where the context makes the meaning clear.

---

### Note about version and release numbering

Where the documentation for Business Transaction Broker - in particular the *Installation Guide* - refers to the actual contents of the product, e.g. dataset names, it will refer to Version 4, Release 5. This is because this product has a history of being distributed internally within IBM as Intelligent Message Broker (IMB) for a number of years. Within IBM the product has matured to a Version/Release level of 4.5.

---

## Bibliography

### Business Transaction Broker for z/OS books

<i>Business Transaction Broker for z/OS General Information</i>	GC27-1580
<i>Business Transaction Broker for z/OS Facilities Guide</i>	SC27-1584
<i>Business Transaction Broker for z/OS Installation Guide</i>	GC27-1581
<i>Business Transaction Broker for z/OS Application Programming Guide</i>	SC27-1583
<i>Business Transaction Broker for z/OS System Administration Guide</i>	SC27-1582
<i>Business Transaction Broker for z/OS User Administration Guide</i>	SC27-1585

Futher copies of the Business Transaction Broker for z/OS publications can be downloaded from the product web site:

<http://www.ibm.com/software/ad/imb>

---

## Books from related libraries

### CICS TS books

<i>CICS Transaction Server for OS/390 Release Guide</i>	GC34-5352
<i>CICS Transaction Server for OS/390 Installation Guide</i>	GC33-1681
<i>CICS Transaction Server for OS/390 System Definition Guide</i>	SC33-1682
<i>CICS Transaction Server for OS/390 Resource Definition Guide</i>	SC33-1684
<i>CICS Transaction Server for OS/390 Operations and Utilities Guide</i>	SC33-1685
<i>CICS Transaction Server for OS/390 System Programming Reference</i>	SC33-1689
<i>CICS Transaction Server for OS/390 Messages and Codes</i>	GC33-1694
<i>CICS Transaction Server for OS/390 Intercommunication Guide</i>	SC33-1695
<i>CICS Transaction Server for OS/390 Performance Guide</i>	SC33-1699

### DataInterchange for MVS books

<i>DataInterchange Messages and Codes</i>	SB34-2000
<i>DataInterchange Programmer's Reference</i>	SB34-2001
<i>DataInterchange Administrator's Guide</i>	SB34-2002

### DB2 books

<i>DB2 for OS/390 Administration Guide</i>	SC26-8957
<i>DB2 for OS/390 Application Programming and SQL Guide</i>	SC26-8958
<i>DB2 for OS/390 Command Reference</i>	SC26-8960
<i>DB2 for OS/390 Utility Guide and Reference</i>	SC26-8967

### Expedite/CICS books

<i>Customizing and Developing Applications with Expedite/CICS</i>	GC34-3304
---	-----------

### IBM EDI Services Information Exchange books

<i>IBM EDI Services Information Exchange Interface Programming Guide</i>	GC34-2222
--	-----------

### MQSeries books

<i>MQSeries Planning Guide</i>	GC33-1349
<i>MQSeries for OS/390 System Management Guide</i>	SC34-5374

### **MVS books**

<i>MVS Planning:Global Resource Serialization</i>	GC28-1759
---	-----------

### **RACF books**

<i>OS/390 Security Server (RACF) Administrator's Guide</i>	SC28-1915
--	-----------

### **Automated Operations Control/MVS books**

<i>AOC/MVS Planning and Installation</i>	GC28-1082
<i>AOC/MVS Operations</i>	GC28-1084

### **ACF/VTAM books**

<i>Advanced Communications Function for VTAM Installation and Resource Definition</i>	SC23-0111
---	-----------

---

## **How to send your comments**

Your feedback is important in helping to provide the most accurate and highest quality information. If you have any comments about this book or any other BTB documentation:

- Send your comments by email to *SPOC@dk.ibm.com*. Be sure to include the name of the book, the part number of the book, the version of BTB, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
- Mail or fax your comments to the address at the front of this book.



---

## Summary of changes

The major enhancements to this release of Business Transaction Broker are:

### **Source Exit**

BTB now supports source exits, which makes it easier to implement support for new document types. One new area in this release is XML document routing support. The source scenario will handle incoming documents like this:

- A business document in its original format is passed to BTB. MailRoom will perform a lookup to determine if a source exit should be invoked.
- For MQSeries based scenarios there is an Unpack Exit that should split the received MQ buffer into records.
- For all scenarios (except BATCH and TIE-IMS) there is a Source exit that can build BTB routing information (the M-record) if it is not available in the received document.
- The received document together with the generated M-record is stored in the transport table. The remaining processing is unchanged.
- The MailRoom source exit table holds registrations of exits used by source scenarios. Source exits are either global for a scenario, or specific for a single sender.
- MailRoom supplied source exits will, as default, be used to build the M-record for the DI-EDI, EXP-FILE, SAP and SAP-MQ source scenario. A source exit with XML support is available (see below). Other user written source exits can be written to extend the format support in MailRoom.

### **XML support**

BTB now supports receiving and dynamic routing of XML documents. A XML source exit is available to extract XML elements for building BTB routing information (the M-record). A new panel is available to define which XML elements or attributes should be extracted for each XML document type. A new kernel / destination exit can automatically convert a received XML document to a basic flat file format.

### **MQSI V1.1 (MVS) reformatter exit**

BTB now has an MQSI reformatter exit. In the first version it can format a document to a stream format (records separated with character delimiters), put it on a queue, wait for MQSI to reformat it, get the reformatted stream and finally return it to MailRoom as an updated version of the document.

### **MailRoom Continuous Receive panel.**

A new online function makes it easy to check the status of DataInterchange and Expedite/CICS continuous receives. DI profiles can be stopped and started. It is possible to define a number of profiles as mandatory. A function key can be used to check if the mandatory profiles are active.

**Archiving depending on status**

Every installation must archive data from the MailRoom DB2 tables in order to remove old data. It is now possible to define the status codes that are applicable for archiving. It is thereby possible to reflect the day to day operational method in the archiving strategy. E.g. FAILED documents are considered open problems and stay in MailRoom until fixed (status changes to FINISH or manually CANCELED). Another principle could be that FAILED documents should be automatically archived.

**Schedule process change**

A change has been made in the Output Schedule in MailRoom to control the order of sending scheduled envelopes for the same destination. Previously the sending order was in envelope key sequence. Now scheduled envelopes to the same destination are grouped in a single destination envelope and the sending order has been changed to the received order.

**Possibility to Mail Text segments**

In error situations, when processing EDI like EDIFACT and ANSI X12, a copy of the EDI document is needed for documentation. A new function has been implemented to mail the DataInterchange reports and segments to a specific user Id and node or e-mail address.

**Reflow of saved DI audit reports**

The text in the DataInterchange audit reports is now automatically reflowed to the width of the panel. It is no longer necessary to scroll right and left to read the audit report.

**Segmentation of saved EDIFACT envelope**

The saved EDIFACT envelope file is now split in segments to improve the readability.

**Handling of a group of envelopes/requests**

A new facility is made available to handle more than one envelope using a single group command. Some of the online functions (cancel, fail, finish, resend, restart, sysack) are made available as a group command for multiple envelopes.

**Automatic resending of MAIL destination to Internet**

We now put documents on the MailRoom XMIT queue if the SMTP server is temporary unavailable.

**New MQSeries Batch Read Utility**

In addition to the MQSeries Batch Write Utility introduced in last release we now also have a MQSeries Batch Read Utility that can get MailRoom formatted messages from a MQ queue and write them to a file.

**SAP R/3 Version 4 improvements**

MailRoom now supports sending and receiving SAP R/3 Version 4 IDOCs.



**Codepage conversion exit**

A new codepage conversion exit is now available as kernel or destination exit. With supplied codepage translation tables it can convert a document from one codepage to another.

**Mercator remapping exit improvements**

The Mercator exit has been improved with support for selection of profiled or fixed length for input and output (:P or :W parameters to Mercator). Now it is also possible to pass additional command options to Mercator.



---

## Chapter 1. Introducing Business Transaction Broker

Business Transaction Broker, or BTB, is a mainframe-based solution that you can use to achieve business integration in a heterogeneous network connecting disparate platforms. BTB enables and controls communication between an organization and its trading partners, allowing business transactions with the trading partners to be run, controlled, and monitored with minimum effort.

Electronic Data Interchange (EDI) services are controlled by the BTB MailRoom components, interfacing with IBM DataInterchange, or directly to Expedite/CICS.

BTB is also a client/server solution — Windows, OS/2, and AIX clients are supported.

BTB supports these access methods:

- 3270-type native or emulated terminal access (also called NPT, or non-programmable terminals)
- Asynchronous data exchange using the IBM Global Network service Information Exchange.
- Client/server access, with Windows, OS/2 or AIX clients accessing the servers. BTB acts as middleware, controlling access to the servers.
- Messaging using MQSeries

BTB runs under CICS and can communicate with:

- IMS applications
- CICS applications
- APPC-connected systems
- TCP/IP-connected systems

### Inter Organizational Systems (IOS)

This is the general term for the cross-company electronic data processing systems needed to support the *Extended Enterprise* concept.

*Extended Enterprise* includes all its Trading Partners such as Agents, Dealers, Suppliers, Banks, Transporters, Government, and Publishers.

By setting up an IOS, an enterprise declares its intention to improve the cooperation between its strategic alliances and business partnerships by substantially improving the flow of electronic information. In a system that supports an *Extended Enterprise*, each party is seen as a potential user of IOS services, *and* as a potential service supplier to the *Extended Enterprise* community.

Systems offering external parties access to internal systems require a high degree of security and auditability. BTB has achieved a *generic*, product IESC certification,<sup>1</sup>.

---

<sup>1</sup> IBM term for "Inter Enterprise Systems Connection"

## Electronic channels

Electronic channel support extends the function and power of electronic services, focusing on exploiting the re-engineering of the marketplace.

Internet connectivity allows an *Extended Enterprise* to reach the ultimate consumer of products and services.

BTB offers an electronic infrastructure when establishing a single electronic channel support system. BTB supports the business processes by enabling and controlling information flow on products and services between an enterprise and its Trading Partners.

The electronic services developed using BTB cover access to business applications, which initiates and reflects the ongoing business between the Trading Partners. The business application can be accessed using 3270 technology (NPT), Personal Workstations (PWS), or invoked by EDI transactions.

The BTB middleware provides extensive support tools to business application designers and developers.

## Process re-engineering

Reuse applied to the *Extended Enterprise* implies that external users are able to access the same business applications as those used by *internal* employees.

Reuse applied to *internal systems*, offers adaptation to change. Existing business processes can easily be split into their components and combined in new ways to better solve changing business needs.

---

## BTB architecture

BTB is a set of middleware components, that enables and controls the communication between client and servers. Instead of direct connections between clients and servers, the BTB middleware acts as the host-based gateway.

The advantages of this design are:

### Reduced cost

Instead of workload processing being carried out at the various organizations where the servers are situated, gateway processing is managed on a small number of gateways, reducing development, maintenance and service delivery effort.

### Increased responsiveness

When new functions are requested by users (for example TCP/IP or Lotus Notes connectivity), all existing servers on all existing Nodes can be made accessible immediately, by implementing the new requirement once—in the gateway.

With the BTB SAP Bridge it is possible to offer any SAP user immediate access to all existing host based legacy systems. It is also possible to offer any distributed user a secure and controlled online connection to newly-developed SAP servers.

## Minimized Service Delivery impact

Defining and maintaining all the required network definitions on all the platforms can be very complicated when implementing a large, distributed client/server environment. By using a host gateway, these network registrations are reduced dramatically. From a network standpoint, all the clients are communicating with one remote node—the gateway. Additionally, the servers are only communicating with one, or a few BTB instances, instead of with many LANs.

## Megacenter philosophy

Because BTB has implemented Multi-Country Operation (MCO), and National Language Support (NLS), it is well suited for a *Megacenter* scenario. One BTB platform can be both cross-country and cross-language. See “National Language Support (NLS)” on page 8.

## BTB services

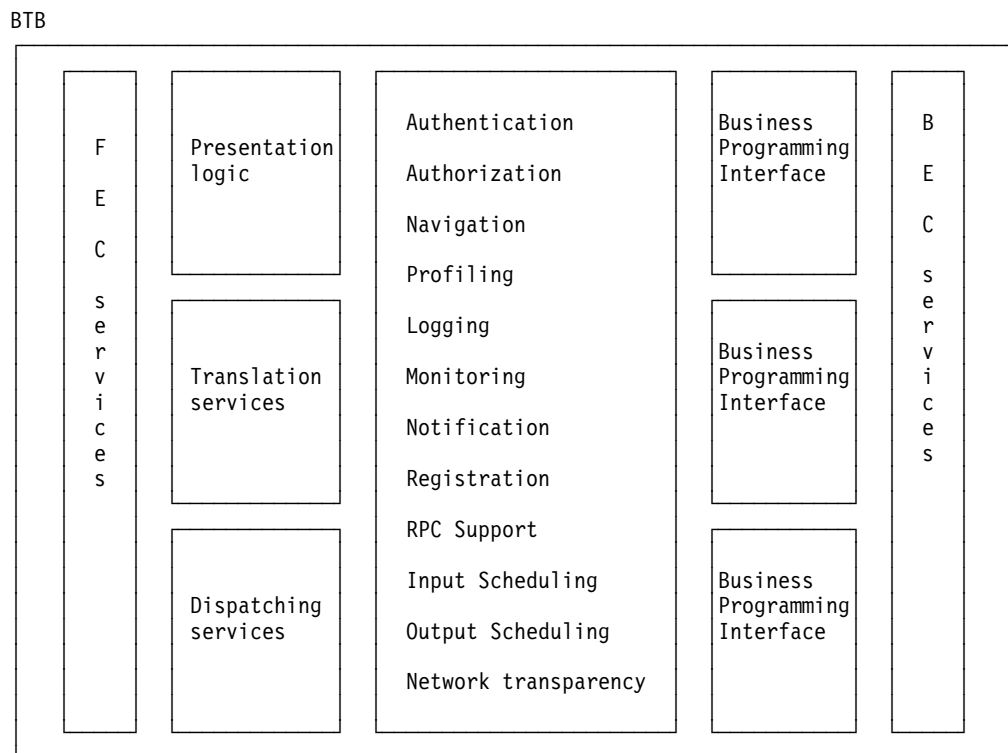


Figure 1. BTB architecture and services

BTB consists of front-end and back-end services, providing the services shown in Figure 1 for clients and servers.

## Transport independence

BTB provides complete transport transparency between the transport methods or protocols used by senders or receivers. Among the transport methods BTB supports are MQSeries, SAP R/3, IBM Information Exchange, e-mail, and fax in any combination between senders and receivers. You can even use BTB to translate incoming faxes using optical character recognition (OCR). By using *thin clients* on a variety of platforms, BTB provides connectivity using either APPC or TCP/IP.

## Rule-based fanning

For most messages the routing in BTB is statically defined using the subscription dialogues. Rule-based routing can also be used. In this case, BTB analyses the incoming message and decides upon the routing based on simple, predefined table-based boolean logic. Fanning is a method used to send one incoming message to many recipients—like a distribution list.

## Routing

Routing functions enable all communication processing to be handled outside of the application. BTB routes messages to destinations using the correct protocol, based on information that has been recorded by the BTB administrator. Dynamic routing is possible where the application data contained in the message dictates where the message is to be routed.

## Data conversion

Exits are provided to allow business applications process, and to optionally update messages.

The remapping of messages can be done by application code, or MQSeries Integrator (MQSI) can be used to do the actual mapping. Support for the Mercator mapping tool is also fully integrated in BTB.

There is specific, generalised support for remapping of XML messages into a more easy to handle flat file format.

EDI mapping is handled by IBM DataInterchange, which is an integral part of the BTB tool box.

## The message repository

BTB provides audit trail and resend facilities. When a message is received by BTB, it is immediately stored *as is* and can be monitored any time. If an exit changes the contents in any way, the new version is also stored and linked to the original. Using the repository it is also possible to resend a message if the receiving application (or Business Partner) should need to.

## Security and authentication

BTB lets internal and external users integrate their business. BTB uses standard RACF protection of its resources. Internal users must provide a user ID and password for authentication. External users must pass additional gateway controls ensuring extended user authentication. All users must be pre-registered. User access is limited to the specific authorizations that exist for every user.

## Profiling

For workstation clients BTB provides an API that returns an *access list* providing authorizations for signed-on users. This information can be used to *personalize* the user interface on the workstation, for example, to make certain options non-selectable on pull-down menus, if the current user is not authorized to perform them.

It has also shown to be a great advantage to be able to handle security in this centralised manner on the mainframe, instead of at the many decentralised server instances.

## **Navigation**

This makes the location of application and data servers transparent to the client application. The location is known only by the table-driven BTB components. When the location of a server changes, only one entry in a controlling table in BTB need be changed.

Not all servers need to be accessible from each BTB system. With the appropriate registrations, it is also possible to route a request to another control and service layer.

## **Network transparency**

Programmers need not issue direct calls to APPC or TCP/IP. Instead, they issue a call to a local server, providing a logical transaction name and related data as parameters.

## **Logging**

Extended online logging and archiving facilities allow for auditing and billing requirements to be satisfied. A starter-set of QMF queries for reconciliation is provided with BTB

## **Proactive monitoring and notification**

BTB continuously scans its own processes in the background, enabling end-to-end control, both at the business application level, and at the transport level.

## **Network protocol conversion**

The front-end and back-end communication services can handle a variety of network protocols, allowing client applications to access BTB using one set of protocols, while BTB accesses the business integration servers using another set of protocols. This allows the internal environment to remain stable while extending client connectivity options with, for example, TCP/IP, or Lotus Notes connectivity.

## **Input Scheduling**

BTB can hold information about expected input frequency as well as server-application planned down-time. This can be used to check that input is arriving as expected, and to alert the appropriate group if a transmission is missing.

## **Output Scheduling**

Asynchronous server requests can be momentarily stopped in the MailRoom and scheduled for later delivery to their respective destinations. BTB automatically holds output messages and releases them according to registered schedules.

## **Multiple access modes to business applications**

BTB enables non-programmable terminal (NPT) access, interactive client/server processing, and asynchronous, queued processing for the same business applications.

Asynchronous, queued processing is handled by the MailRoom component of BTB.

MailRoom is also used for EDI implementations, as well as to *batch* on-line transactions, while waiting for an application to be released.

The different access modes exist concurrently, facilitating transition from legacy applications to distributed solutions.

An advantage of BTB supporting three access modes is that the interactive service can strengthen the generic EDI service and vice versa.

E.g. the progress of the (asynchronous) EDI process can be queried by an external partner organization using interactive services, while an interactive service can be used to request an asynchronous upload of data using generic MailRoom services.

### **Client/server design**

The business applications are activated as a server in a layered client/server chain initiated by the user client.

### **Modularity**

BTB is scalable, and additional function can easily be added. Additional services include generic error and information message service, generic help services, and statistics gathering.

### **NPT development environment**

BTB has an extensive application development platform, providing application skeletons, tools packages, and application design guidance to enable a rapid application development process.

---

## **Introducing the MailRoom**

The MailRoom enables and controls EDI communication between an organization and its trading partners. The interface to business applications is provided through source and destination setups supporting:

- APPC
- Batch
- CICS programs
- IMS programs
- MQSeries
- SAP
- TCP/IP

The BTB MailRoom is based on BTB back-end, DataInterchange, Expedite/CICS and optionally TIE running in CICS, and partly in IMS. The application programming interfaces (APIs) also support batch jobs.

---

## **Document formats**

The MailRoom can process various types of business documents, such as invoices and orders. The primary format is M- and D-records, other recognized formats are XML, SAP R/3 IDOC and EDI (EDIFACT and ANSI X12). Even flat files in other formats can be routed through MailRoom if necessary registrations are done or user exits written.

All processing and routing of documents in MailRoom are depending on the MailRoom registrations and identification of sender, receiver and format contained inside the document in an M-record, see "The M-record" on page 21. If a document is received without an M-record it is necessary for MailRoom to build one. For some source scenarios and well known document types, this can be done automatically in MailRoom supplied source exits. These exits will extract fields from the received document and dynamically build an M-record and if necessary also encapsulate other records in D-records.



If you have other document formats, you must either build the M-record before passing it to BTB or you can write your own source exit. Writing your own source exit is a way to extend MailRoom to understand other document formats, see: "Source exits" on page 24. The received file (or transmission buffer for MQSeries) is passed to the source exit which must return a valid document with M- and D-records.

The document will now be stored in the transport tables as a standard MailRoom document, where the first record is the M-record. The remaining records are data records (first character usually **D**) or special purpose control records (for example DataInterchange C-, E-, I-, and Q-records). One or more documents received within the same syncpoint are considered to be one envelope, which is assigned a unique envelope ID by the MailRoom.

## **MailRoom M- and D-records**

MailRoom M- and D-records is the native MailRoom format. The received document is stored *As Is* in the MailRoom transport table. The M-record controls the processing and routing in MailRoom.

Read about the M-record in "The M-record" on page 21.

## **EDI, EDIFACT, ANSI X.12**

EDI sent to or received from MailRoom via IBM Information Exchange with the DI-EDI scenarios will be translated to or from InHouse by DataInterchange under MailRoom control. The Inhouse format, together with an M-record, is stored in the MailRoom transport table.

It is also possible to use other scenarios than DI-EDI to send and receive EDI to/from MailRoom. The EDI could be sent to MailRoom as a flat file (see below) with an M-record built by a source exit. Translation to InHouse format can then be performed with an exit. The opposite direction is also possible.

## **XML**

The XML document will be encapsulated in M- and D-records (i.e. including the generation of a necessary M-record), before it is stored in the MailRoom. For more information about processing XML documents, refer to the Application Programming Guide.

## **SAP R/3 IDOC**

The SAP R/3 IDOC will be encapsulated in M- and D-records (i.e. including the generation of a necessary M-record), before it is stored in the MailRoom.

## **Flat files**

The flat file will normally be encapsulated in M- and D-records (i.e. including the generation of a necessary M-record), before it is stored in the MailRoom. A record starting with M would be interpreted as an MailRoom M-record.

## Other formats

The MQ source scenario can be used to receive other buffer formats than the normal MailRoom MQ buffer format.

It is necessary to use a user written source unpack exit to split the buffer into records. MailRoom can also cut records at 1999 byte boundary with MQ default unpack exit.

The MQ destination scenario can also be used to send other buffer formats than the normal MailRoom MQ buffer format.

The received buffer must be cut into appropriate records which will be stored in the MailRoom transport tables.

---

## Auditing and logging

BTB has built-in auditing and logging features. Some of these features can be used by developers to get an audit trail of application usage. Refer to the *Application Programming Guide*, and to the *System Administration Guide* for more details. The MailRoom also has full logging capability for all incoming and outgoing documents.

---

## National Language Support (NLS)

BTB is designed with MLS (Multi Language support). The components of BTB that are intended for general users support many languages. Language is a user attribute, so one copy of BTB can support users with different languages at the same time.

All panels in BTB are in English. The unloaded message table contains English, Danish, and BTB technical support messages.

The language codes used are the ISO 639.2 2-character language codes. BTB accepts all two-character language codes in ISO 639.2.

**Note:** *UK* indicates *UK English*. *EN* is defined for eventual future migration to the standard language code.

The contents of all the MLS panels are kept in CSP tables and *language code* is part of the key. To add a new language, these tables must be extended or updated with the texts in the new language. This is also the case for the message table. The *Application Programming Guide* shows how the CSP tables are built.

## BTB Multi Country Operation.

BTB can operate several countries (or organizations) within the same copy of BTB. The DB2 tables are all assigned country-specific IDs (*countryid*), so users of one country will not be confused with users from another country with same trading partner number.

A scope check facility lets you delegate some administrative tasks on a country level. For more information, refer to *User Administration Guide*.

## Chapter 2. Administering resources in BTB

BTB includes many resources, and it is the administrator's responsibility to define and make these resources available for Trading Partners and users of BTB.

Who should have access to which resources and to what data is decided by the organization owning and running the BTB system. An administrator delegates this access to Trading Partners and users.

Administration can be divided into three areas:

- System administration, described in the *System Administration Guide*
- Central administration, described in the *User Administration Guide*
- Local administration, described in the *User Administration Guide*

Tasks and responsibilities for each type of administrator are shown in Figure 2.

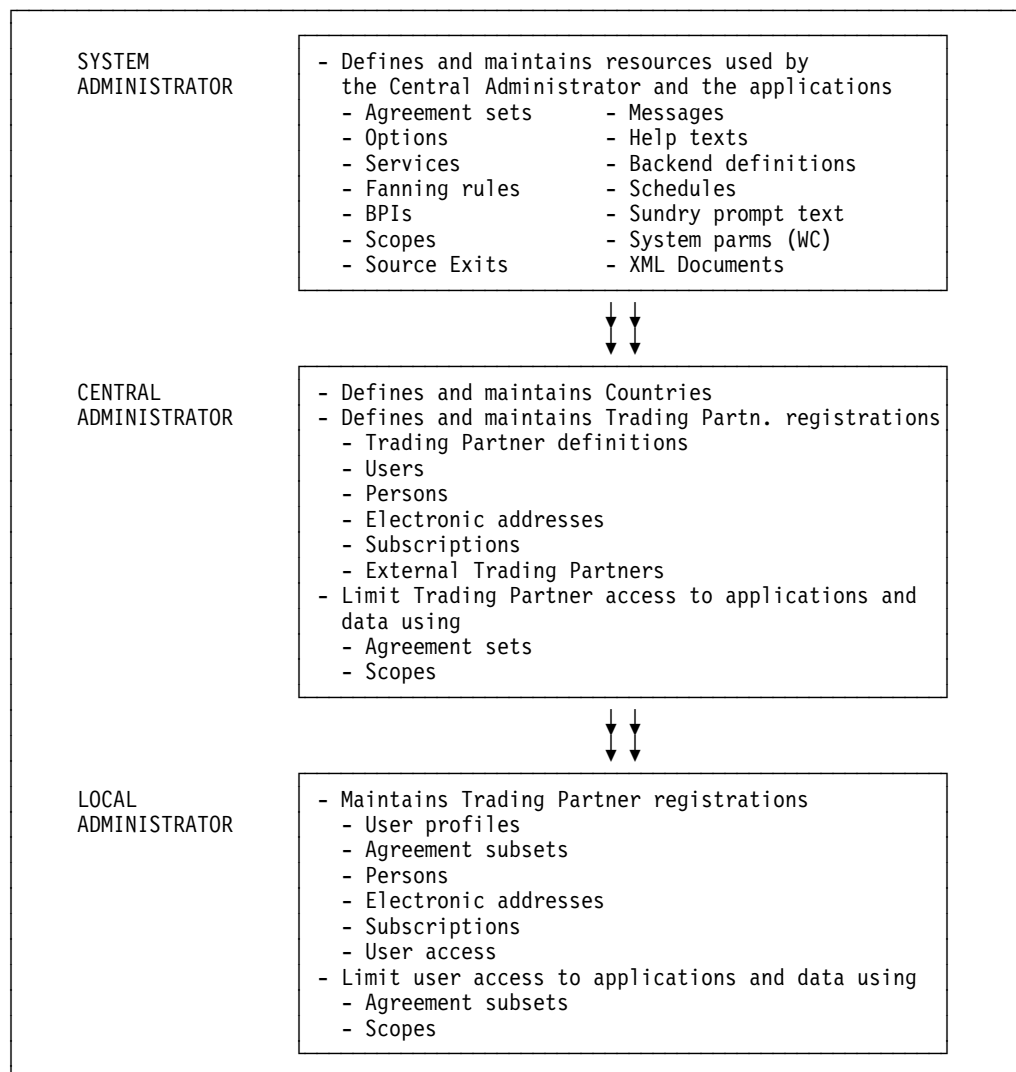


Figure 2. Administration responsibilities

The System and Central user administrators are defined centrally by the BTB-owning organization.

The users within the Trading Partner's organization by default have the same access to resources as the Trading Partner. A Local user administrator is defined only if the Trading Partner organization needs to give user groups different access to BTB resources.

---

## Understanding resources

Resources are divided into two groups:

- Resources for defining and grouping the server applications:
  - Agreement Sets
  - Options
  - Services
  - BPIs
  - Scopes
- Resources to be used by the server applications:
  - Messages
  - Help texts
  - Back-end definitions
  - Input scheduling
  - Output scheduling
  - Rule-based fanning rules
  - Sundry prompt text
  - System parameters
  - Source Exits
  - XML Definitions

### Agreement Sets

An Agreement Set is an authorization pool consisting of online menu options, MailRoom services and client/server Business Programming Interfaces (BPIs).

Agreement Sets are project oriented. One or more Agreement Sets are linked to a Trading Partner account. Multiple Trading Partners can be linked to the same Agreement Set.

### Options

BTB provides an execution environment for host-based 3270 client applications. These clients are registered as Options in BTB.

By specifying an Option, you can invoke a user-written program, or any business application product from the BTB panels.

### Services

Asynchronous MailRoom processes are defined as Services. Services and Subscriptions combine to describe the characteristics of the sender and receiver systems, and to describe the data format to interchange. A defined Service can be subscribed to by a number of Trading Partners (see Figure 3 on page 11).

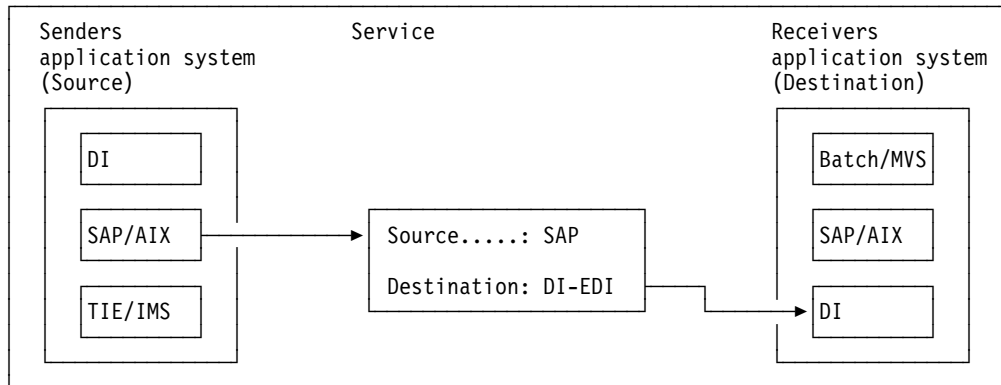


Figure 3. Services

The source and destination systems are independent, so any source system can be combined with any destination system in a Service.

### Business Programming Interfaces (BPIs)

The Business Programming Interface lets you use BTB as a synchronous client/server gateway for clients needing to activate various business application servers.

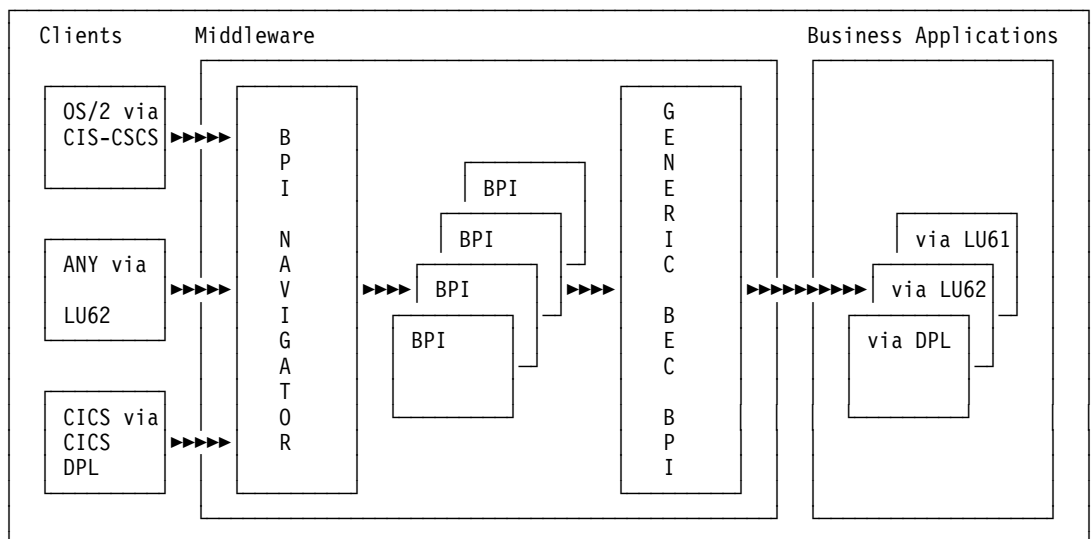


Figure 4. Business Programming Interfaces

The business applications are activated as the ultimate server in a layered client/server chain.

### Scopes

Scopes are a restriction filter for applications and to data. A Trading Partner can be given a Scope to limit which data to work with. Users will by default be given access to the same resources and data as the Trading Partner that they are linked to. However, it is possible to limit user access by using Scopes.

## Messages

Error Messages and Event Messages are message numbers and message IDs with associated message texts to be used by BTB and user applications running in BTB. Messages can be inserted and customized from the BTB panels.

Error messages are used by BTB itself and by user applications (both online NPT applications and BPI client/server applications).

MailRoom Event Messages are used by the MailRoom online monitoring facility, exits, and user applications.

## Help texts

The BTB help system is a generic component that stores text information and presents it to users from the BTB panels. It consists of a DB2 table to store the help text, a CSP call to an API to present the help, and a utility program to control the loading of help text.

BTB uses this system for online help for menus, monitoring panels and registration panels. It can also be used from other 3270-based applications running under BTB.

## Sundry prompt text

This is a generic place to store character, decimal and integer values, and related descriptions for later use in applications as *F4 Prompt Lists* and as input validation.

The Sundry support consists of a DB2 table to store the sundry texts, a CSP call to an API to verify values, and a set of CSP applications to maintain Sundry texts online in BTB.

## Back-end definitions

### Back-end Communication (BEC) destinations

Describe the technical attributes of the back-end business server (for example, the communication protocol used and the LU name of the system). These definitions are used when defining BEC applications.

### Back-end Communication (BEC) Applications

Describe the transaction to be activated in the back-end system.

Figure 5 shows the relationship between Services, BEC applications, BEC destinations, and business applications:

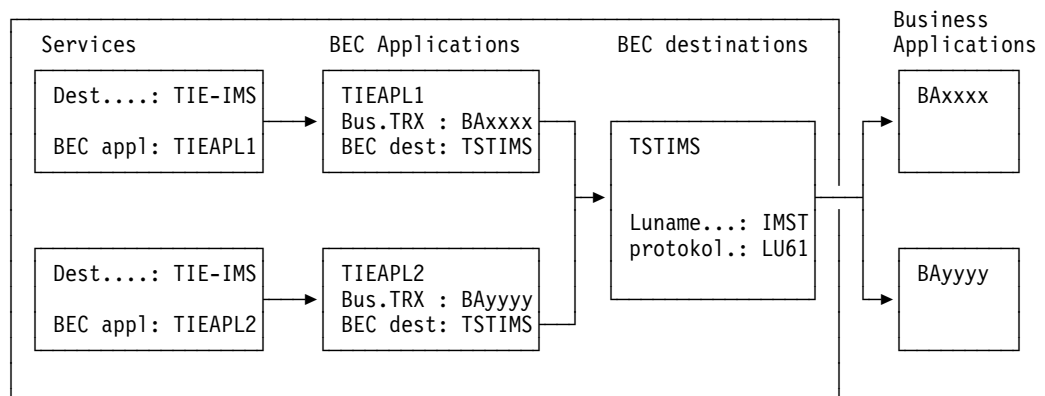


Figure 5. Back-end definitions

## **Input Scheduling**

This feature lets MailRoom check that input is arriving as expected. The input schedule can detect a missing daily transaction and then alert the Service Point of Contact (SPOC).

## **Output Scheduling**

Can be used to specify if certain asynchronous requests should be temporarily stopped in the MailRoom and scheduled for later delivery to their respective destinations. This is done by defining the schedules, and then referring to the Schedule entity on the Service definitions.

It can also be used in the synchronous client/server scenario to temporarily stop access to the BPIs.

## **Rule-based fanning rules**

These are used to decide which Trading Partners can receive a copy of the document currently being processed. The rules can, for example, instruct MailRoom to look for specific text strings in the document and to use the retrieved values to make the final list of Trading Partners.

Rule-based fanning will be activated if rules have been defined, extended routing is selected on the Service, rule-based fanning is selected as extended routing type, and the defined set of rules is referenced by the Service.

## **System parameters**

These are stored in the Working Criteria (WC) table. Default system parameters are delivered with BTB, for example parameters to control MailRoom scanner frequency. For information about these parameters, refer to the *Installation Guide*.

Other parameters will be defined as the system administrator sets up the receive and send environments for the Trading Partners.

## **Source Exits**

Source Exits is used by MailRoom before storing the incoming data. The Source Exits can either be a user written program or one of the MailRoom standard Source Exits. MailRoom looks in the Source Exits definitions and checks whether a Source Exit should be executed before storing the data.

## **XML Definition**

XML Definition is used by MailRoom to dynamically assign MailRoom information for arriving XML Documents. The XML Processor source exit must be activated to enable this function.

---

## **Trading Partner registrations**

Trading Partner resources are defined and maintained by the Central Administrator (refer to the *User Administration Guide*).

## Trading Partners

The Trading Partner is the main entity in the user area. It is used to identify accounts or companies.

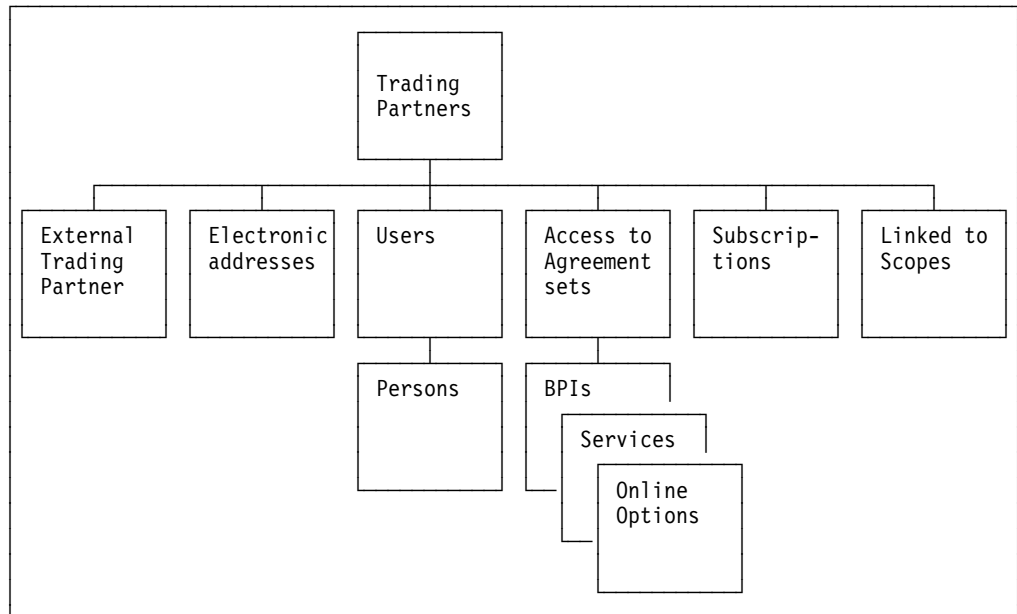


Figure 6. Trading Partners

Users and Persons can only exist if they are associated with a Trading Partner account.

Nodes and Electronic Addresses are defined for a Trading Partner, and can be referenced by all Users and resources that this Trading Partner has access to.

Agreement Sets, Subscriptions and Scopes are authorization entities defined by a System Administrator. They can be linked to different Trading Partners and Users, controlling their access to the resources.

## External Trading Partner

This is an alias for a specific Trading Partner. An external Trading Partner ID can be mapped to an internal Trading Partner ID.

A Trading Partner can have zero or more external Trading Partner IDs defined as aliases. The same external Trading Partner ID cannot exist for two different Trading Partners.

## Users

These can be users for a Trading Partner, or BTB administrators. The user IDs must be defined to RACF.

The default access is for the users within a Trading Partner account have access to the same objects and resources as the Trading Partner.

User access can be further limited by adding Agreement Subsets and security Scopes, allowing many users with different security Scopes for the same Trading Partner.



## **Persons**

The Person entity can be used to register personal data for a user, such as name, job title, and telephone number. If required, a person can have multiple user IDs.

## **Electronic addresses**

These identify a destination or mailbox within a remote system. For example, user IDs for an IE mailbox, or user IDs defined in Mail Exchange.

Electronic Addresses are defined locally for a Trading Partner, or they can be shared on country or system level. The Users and MailRoom Services and Subscription defined for that Trading Partner use these Electronic Addresses.

BTB supports these electronic address types:

<b>FX</b>	Telefax number
<b>HO</b>	Host address
<b>IE</b>	IE Mailbox (IBM Information Exchange)
<b>IN</b>	Internet e-mail address
<b>LI</b>	Distribution list
<b>ME</b>	IBMMAIL IEA (IBM Mail Exchange)
<b>TC</b>	TCP/IP Server (Hostname or IP address)
<b>TL</b>	Tools Append
<b>VM</b>	VM/Host address

## **Subscriptions**

Before a Trading Partner uses a MailRoom Service, the corresponding Subscriptions must be registered.



---

## Chapter 3. Understanding the MailRoom

The MailRoom provides an asynchronous solution for different types of standardized data interchange between various Business applications.

All the data processing is event driven, it is processed as it arrives. However, it is possible for an administrator to delay the sending by using the output scheduling facility, and thereby save network charges.

The MailRoom consists of all the functions and components in BTB that are involved in the process of receiving and sending data. These are the MailRoom layers:

- Source scenarios
  - Source exits
- Kernel
  - Security checking functions
  - Data preprocessing functions
  - Kernel exit
  - Output scheduling functions
  - Routing functions
  - Fanning functions
- Destination scenarios
  - Destination exit

When a transaction enters the MailRoom through one of the source scenarios, it can easily be *switched* to any one of the destination scenarios. The destination scenarios to which the transaction can be switched depends on the MailRoom registrations. Figure 7 on page 18 shows an overview of the MailRoom layers

## Overview of the MailRoom

Figure 7 shows an overview of the MailRoom components.

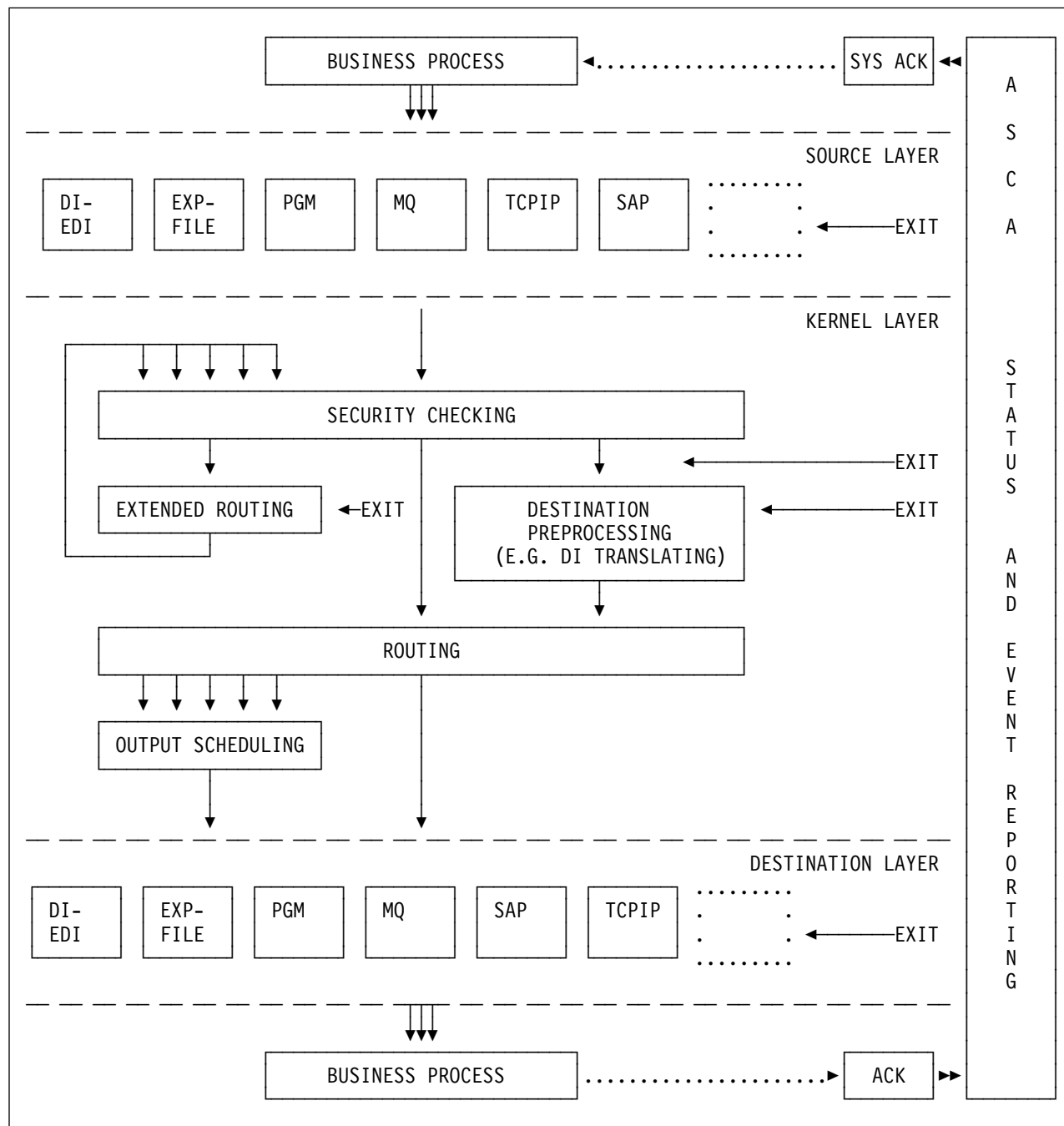


Figure 7. BTB MailRoom component overview

## MailRoom layers

The MailRoom is structured into a number of layers, each with a distinct responsibility.

## **Sending business process**

While not part of the MailRoom, the sending business process is the primary initiator of delivery services in MailRoom. It passes a business document to MailRoom and might later receive a System Acknowledgment when processing has completed.

## **Source Layer**

The Source Layer receives the passed document from the sending business process and if necessary converts it to MailRoom format (M- and D- records) via source exits (supplied or user written).

## **Kernel layer**

The Kernel Layer determines the actions to be performed on the received document based on the M-record and MailRoom service and subscription registrations. The document can be inspected or even converted to a different format by a Kernel exit (supplied or user written).

Normally a document is just sent to a single receiver but if necessary it can also be routed to multiple receivers. In Extended Routing it is possible for supplied or user written code to dynamically determine the proper receivers (one or more) based on the document contents or other factors.

When one or more receivers have been found and checked, Routing and Output Scheduling determines if the documents should be sent immediately or at a later point in time.

## **Destination Layer**

At this point the document is ready to be sent, but before doing so, it is possible to let a Destination exit (supplied or user written) inspect or convert the document. Then the document will be passed or sent to the receiver and MailRoom know from the registrations if it should expect acknowledgments from the receiving business process. If sending is not possible due to temporary transmission problems, MailRoom will later automatically initiate a retransmission.

## **Receiving business process**

Finally the document reaches its intended receiver which might generate a network or business acknowledgment after successful processing. The acknowledgment is returned to MailRoom and matched against previous status information about the document and it might also trigger a System Acknowledgment to be returned to the original sending business process.

## **Monitoring the processing**

From the point where the document is received in the source layer until the acknowledgment from the receiving business process has been received, MailRoom monitors the processing and takes *before and after* images of the document. Using monitoring panels, an administrator can watch the progress, view the document before and after an exit has made changes as well as manually initiate certain MailRoom actions like resend, restart, fail, etc..

---

## Terminology used in the MailRoom

This chapter provides an introduction to the terminology used in the MailRoom.

### Source and destination scenarios

These MailRoom components either receive data from a Business process (a source scenario), or send data to a Business process (a destination scenario). Different source and destination scenarios are available, one type for each type of Business process. It is possible to combine any source scenario with any destination scenario. This is done by setting up a MailRoom service, and creating subscriptions to the service for relevant Trading Partners.

### Abbreviations

Here are some commonly used abbreviations:

<b>BEC</b>	Back-end communication
<b>BPI</b>	Business programming interface
<b>DI</b>	DataInterchange
<b>M-record</b>	MailRoom record
<b>MDI</b>	Multiple document interchange
<b>MR-layout</b>	Transaction-type id, e.g. SWORDER.
<b>NPT</b>	Non-programmable terminal (3270 device)
<b>PWS</b>	Programmable workstation
<b>SDI</b>	Single document interchange
<b>TP</b>	Trading Partner

### MailRoom services

The purpose of a service definition is to combine one or more source scenarios with one or more destination scenarios. Together with the subscriptions (receive and send) the service completely describes the exact source and destination scenarios that exist for that service.

The subscription type on the service describes what kind of scenarios are allowed for the service. These are the subscription types:

- S** TP Sender—Inbound traffic  
A number of Trading Partners (with send subscriptions) can send a document to a Fixed TP Receiver defined on the service.
- R** TP Receiver—Outbound traffic (one-to-many)  
One fixed TP Sender can send a document to one of the Trading Partners (with receive subscriptions).
- T** TP to TP—third party traffic (many-to-many)  
One of the *From TPs* (with send subscription) can send to one of the *To TPs* (with receive subscription).
- X** Extended Routing (Unconditional, Rule-based and Exit-based Fanning)  
One of the *From TPs* (with send subscription) can send a document to the MailRoom.

For more information about subscription types, refer to the *System Administration Guide*.

## The M-record

The M-record is a MailRoom record with its own layout (MR-layout). The M-record is stored in the MailRoom databases, one M-record for each document received. When the requester invoking the MailRoom is a user-written program, it is the responsibility of the business process to provide an M-record with every document passed. When the requester is an application product, such as DataInterchange, the source scenarios will create an M-record for each document and store it with the data. The way to create an M-record depends on the subscription type and the source scenario.

In the case of DataInterchange being the requester calling a source scenario, the information in the DI control record will be used.

There are two types of M-record:

**Simple** Used when the subscription type on the service is *S*(sender) or *R*(receiver).

The keys necessary to interface with the MailRoom are:

- Country code
- Trading Partner number
- MR-layout

### Extended

Used when the subscription type on the service is *T* (Third Party Traffic) or *X* (Extended Routing).

The keys necessary to interface the MailRoom are:

- Country code
- To Trading Partner number
- From Trading Partner number
- MR-layout

These fields are the most important fields in the M-record. To a developer, the MR-layout describes the data that is parsed or received. To an administrator the MR-layout is used in different ways, such as:

- When defining an Expedite/CICS source, the MR-layout acts as the message user class
- For a DI destination, the MR-layout acts as the DI in-format

For more information about the M-record, refer to the *System Administration Guide*.

For more information about building an M-record, refer to the *Application Programming Guide*.

## Source envelopes

Everything that is passed to the MailRoom in one invocation (one API call) is considered to be in one source envelope. The source envelope is the *commit entity* when data enters the MailRoom.

Depending on the sending system, the source envelope can contain Multiple or Single Documents.

When the Trading Partner represents the source, the documents will be from the same Trading Partner, but they can have different format IDs and services specified.

Wherever possible the MailRoom supports MDI.

For more information about source envelopes, refer to the *System Administration Guide*.

## Derived envelopes

For every different MR-layout found in the source envelope, a *derived envelope* is created. If a source envelope consists of 10 documents, with five M-records indicating one MR-layout and the remaining indicating another, then two derived envelopes are created with five documents in each.

For more information about derived envelopes, refer to the *System Administration Guide*.

## Destination envelopes

For the destination scenarios it is relevant to group together many derived envelopes. This is done by having the Scheduler create *destination envelopes*, linking (optionally) many derived envelopes.

The grouping of derived envelopes differs, depending on circumstances:

- For the DI destination scenario, the destination envelope will contain all the data (*different MR-layouts*) for *one* Trading Partner.
- For the Batch destination scenario, the destination envelope will contain all derived envelopes with the *same MR-layouts*, but from *different* Trading Partners.
- For all other destination scenarios, the documents are group in destination envelopes where the Trading Partner, the layout and sender all are the same.

Documents in the same destination envelope will be sent from MailRoom in the same order as they are recieved by MailRoom

For more information about destination envelopes, refer to the *System Administration Guide*.

## Envelope ID

The envelope ID for a source envelope consists of a 12—byte alphanumeric field ending with three blanks.

The envelope ID for a derived envelope consists of the same first nine bytes as the source envelope it is associated with, suffixed with *001-mmm*.

The envelope ID for a destination envelope consists of the same first nine bytes as the source envelope it is associated with, suffixed with *nnn-ZZZ*. The scheduler continues with the first free suffix from the same suffix-range as used for the derived envelopes.

**Note:** If the scheduler finds more than one derived envelope that can be linked to the same destination envelope, the keys will differ.



For more information about the envelope ID, refer to the *System Administration Guide*.

## MailRoom priority

The priority level on the service is used to control the internal processing of documents in the MailRoom when many documents arrive at the same time.

Documents in the same priority class will be served on a *first come, first served* basis, on the entry to various MailRoom processes. This means that a high arrival rate might temporarily build a queue of documents waiting to be processed. This is normal processing, and the queue will disappear when the arrival rate goes returns to normal.

By specifying High priority on services with important documents, these documents will be processed by another set of processes and queues than the Normal and Low priority documents. These important documents will then be processed before other documents in the low priority queues.

For more information about the envelope ID, refer to the *System Administration Guide*.

## Extended Routing

The Extended Routing scenario lets you dynamically determine who should receive a document. Using extended routing, one or more Trading Partners can receive the same document. When a document is sent to more than one Trading Partner, it is called *Fanning*.

The Extended Routing Type on the service determines how the appropriate *To Trading Partners* are selected during the processing of a document.

All the possible *To TPs* must have a Receive subscription. The receive subscriptions can be of different types, that is, one can receive from MQ and another from MAIL or FAX.

For more information about the envelope ID, refer to the *System Administration Guide*.

## Requests

In the MailRoom online status dialogs a request is synonymous with a document. If a document is fanned to multiple destinations it will appear in MailRoom online status dialogs as one request for each destination (receiver).

## MailRoom Exits

As shown in the MailRoom overview (see Figure 7 on page 18), you can call a user exit at certain points during MailRoom processing.

On the MailRoom registrations you can specify that an exit should get control of a document during processing.

Exits are called to enable specific, application related processing to be performed while the data is within MailRoom responsibility, for example building routing information, data validation, data editing, and reconciliation.

When an exit is called by the MailRoom, application data will be made available in one or more CICS temporary storage (TS) queues. The exit can then read and potentially update the document.

### Source exits

The MailRoom source layer has two exit points which can be used to reblock or enrich a document before it is saved in MailRoom transport table. Selection of a source exit is depending on the identity of the sender (name/queue/id/etc.) and the source type. An exit is considered *specific* if it is only invoked for a single sender. An exit is considered *global* if it is invoked for all senders of a source scenario.

**Unpack Exit** Gets control during the source processing of a document, immediately after data is received and just before a Source Exit. The Unpack Exit should unpack the passed buffer into record format. A Source Unpack Exit can reject a document if the format is unrecognized. The Unpack Exit is available for the MQSeries based scenarios (MQ, SAP-MQ, TIE-MQ) where data is received as one long buffer.

**Source Exit** Gets control during the source processing of a document, immediately after data is received. A Source Exit can reject a document if the format is unrecognized.

While the two Source Exits can change data, the intended purpose is only to split the buffer into smaller records and to dynamically build the M-record.

The source exits are defined in the source exit table, see the *System Administration Guide*.

Both user written source exits and MailRoom supplied source exits can be used. For more information about MailRoom source exits, refer to the *Application Programming Guide*.

### Document exits

The MailRoom kernel and destination layer also have exit points called document exits which can be used to read and potentially update the document.

Exits are called once per Document (Request) in the Envelope.

**Kernel Exit** Gets control during the kernel processing of a document, immediately after the security check has been performed. A Kernel Exit can read or update the document and if necessary *FAIL* or *LOCK* it to stop further processing.

The exit can also insert up to five events as MailRoom meta data. These event records can potentially be used to store results of the exit processing. The events can be viewed using standard on-line dialogues.

**Destination Exit** Gets control during the destination processing of a document, immediately before further sending or translation. A Destination Exit can read or update the document and if necessary *FAIL* it to stop further processing.

The exit can also insert up to five events as MailRoom meta data. These event records can potentially be used to store

results of the exit processing. The events can be viewed using standard on-line dialogues.

### **Display Exit**

Gets control during on-line screen display before a specific version of a document has been selected for viewing. Such an exit is selected from a list at display time. The exit can reformat the document to give a better visual presentation of the document on the screen. The result is not saved.

A kernel or destination (document) exit is defined on the MailRoom service or subscription.

Both user written document exits and MailRoom supplied document exits can be used. For more information about MailRoom document exits, refer to the *Application Programming Guide*.

## **Extended logging**

Extended Logging is a way to select how much tracing information and events will be created by the MailRoom during processing.

These are the possible values:

- |                |   |
|----------------|---|
| <b>Blank</b>   | Normal logging. Only the most relevant events will be logged, and no processing trace will be created for Extended Routing. This is the recommended value for production. |
| <b>Verbose</b> | Extended logging. The MailRoom will create more events, and during Extended Routing, a processing trace will be created and saved.  |

## **MailRoom databases**

The MailRoom tables can be divided in three groups:

- |                         |  |
|-------------------------|--|
| <b>Transport Tables</b> | Contain the data passing through the MailRoom.   |
| <b>Status Tables</b>    | Contain control data. It is stored there for as long as the data is within the MailRoom responsibility. The online monitor dialog is built on these tables.  |
| <b>Audit Tables</b>     | Contain audit information logged in the source scenarios. Expedite/CICS and DataInterchange also log in these tables before data reaches the MailRoom. The audit reconciliation reports are built on these tables. |

## **MailRoom archiving**

The archiving function consists of five groups of logically-linked tables. One of the groups handles the Transport tables and for this group the archiving function can either delete data from tables and archive the deleted data in a sequential data set, or simply delete data from tables.

For the other four groups the data is deleted from the tables and archived in sequential data sets.

The five groups of tables are:

- MailRoom Transport tables group
- ASCA tables group

- Error Message table group
- Export table group
- Administration table group

For more information about the archiving function, refer to the *Installation Guide*.

---

## Chapter 4. Functionality of the MailRoom components

This chapter describes the functionality of these components:

- Source scenarios
- Kernel
- Destination scenarios

---

### Source scenarios

A source scenario in BTB is invoked from a Business Process, which passes data to the source scenario.

#### Format of input data

The data passed to a source scenario can be flat files, EDI documents translated to an in-house format, or documents in free format. Depending on the scenario type, one or multiple documents can be received upon one invocation. Documents received in the same invocation are regarded as one source envelope.

Envelopes can contain one or more documents, as indicated by M-records. The MailRoom does not modify the data. Logic is dependant upon a scan for M-records to split into MailRoom documents. Only application exits (source, kernel or destination) can modify the data.

#### Functions of the source scenarios

- Receive the data.
- Check and optionally execute source exits.
- Assign a source envelope ID for the passed data.
- Split the data into documents. How the splitting is done will depend on the source scenario type, which is specified either on the service or on the subscription.
- Store the documents in the MailRoom transport tables for further processing. When stored, each document will consist of one M-record and one or multiple D-records.
- Perform necessary logging used for reconciliation reports.
- Start the MailRoom kernel.

#### Source scenario types

These source scenarios are available to Business Processes accessing BTB:

<b>APPC</b>	Files received from APPC client (OS/2 or Windows or AIX)
<b>BATCH</b>	Batch input to MailRoom
<b>DI-EDI</b>	EDI from DataInterchange
<b>EXP-DOC</b>	Documents from Expedite/CICS
<b>EXP-FILE</b>	Single files from Expedite/CICS
<b>MQ</b>	MQSeries
<b>PGM</b>	Input from programs running in CICS environment
<b>SAP</b>	SAP/R3 input to MailRoom
<b>SAP-MQ</b>	SAP/R3 IDOCs via MQSeries
<b>TCPIP</b>	Files received from a TCP/IP client (OS/2 or Windows or AIX)

**TIE-IMS** Transactions from TIE running under IMS  
**TIE-MQ** Transactions from TIE running under IMS via MQSeries.

### **APPC**

A business application on the client calls the MailRoom write API with a file containing the document. The MailRoom write API reads the document from the file and starts communication with the MailRoom APPC receiver via SNA. The APPC receiver splits the data into one or multiple documents, which is inserted in the MailRoom transport tables and processed by the MailRoom.

Multiple document interface (MDI) is supported.

### **BATCH**

A Batch job can use the BTB MailRoom Batch Utility to feed documents into the MailRoom.

The Batch Utility inserts data into the transport tables for further logging and processing. Data inserted in one *run*, is regarded as one source envelope. A scanner is started by the Timer, catches the Batch envelopes, and triggers the MailRoom

The source envelope can contain one or more documents—each document must begin with an M-record followed by data records. The data records can be prefixed with any character except an *M*. The MailRoom does not modify the data in any way. Logic is dependant on a scan for M-records to split into MailRoom resets.

MDI is supported.

### **DI-EDI**

EDI envelopes are sent through Information Exchange to Expedite/CICS and further on to DataInterchange through continuous receive. In DataInterchange the envelopes are de-enveloped and translated to in-house format—each document in the envelope will contain one control record and many data records.

The MailRoom response program is called to get the envelope into the MailRoom transport table for further logging and processing.

The input standard for DataInterchange can be any type DataInterchange can translate (for example, EDIFACT or ANSI-X12).

MDI is supported.

### **EXP-DOC**

A file containing one or multiple documents is sent through Information Exchange to Expedite/CICS where a single or continuous receive profile is set to call a MailRoom response program to get the documents into the MailRoom transport table for further logging and processing.

MDI is supported.

## **EXP-FILE**

A flat file is sent through Information Exchange to Expedite/CICS where a single or continuous receive profile is set to call a MailRoom response program to get the file into the MailRoom transport table for further logging and processing.

The response program creates an M-record, and prefixes all records in the file with a *D*, as data records.

Single document interface (SDI) only.

## **MQ**

An MQ application (on any MQ-supported platform) can format a document into MailRoom MQ format and perform an MQPUT call to send a document through the local MQSeries queue manager to MailRoom. Here it is inserted in the transport tables and processed by the MailRoom.

MDI is supported.

As part of BTB a MQ batch utility can be used to format documents from a flat file in BTB file format and put them to an MQSeries queue, which can be set up to trigger the MailRoom. This utility allows for simpler transmission of documents from a remote MVS system.

For more information about the utility, refer to the *Application Programming Guide*.

## **PGM**

The PGM scenario is used by a Business application located in a local or remote CICS. The Business application performs an EXEC CICS LINK to the MailRoom write API to get a source envelope into the MailRoom.

MDI is supported.

## **SAP**

A MailRoom AIX program is running on a server scanning for files in a SAP directory. When a new file is detected, it is transmitted through TCP/IP to the MailRoom CICS, where it is recreated as a file of multiple documents, inserted in the transport tables and processed by the MailRoom.

MDI is supported.

## **SAP-MQ**

A MailRoom AIX program is running on a server scanning for files in a SAP directory. When a new file is detected, it is transmitted via MQSeries Link for R/3 to the MailRoom CICS, where it is recreated as a file of multiple documents, inserted in the transport tables and processed by the MailRoom.

MDI supported.

## **TCPIP**

The business application calls the MailRoom TCP/IP write API with a document in a file. The MailRoom API reads the document and initiates communication with the MailRoom CICS. The MailRoom TCP/IP receiver splits the document into one or multiple documents, which is inserted in the MailRoom transport tables and processed by the MailRoom.

MDI is supported.

## **TIE-IMS**

The MailRoom IMS response program is triggered when a Business application has performed a TIE-WRITE. The response program performs a TIE-READ, and inserts the documents into the transport tables.

For compatibility reasons, each envelope is split with an asterisk record.

MDI is supported.

## **TIE-MQ**

The MailRoom TIE-MQ IMS response program is triggered when a Business application has performed a TIE-WRITE. The response program performs a TIE-READ, and formats the document into MailRoom MQ format. It then performs an MQPUT call to send the document through the local MQSeries queue manager to MailRoom. Here it is inserted in the transport tables and processed by the MailRoom.

MDI is supported.

---

## **The kernel component**

### **Security checking**

- Check that Service/Trading Partner send subscriptions for all documents in the source envelope are correct.
- Create derived envelopes as necessary.
- Call kernel exits, if specified on the service, to make application related processing on documents. For example data validation and data editing.
- Copy documents if fanning applies.
- Insert a MailRoom request for every document.
- Update counters in the derived envelope for each document that passes or fails security check.
- Update counters on the source envelope for each derived envelope created.
- If a derived envelope is destined for DataInterchange translation, then start the preprocessing layer.
- Otherwise pass the derived envelopes to Routing.



## Preprocessing

In some cases *preprocessing* of documents will be necessary. For example, documents in envelopes destined for DataInterchange will be translated as they arrive at the MailRoom. They will later be enveloped and sent when the scheduled time expires.

**Note:** Documents will be passed for translation to DataInterchange one by one, and will be associated in DataInterchange by using the derived envelope ID as a DI—batch ID on the translate command.

For each translated document, DataInterchange returns an Audit file, and the MailRoom will save this file with the request. If a translation fails, a note is sent to SPOC and other documents in the derived envelope continue to be processed as usual.

The number of documents that were translated successfully is registered on the derived envelope. The number that were not successfully translated is also recorded.

When all documents are translated (good or bad), the derived envelope is passed to Routing for further processing.

## Kernel exit

If an application kernel exit is assigned, this processes the document.

## Routing

Checks if Output Scheduling is established for this Service. If not, the envelope is passed to the correct destination scenario.

If yes, it is passed to Output Scheduling.

## Output Scheduling

The schedule part checks if schedule time is due. If schedule time is due, the correct destination scenario is invoked.

If schedule time is not due, the derived envelope and related entries in the Transport Index table are marked with status *SCHDUL* and the specific time. Process is temporarily stopped or ended, waiting for the scheduling scanner to pick up and trigger the sending.

**Back-end availability:** The scheduling attribute can also be used to register expected (Service Level—agreed) back-end availability.

By implementing a *manual override* on the schedule attribute, an administrator can also temporarily stop a service, in case a temporary operational problem. The result is that a service, that *should* be operational and where the MailRoom would unsuccessfully try to access a back-end server that is *down*, is instead stopped *until further notice*. This will reduce the number of error messages sent to the SPOC.

---

## Destination scenarios

The destination layer consists of different destination scenarios, that is, the different ways and methods by which the data in the Transport tables can be passed or pushed onwards for further processing.

Based on the service receive subscription, the MailRoom assigns a schedule for all derived envelopes for the service. If no such schedule is assigned for a derived envelope, the documents will be sent to the destination immediately. If a schedule is assigned, the MailRoom will schedule the documents for the time frame specified in the schedule, and a schedule scanner will create destination envelopes and send the documents when the time is expired.

Before sending the document or envelope to the receiving Trading Partner the destination scenario will trigger a destination exit if one is defined on the service or subscription.

### Destination scenario types

These destination scenarios are available to Business Processes accessing BTB:

<b>APPC</b>	Documents to a APPC server
<b>BATCH</b>	Batch Output from the MailRoom
<b>DI-EDI</b>	EDI to DataInterchange
<b>EXP-DOC</b>	Files containing documents to Expedite/CICS
<b>EXP-FILE</b>	Files to Expedite/CICS
<b>FAX</b>	Documents to a FAX
<b>MAIL</b>	Document sent as e-mail or a file
<b>MQ</b>	Documents to Trading Partner via MQSeries
<b>NULL</b>	Documents to a dummy destination
<b>PGM-CICS</b>	Documents to a CICS program
<b>SAP</b>	Intermediate documents to SAP via TCP/IP
<b>SAP-MQ</b>	Intermediate documents to SAP via MQSeries
<b>TCPIP</b>	Documents to a TCP/IP server
<b>TIE-IMS</b>	Transactions to TIE on IMS systems
<b>TIE-MQ</b>	Transactions to TIE on IMS systems, sent via MQSeries

### APPC

The MailRoom APPC send program sends the document to the APPC MailRoom daemon on the server. The APPC MailRoom daemon writes the document to a file, which can be processed by a user application.

SDI only.

### BATCH

A Batch job using an BTB MailRoom Batch Utility reads the documents from the MailRoom. The documents stay *indefinitely* in the MailRoom until picked up by the Batch utility.

The documents are sent *as is* with a Control record followed by data records.

If the Utility is used, all documents, matching the search criteria given in a UTIL—member are saved in the flat file defined in the JCL.

MDI supported.

## **DI-EDI**

Documents are to be sent to an Information Exchange mailbox through Expedite/CICS and DataInterchange. The translation (and storage) into the DataInterchange repository is performed as *preprocessing*, immediately data enters the MailRoom. The enveloping and sending is performed according to specified schedule.

MDI supported.

## **EXP-DOC**

Files containing documents are sent to an Information Exchange mailbox through Expedite/CICS as a flat file. (The MailRoom will trigger Expedite/CICS according to schedule)

MDI supported.

## **EXP-FILE**

Files are sent to an Information Exchange mailbox through Expedite/CICS as a flat file. The MailRoom will trigger Expedite/CICS according to the schedule.

Before sending, the M-record and the first char (*D*) are removed from the data records.

SDI only.

## **FAX**

Data will be sent to the FAX number registered on the Service or Subscription, where an attention can also be specified.

An exit can be used to format the document into letter format if this has not already been done by the sender.

SDI only.

## **MAIL**

The data will be sent as e-mail or a file to the electronic address specified on the subscription. The address can be a traditional user ID at node ID, an Internet e-mail address or a Tools disk.

SDI only.

## **MQ**

The MailRoom in CICS initiates a transmission of the destination envelope to the MQ queue ID specified on the service or subscription. The MQ queue manager on the local CICS routes the document to the remote queue ID controlled by the remote MQ queue manager, which triggers the necessary local processes.

SDI only.

As part of BTB a MQ batch utility can be used to read the documents from the MQSeries queue and write them to a flat file in BTB file format. This utility makes it simple to read a document in BTB MQ format from a MQSeries queue on a remote MVS system and write the document into a flat file.

For more information about the utility, refer to the *Application Programming Guide*.

## **NULL**

The data will be processed by the MailRoom according to the specification on the service. The data will remain in the MailRoom and no send is performed. This is useful in testing, where you do not want the document to be send.

SDI only.

## **PGM-CICS**

A CICS program will be invoked by the MailRoom via a local or remote (as defined in CICS) CICS transient data (TD) queue.

The CICS program performs an EXEC CICS LINK to the MailRoom read API in order to get the document.

SDI only.

## **SAP**

A MailRoom AIX agent is running on a server listening to a TCP/IP port. The MailRoom in CICS initiates a transmission of the destination envelope to the IP address and port specified on the service or subscription (Socket Interface for CICS is used). The AIX program receives the documents, dynamically generates a file name and places the IDOC in a SAP directory on the hard disk and triggers SAP.

SDI supported.

## **SAP-MQ**

The MailRoom in CICS initiates a transmission of the destination envelope to the MQSeries queue ID specified on the service or subscription. The MQSeries queue manager on the local CICS routes the document to the remote queue ID controlled by the remote MQSeries queue manager located on the AIX system. The IDOC document is saved in a SAP directory on the hard disk and SAP is triggered.

SDI supported.

## **TCPIP**

A MailRoom OS/2 or Windows or AIX agent is running on a server listening to a TCP/IP port. The MailRoom in CICS initiates a transmission of the destination envelope to an IP address and port specified on the service or subscription (Socket Interface for CICS MailRoom agent is used). It will receive the documents, dynamically generate a file name and place the file in a directory on the hard disk. If specified in the setup the file name will be passed to a user application on the server.

SDI supported.

## **TIE-IMS**

A MailRoom IMS transaction is triggered with a document key. The IMS program reads the document from the Transport table and performs a TIE—WRITE. The envelope is written to TIE *as is* with one Control record followed by data records.

SDI only.

## TIE-MQ

The MailRoom in CICS initiates a transmission of the destination envelope to the MQSeries queue ID specified on the service or subscription. A system control structure is appended to the data, containing the TIE Application and BTX. The MQSeries queue manager on the local CICS routes the document to the remote queue ID controlled by the remote MQSeries queue manager, which triggers the IMS response program. This program reads the document from the MQSeries queue and performs a TIE—WRITE to the Application and BTX in the appended control record. The envelope is written to TIE *as is* with the Control record, the M-record and the data records.

SDI only.

---

## Automated proactive monitoring

The MailRoom has facilities to let you confirm that your data has reached its destination. During the registration phase, you can define a time-span indicating the length of time the MailRoom process can be expected to run for a particular transaction type (Service). Before this time expires, the transaction must have reached *final* status. If it has not reached *final* status, the transaction is considered to be *overdue* and an alert mail message is sent to the Service Point of Contact (SPOC). This facility is implemented by MailRoom background scanners constantly scanning the MailRoom tables checking time-stamps.

The registration dialogs allow you to choose between three *points-in-time* when the MailRoom should set the transaction to *final* status.

Here is an example using the DI-EDI scenario:

- Define the transaction to be completed when it has sent *OK* to the IBM Global Network (IGN) Mailbox. For this destination type this is the minimum level of MailRoom monitoring.
- Alternatively, if you and your Trading Partners agree that transactions you send should be retrieved by them within a specified time-span, you can have the MailRoom monitor the transaction until IGN reports that the Trading Partner has read the transaction.
- Or, your Trading Partners agree to send you functional acknowledgments. In this case, you can extend your time-span to include an agreed length of time for your Trading Partner to process the transaction. The MailRoom registers the status changes but does not set the transaction to *final* status until the *FUNACK/997* acknowledgment has been received.

---

## MailRoom acknowledgment settings

When you set up the services for data passing through the MailRoom, you must also decide how far the data sent from the MailRoom should be monitored. For some destinations it will be enough to know that the sending was successful, and for other destinations you will need to know if the data was successfully received and processed by the server application.

There are four acknowledgment levels:

- Level X** The document has been sent successfully from the MailRoom. This is only relevant for DataInterchange when sending to a Trading Partner that is using another value-added network (VAN).
- Level X lets the MailRoom set the transaction in terminal state (FINISH) in the MailRoom when the transaction reaches status *Send requested* in DataInterchange (that is, when Expedite/CICS has responded OK back to DI).
- Level 0** The document has been sent successfully from the MailRoom.
- For DataInterchange this means that the document has been accepted by the network.
- Level 1** The document has been received by the server application, but it has not yet been processed. For TIE-IMS and TIE-MQ, level 1 indicates that the document is in TIE.
- Level 2** The server application has processed the document and sent a final acknowledgment back to the MailRoom
- The final acknowledgment will be an indication of whether the document was processed successfully or not.
- When this level is specified, it is the responsibility of the server application to call the MailRoom acknowledgment API to finish the monitoring. For information about calling this API, refer to the *Application Programming Guide*.

---

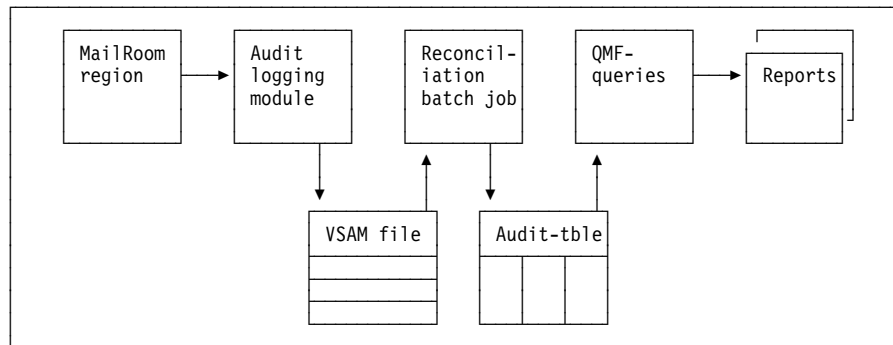
## Chapter 5. MailRoom Audit Facilities

---

### Audit logging and reconciliation

The Audit scope handles the receive and send process in the MailRoom region from the point where Expedite/CICS has completed a receive request for Information Exchange, to the point where the transaction is available for the Business application, or to the point where the Business application has finished processing the data. When to stop the Audit scope is decided by the MailRoom administrator, depending on the acknowledgment level set for the MailRoom service.

#### Overview



On an event driven basis, an audit logging module will be invoked at certain points during the process. Data is then stored in a VSAM data set.

The log records are read (using MVS batch JCL) and added to a set of DB2 audit tables.

The audit tables can then be used to perform reconciliations. There is a set of QMF procedures delivered with the package, which satisfy a basic set of requirements. When necessary, the delivered QMF procedures can be modified and amended according to your specific needs.

### Daily reconciliation

There are three reasons for executing the audit reconciliation procedures:

1. Detecting errors during data processing, such as loss of data before it reaches the MailRoom
2. Keeping track of data sent from the MailRoom to all back-end systems
3. Gathering statistical information

You should run this job daily for statistical purposes. If the purpose was to detect errors, you might wish to have the data as current data as possible. You can then run the job with shorter intervals during online processing. You can run this job in parallel with online processing, as often as required. The job will automatically continue with the first unextracted VSAM record.

The job is **KBASCDAY**.

## **Periodic reconciliation and cleanup**

Periodically the VSAM file will need to be reset. Job **KBASCPER** can be used for this purpose.



---

# Index

## A

Administration table group 26  
AIX  
    agent 29, 34  
    client/server access 1  
    program 29, 34  
    system 34  
APPC 1, 28, 32  
architecture 2  
archiving 25  
ASCA tables group 25  
asynchronous 1  
audit facilities 37  
auditing 8  
authentication 4

## B

BATCH 28, 32  
bibliography x—xi  
BTB  
    *See* Business Transaction Broker  
Business Transaction Broker  
    administering resources 9  
    administrator 17  
    architecture 2  
    audit facilities 37  
    auditing 8  
    changes in the release xiii  
    enhancements xiii  
    functionality of MailRoom components 27  
    introducing 1  
    logging 8  
    MailRoom acknowledgment settings 35  
    terminology 20

## C

changes in this release xiii  
CICS  
    EXEC CICS 29, 34  
    source scenario 27  
    temporary storage 24  
    TS queue 24  
CICS TS  
    *See* CICS  
CICS/ESA  
    *See* CICS  
client/server design 6  
CSP  
    application 12  
    call 12

CSP (*continued*)  
    tables 8

## D

data conversion 4  
DataInterchange  
    acknowledgment levels 36  
    audit file 31  
    logging 25  
    repository 33  
    translation 28, 30  
DB2  
    audit tables 37  
    QMF 37  
derived envelopes 22  
destination envelopes 22  
destination exit 24  
DI-EDI 33  
display exit 25  
document exit 24

## E

enhancements to BTB xiii  
envelope ID 22  
Error Message table group 26  
exits 23  
EXP-DOC 28, 33  
EXP-FILE 29, 33  
Expedite/CICS  
    audit scope 37  
    logging 25  
Export table group 26  
extended logging 25  
extended routing 23

## F

fanning 4, 23  
FAX 33

## I

IBM Global Network  
    asynchronous exchange using 1  
    destination type 35  
    reporting 35  
idoc 27, 34  
IE  
    *See* Information Exchange  
IGN  
    *See* IBM Global Network

IMB ix

## IMS

- program 34
- response program 30, 35
- systems 32
- TIE-READ 30
- TIE-WRITE 30
- transaction 28

Information Exchange 28, 29, 33, 37

input scheduling 5

Intelligent Message Broker ix

## J

JCL 37

## K

kernel exit 24

## L

layer 18

Link for R/3 29

logging 8

## M

M-record 21

MAIL 33

### MailRoom

- abbreviations 20
- archiving 25
- audit facilities 37
- components 17, 18
- databases 25
- derived envelopes 22
- destination envelopes 22
- destination exit 24
- destination scenarios 20, 32
- display exit 25
- document exit 24
- document formats 6
- envelope ID 22
- exits 23
- extended logging 25
- extended routing 20, 23
- functionality of components 27—35
- kernel exit 24
- layer 18
- layers 17
- M-record 21
- overview 18
- priority level 23
- requests 23
- services 20
- source envelopes 21

### MailRoom (*continued*)

- source exit 24
- source scenarios 20, 27
- terminology 20
- Trading Partner receiver 20
- Trading Partner sender 20
- Trading Partner to Trading Partner 20
- unpack exit 24

MCO 8

message repository 4

message repository 4

middleware 1

migrating BTB

MLS 8

modularity 6

### MQSeries

Link for R/3 29

MailRoom format 29, 30

queue manager 29, 30, 33, 34, 35

receive subscription 23

multi country operation

See MCO

multi language support

See MLS

multiple access modes 5

## N

national language support

See NLS

navigation 5

network logging 5

network protocol conversion 5

network transparency 5

NLS 8

notification 5

NPT development environment 6

NULL 34

## O

OS/2

agent 34

client/server access 1

output scheduling 5

overview 18

## P

panels

PGM 29

PGM-CICS 34

proactive monitoring 5

profiling 4

## Q

QMF 5, 37

## R

R/3 29  
RACF 4, 14  
requests 23  
routing 4

## S

SAP 34  
    destination scenario 29  
    directory 29  
    IDOC 27, 34  
    MQSeries SAP Link for R/3 29  
    R/3 27, 29  
    source scenario 27  
    trigger 34  
    via MQSeries 32  
    via TCP/IP 32  
security 4  
sending comments xi  
services 3  
source envelopes 21  
source exit 24

## T

TCP/IP 1  
TCPIP 30, 34  
temporary storage 24  
terminals  
    3270 1  
    emulated 1  
    non-programmable 1  
    NPT 1  
TIE 6  
TIE-IMS 30, 34  
TIE-MQ 35  
Transaction Server  
    See CICS  
transport independence 3  
transport tables 25  
transport tables group 25  
TS queue 24

## U

unpack exit 24

## V

VSAM  
    audit trail 37

VSAM (*continued*)

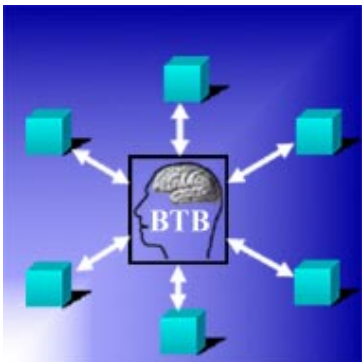
    cleanup 38  
    daily reconciliation 37

## W

Windows  
    client/server access 1  
Working Criteria table



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.



SC27-1584-01

