

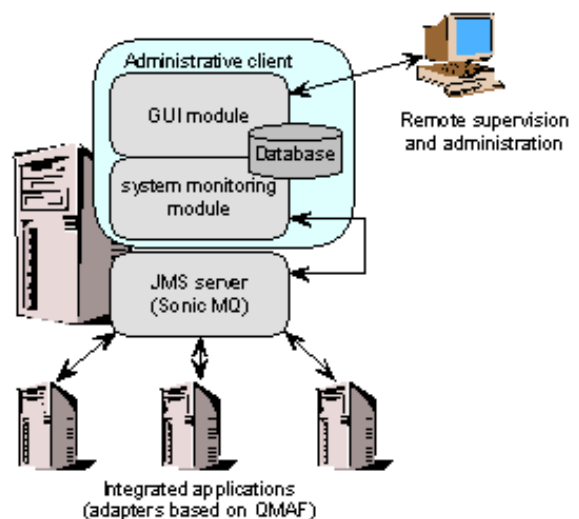


Using *JMS* technology (*Java Messaging Services*) in integration projects (EAI and B2B) leads to situations when e.g. older systems don't support the JMS standard, or there is a good reason to avoid using their functions directly.

In such cases it's necessary to create a *messaging client* as an external program (*adapter*) and link it with the host application. Ad hoc development of such *adapters* from scratch not only extends the project and adds to its cost, but at the same time makes the project difficult to maintain and hard to extend.

**QMAF** (*Qbizm Universal Messaging Adapter Framework*) is a specialized component framework for fast deployment of JSM adapters to virtually any application. Besides the JMS support, **QMAF** provides quality administration interface for comfortable maintenance.

With the help of **QMAF** it's possible to reduce the labor and time consumption needed for the development of specialized adapters from original man-weeks of programmer labor merely to hours of configuration. Thanks to the Java platform it's possible to use **QMAF** in virtually any environment ranging from Fox Pro to SAP. This is quite useful especially in pilot projects where time of deployment and its cost is crucial.



The **QMAF** component framework consists of reusable components, a container and an XML configuration file. The **QMAF** adapter gets linked with the host application by one of the following methods:

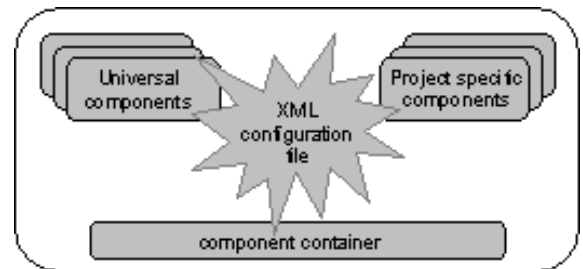
1. Synchronous communication  
*In this case, the adapter uses one of the synchronous communication protocols (RMI, CORBA, SAP/R3, RFC etc.).*
2. Shared database  
*This method can be used if the host application uses a client/server database for persistent data storage and makes it available also for the **QMAF** adapter.*
3. Filesystem  
*This method takes the advantage of filesystem for data exchange. **QMAF** adapter stores its received data into files on a disk where the host application can gain access to them and vice versa. In the simplest scenario, the adapter doesn't even have to be constantly running and can be invoked only in the moment when data sending or receiving is required (e.g. periodical initiation by cron or scheduler). This method is the simplest and to some extent depreciates features of Java messaging, however it's useful in situations where no other method is possible*



**QMAF** container based on the XML configuration file manages initialization of individual component, transfer of configuration parameters, their lifecycle and allows their mutual interaction. The components themselves secure individual executive functions of the client. Most of the components are general “reusable” - in the case of more complex situations, specialized components can be created in Java as required by the integration project.

Particular JMS adapter needed for a specific deployment is “assembled” from individual configurable components. **QMAF** distribution covers e.g. the following reusable components:

- Initialization of connection to JMS server SonicMQ (with the recovery function useful in the case of lost connection to the SonicMQ server).
- Reception of JMS messages from Topic and Queue.
- Dispatch of JMS messages to Topic and Queue.
- Reception of e-mails from a POP3 mailbox and an IMAP server.
- Dispatch of e-mails using SMTP.
- Component for connection management to a database (RDBMS).
- Message content storage and load from/to files in filesystem.



**QMAF** framework is a fully compliant Java application, which yields to easy deployment on a wide variety of systems.

#### **QMAF Administration client**

secures monitoring of the whole system of message exchange allowing to detect problems occurred during its operation. The fundamental functions of administrative client cover:

- Monitoring and logging messages traversing through the system (storing of headers, or their bodies in a database).
- Monitoring of undelivered messages and automatic notification of appropriate individuals in case of undelivered message occurrence (by e-mail, SMS etc.).
- Monitoring of operational state and occurred errors in individual adapters in integrated system.

Administrative client consists of two main parts:

1. **System monitor module** – monitors sent/received messages and logs their information into database. Based on this information performs other functionality (monitoring of undelivered messages, error notification to appropriate person etc.)
2. **GUI (Graphical User Interface) module** – allows system administrators to perform operations on data available in the *System monitoring module*, configure its parameters or parameters of other individual adapters and of the whole integrated system, check information regarding operation of individual adapters etc.

*System monitor module* is built in the **QMAF** environment from specialized components and it's therefore rather easy to add new functionality to it based on the current needs.

*GUI module* is built on the bases of internet technologies (HTML, HTTPS), thus its possible to use it in a standard web browser environment on any computer in the network.

**QMAF** uses a standard Java database driver to store its data, it's therefore possible to use any database supporting JDBC (Java DataBase Connection) as e.g. Oracle, Sybase, Progress, IBM DB2 etc..