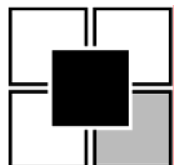
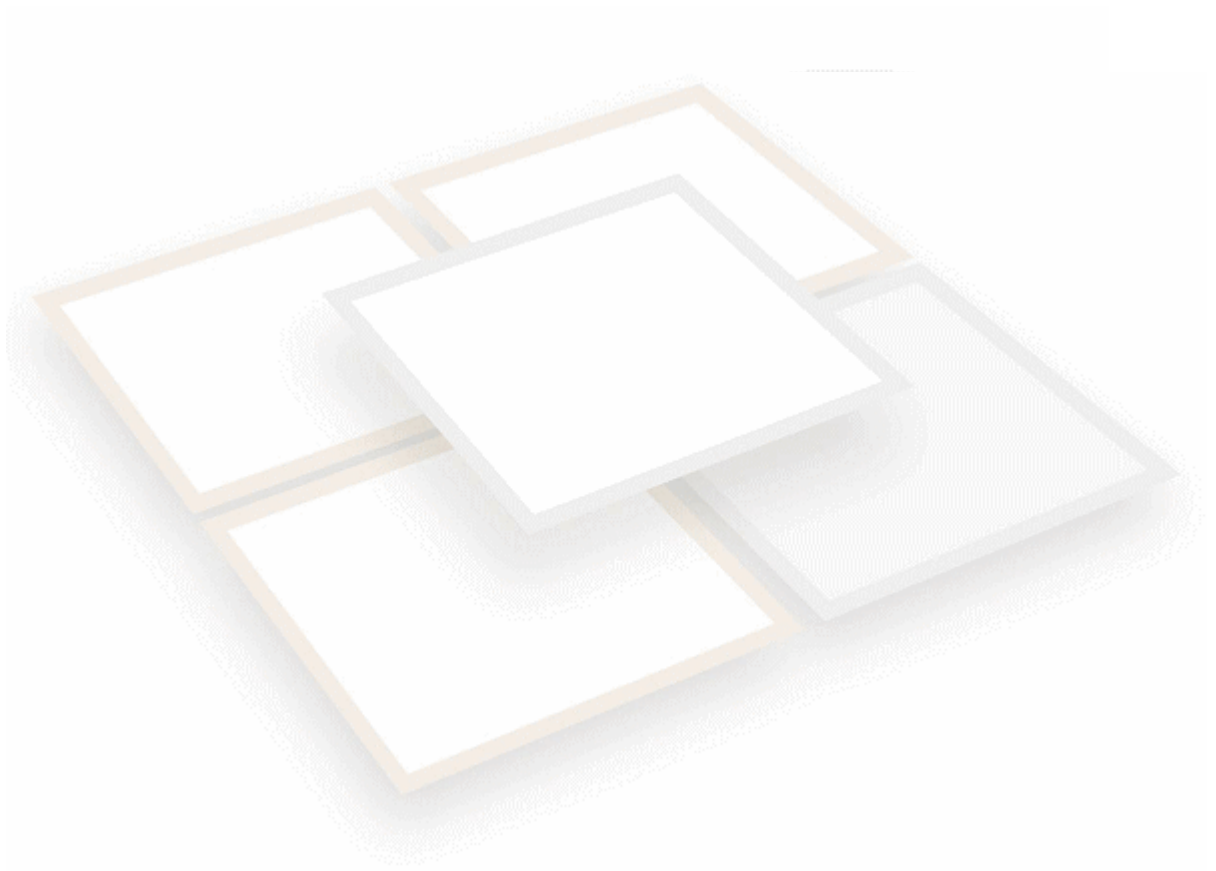


SPAZIO File Extender for
WebSphere Business
Integration Brokers

General Information
Manual

EMAFEM001/01

November 2003



SPAZIO File Extender for WebSphere Business Integration Brokers: General Information Manual

Date of issue	Reference number	Brief description
November 2003	EMAFEM001/01	First Edition - Draft 9

Copyright © 2003 Primeur Group. All rights reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any other language in whole or in part, in any form or by any means, whether it be electronic, mechanical, magnetic, optical, manual or otherwise, without prior written consent of Primeur Group.

Primeur Group may revise this publication from time to time without notice. A new release of this manual contains changes made to the product since the previous version.

The software product that this manual documents is the exclusive property of Primeur Group. The use of this software is governed by the license agreement that accompanies the product. The following conditions must be observed in all cases:

- The product may be used only on the number of computers for which the client is licensed.
- The client may make only one copy of the product, and this only for backup purposes.
- The client may not reverse engineer, decompile, or disassemble the product.
- The client may not loan, rent or lease neither the product, nor any of the documentation or user manuals related to the product, whether this is for free or for a fee.

Primeur Group warrants that the product will perform substantially in accordance with the accompanying product manual(s). Primeur Group disclaims all other warranties either expressed or implied. Primeur Group and its suppliers shall not be liable for any damages whatsoever (including damages for loss of business profits, business interruption, loss of business information or other pecuniary loss) arising out of the use of, or inability to use, the product.

SPAZIO, SPAZIO Messaging & Queuing, SPAZIO M&Q, SPAZIO File Transport, SPAZIO Data Extract, SPAZIO Legacy Interface, SPAZIO Data Secure, SPAZIO Data Compress, SPAZIO Software Control and THEMA are trademarks of Primeur Group. Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

Company Headquarters

Primeur Group
Corso Paganini, 3
16125 Genova
ITALY

Tel: +39 010 27811
Fax: +39 010 2781239

Local Agent

--

About this manual

Purpose

This manual describes some general information of *SPAZIO File Extender for WebSphere Business Integration Brokers (SPAZIO FE)*.

It describes the characteristics and functionality of SPAZIO FE, i.e. a series of components that can be used to enable file handling within IBM WebSphere Business Integration Brokers.

Reader

This manual is aimed primarily at those whose task it is to plan for using the product.

Knowledge of WebSphere Business Integration Brokers is required.

Area Covered

The manual is organized as follows:

- *Chapter 1: Overview* is a brief introduction to the type problems addressed by **SPAZIO FE**.
- *Chapter 2: Product Architecture and Components* provides information on **SPAZIO FE**.

Recommended Use

It is suggested that all SPAZIO FE and WebSphere Business Integration Brokers planners and administrators should read this manual.

Conventions

Product names

We will use the term *WebSphere Business Integration Brokers (WBI Broker)* to indicate the whole family of broker included in IBM business integration offering, including:

- WebSphere MQ Integrator 2.1
- WebSphere MQ Integrator Broker 2.1
- WebSphere Business Integration Message Broker 5.0

When a more specific identification of the product is needed we will either use the full product name and possibly the version (e.g. *WebSphere Business Integration Brokers 5.0, WBIMB 5.0*), or only the name indicating the product “generation”, for instance *WebSphere MQ Integrator (WMQI)* will be used to indicate the two products:

- WebSphere MQ Integrator 2.1
- WebSphere MQ Integrator Broker 2.1

It is assumed that all the information provided in the book does **not** apply to these products:

- WebSphere MQ Integrator Event Broker 2.1
- WebSphere Business Integration Event Broker 5.0

Typographical conventions

The following are common styles that may appear in the manuals. Typographical styles change the appearance of certain areas of text. The instances where different styles are used and what these styles mean is explained below:

Italics

Cross-references to other manuals appear in italic font. For example: *SPAZIO FE Repository: Developer's Guide*.

Courier font

This style illustrates system code or text. It may be input that needs to be entered by the programmer exactly as illustrated or output from the system. For example:

```
correlation id
```

Text in Courier font surrounded by a box with shading represents a listing.
For example:

```
* PRINTOUT QUEUED *
```

Text in Courier font surrounded by a box, without shading represents a screen. For example:

```
PRIMEUR S.R.L. ----- S P 1 0 ----- SPAZB -----
      FUNCTION ==>  _
                                     Userid : PRIAMPO
                                     Date   : 02/02/21
                                     Time    : 14:40

      0 PARAMETERS                   Defines User and System parameters
      1 QUEUES                        Defines User Queues
      2 CLASSES                       Defines Classes
      3 FILE UTILITIES                File Utility functions
      4 MESSAGE UTILITIES             Message Utility functions
      5 TRANSFER                      Transfers Files to/from Q_Manager
      6 LISTS                         Defines Distribution Lists
      7 TRANSMISSION MONITORING       Monitors Transmission Status
      8 SYSTEM LOG INQUIRY            Displays System Log data
      9 ACTIVITIES CONTROL            Activities control

      M PRODUCT MAINTENANCE           Inquiries/lists installed PTFs
      T TUTORIAL                      Displays TM10 manual
      X EXIT                          Quits TM10 transaction
```



Text that appears in this format is important information and should be read in all cases.

About this manual

Contents

About this manual	i
Purpose	i
Reader	i
Area Covered	i
Recommended Use	i
Conventions	ii
Chapter 1 Overview	3
1.1 Overview of message and integration brokers	3
1.2 Overview of IBM Business Integration offering	5
1.3 Spazio FE: unified brokering for files and messages	6
Chapter 2 Product architecture and components	9
2.1 Spazio FE at a glance	9
2.2 Spazio FE Repository components a glance	11
2.2.1 File management with SPAZIO FE Repository	12
2.2.2 SPAZIO FE Repository features and functionality	12
2.2.3 User Perspective with Spazio FE	13
2.2.4 Workflow in Spazio FE	14
2.2.5 What you require	15

Contents

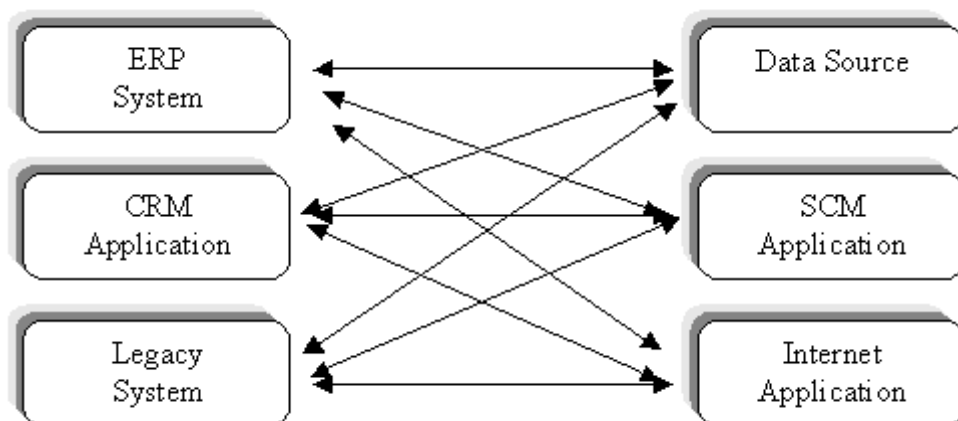
Chapter 1 Overview

This chapter discusses broker technology and some of the concepts that have led to the development of modern infrastructure brokers, such as the those that are part of *WebSphere* suite from IBM. Following this, it discusses **Spazio FE** and the motivational reasons for its development.

1.1 Overview of message and integration brokers

"An enterprise nervous system requires middleware services to manage the flow of data between its constituent parts. An integration broker runs in the middle of these services to perform data transformation and intelligent routing".

The terms Middleware, MOM, EAI, and Integration Brokers are all used to describe the concept of connecting applications together and enabling the systematic exchange of data. However, they each describe different technologies and concepts and cannot be used interchangeably.



The above diagram illustrates how "Application to Application Integration", commonly resulted in 'spaghetti syndrome' before the advent of integration software and tools: if applications needed to exchange data, it was manually re-keyed, exchanged via specific software modules on a peer-to-peer basis or performed using files. Although files may sometimes be appropriate or even necessary as a method of data exchange, as a general integration method they lack the immediacy and granularity of a message.

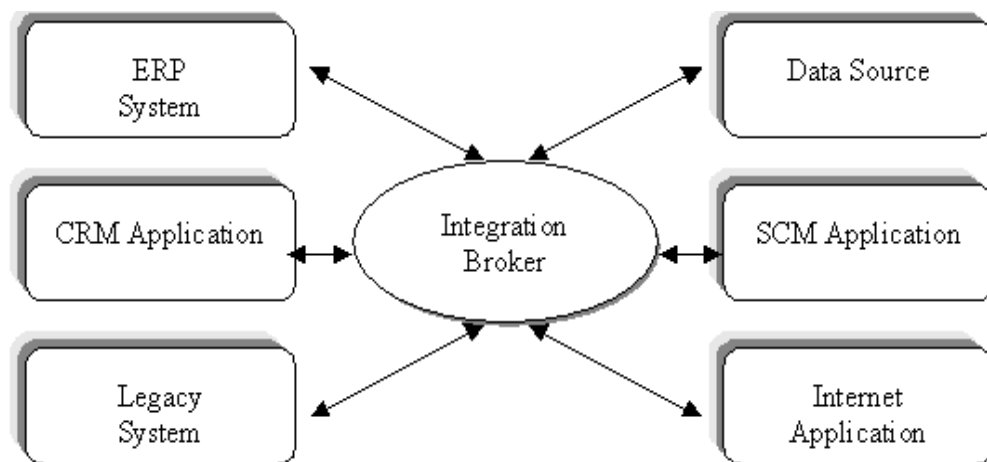
Early attempts at integrating applications using low-level mechanisms such as RPC or native TCP sockets were little more than (often unreliable) basic transport systems. IBM laid the foundations of modern (message-based) middleware with the introduction of MQSeries, known today as *WebSphere MQ (WMQ)*.

Middleware

Middleware is a general term for any programming that serves to "glue together" or mediate between two separate and often already existing programs, typically through the provision of messaging services.

Enterprise Application Integration (EAI)

EAI, on the other hand, is the systematic tying together of disparate applications, through the use of middleware, along with plans, methods, tools, adapters and business rules. Middleware is therefore one small part of an EAI solution.



Integration Broker - middleware plus!

Message Orientated Middleware (MOM)

Early middleware systems adopted synchronous communications paradigms, where an application that sends data is blocked until the data is delivered and/or a response is received. However, the fact that most applications do not actually need an immediate response led to the emergence of *Message Orientated Middleware* (MOM), queuing software that uses messages (byte-sized units of information that move between applications) as a mechanism to move information from point to point. MOM relies on an asynchronous paradigm, where the sending application dispatches a message to a queue manager, which guarantees that the message is delivered to its final destination. Messages returning to the calling application are handled when the calling application finds the time.

The most common example of MOM in use today, is of course IBM's WMQ.

1.2 Overview of IBM Business Integration offering

The first “middleware” developed by IBM, *MQSeries*, was a MOM offering that provided a message transport system enabling the movement of data between applications. It included none of the transformation, flow, adapters or other business tools that characterize a modern EAI solution.

The strength of *MQSeries* lies in its ability to allow applications to communicate asynchronously. Sending data consists of putting a message in a queue, receiving data consists in getting a message from a queue. The transfer of a message from one queue to another is handled by the middleware, allowing the communicating applications to proceed at their own pace and without interruption. Methods such as prioritization, load balancing, and thread pooling can optimize performance.

Today, *WebSphere Business Integration (WBI)* is a very different product from that early MQ offering. WBI Broker is a true EAI solution offering ready-made application adapters, message management and delivery, routing, transformation and business workflow. As with all EAI solutions, WBI Broker provides four basic functions. These can be categorised as:

Workflow Management	Business process execution, organization based routing, work lists, monitoring
Loose Coupling integration	System adapters and business records transformation
Message Broker	Physical routing, management and resource tracking
Asynchronous messaging	Message queue management

Here, we are mostly concerned with the lower three layers, since it is these that Spazio FE addresses for files.

1.3 Spazio FE: unified brokering for files and messages

With WBI Broker providing full EAI capabilities, one might ask, "where does **Spazio FE** come into this?" Files and batch based applications; every company has them and these seem to have been left out of the equation. You've invested in Websphere for your real-time applications but the file-based ones don't fit in.

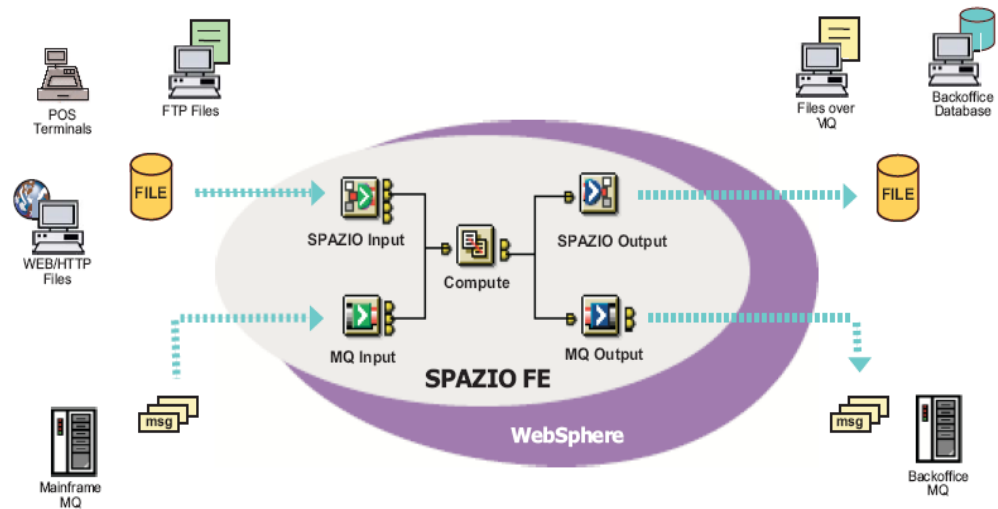


Figure 1: SPAZIO FE: Enabling File Brokering via WBI Broker

More often than not they're left out, unable to take advantage of your WBI Broker investments. With an estimated 50% of all data being exchanged through files, Primeur believe it is time this oversight was corrected.

Spazio FE provides plug-in nodes for *IBM's* WBI Brokers. Coupled with advanced file transfer and management systems, **Spazio FE** enhances Websphere Business Integration beyond pure messaging into a complete brokering solution. Moreover, with **Spazio FE** file routing, content enrichment and transformation can all be performed within an existing WebSphere Integration installation.

Using the four-tier scheme and adjusting for **Spazio FE**, we can therefore show the following benefits of using **Spazio FE** with *WebSphere Business Integration Broker*:

Workflow Management	
Loose Coupling integration	<ul style="list-style-type: none"> ■ Aggregation of messages into files according to rules specified in message flow ■ Splitting of files into messages ■ Routing of files based on header, context or content ■ Transformation and mapping of file records ■ Uses standard WBI Broker tools
Message and File Broker	<ul style="list-style-type: none"> ■ Mix and match file and message information sources in a seamless fashion ■ Provides a central point of control and management for all brokering functions ■ Provides a mechanism for transferring files over MQ or using industry standards such as FTP, HTTP etc
Asynchronous messaging and File Queuing	<ul style="list-style-type: none"> ■ Store and forward communication of files AND messages ■ Holds the file in a queue until the recipient formally accepts it ■ Assures the file is recoverable and is received once and once only

Chapter 2 Product architecture and components

2.1 Spazio FE at a glance

Spazio FE is a tool set of plug in nodes, extending the capabilities of base WBI Broker, **Spazio FE** is fully integrated with WBI Broker tooling and leverages the advanced file management and data moving features of **Spazio File Transfer** product line.

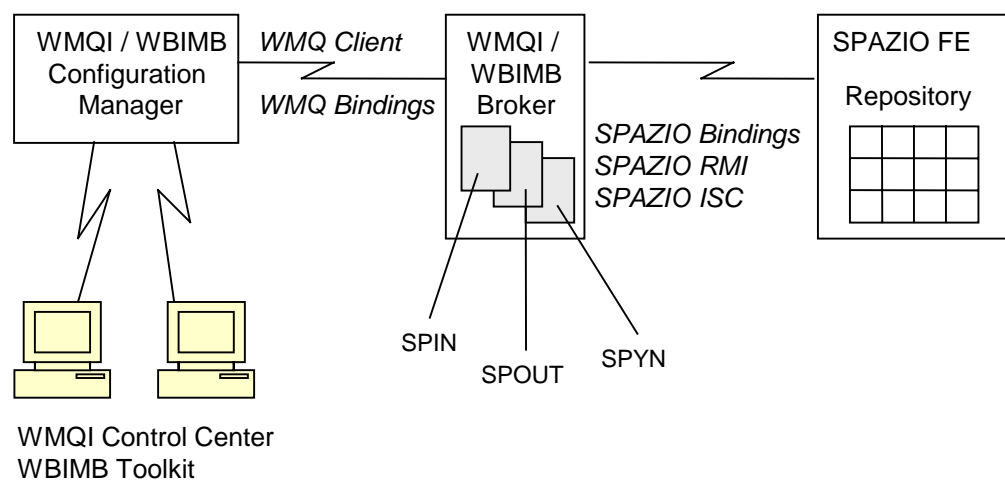


Figure 2 - Spazio FE components

As you can see in *Figure 2*, **Spazio FE** can be divided into two main categories:

- *Tooling components*: this is a set of icons and static definition files that must be installed on all the workstations using the GUI tooling provided by WBI Broker (e.g. WMQI 2.1 Control Center application) and extend the palette of nodes available to the message flow developer.
- *Broker components*: this is a set of executable components and prerequisite products that actually enable file handling from within the broker.

No **Spazio FE** components will be installed on the Configuration Manager.

Let's now have a closer look to the broker components. The standard mechanism to extend WBI Broker functionality is to provide custom written artifacts compliant with IBM defined interfaces. **Spazio FE** comes with three such nodes:

- Spazio Input Node (*SPIN*): this node reads business information from a file and passes its contents to a message flow.
- Spazio Output Node (*SPOUT*): this node receives business information from a message flow and stores it onto a file.
- Spazio Proxy Node (*SPYN*): this is a helper node that must be used in conjunction with *SPOUT* in all message flows rooted in input nodes other than *SPIN* (e.g. IBM provided MQ Input node).

Spazio FE nodes do not read/write files directly on the local file system, but use the value-added queued file repository (**Spazio FE Repository**) derived from **Spazio FTFI** file transfer product (**Spazio FE Repository** is bundled together with **Spazio FE**).

There are several ways to insert and/retrieve/browse files in this **Spazio FE** repository, please refer to *Section 2.2: SPAZIO FE Repository components at a glance*.

Spazio FE nodes access **Spazio FE** repository using *Spazio JMS* provider (*SPJMS*), that implements an extension of the point-to-point JMS 1.1 specification suited for file handling (rather than message handling as in "classic" JMS).

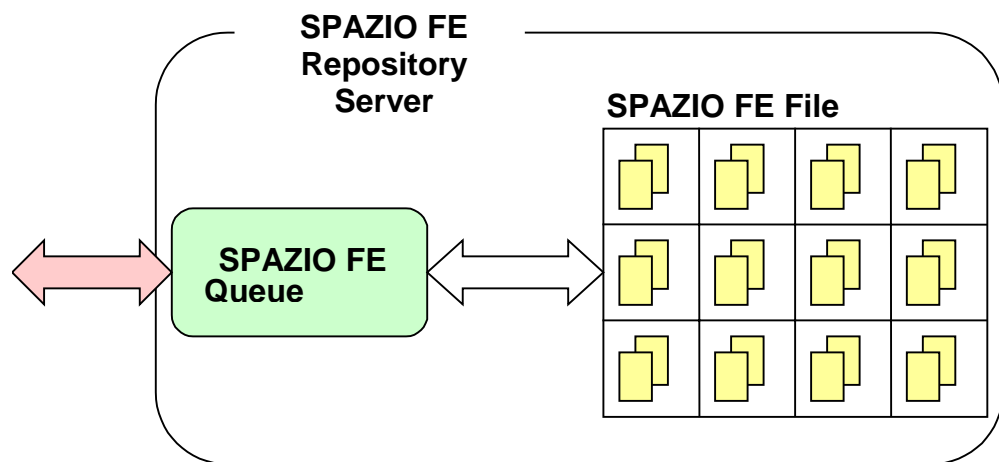
SPJMS is installed together with **Spazio FE** core components, and in most scenarios it will be a transparent component to the **Spazio FE** user; anyway there are situations (e.g. when the broker machine is remote respect to **Spazio FE** repository) where a basic administrative knowledge of the component is needed.

To learn more about *SPJMS* please refer to the relevant section of *SPAZIO FE User Guide* dealing with *SPJMS*.

2.2 Spazio FE Repository components a glance

As already mentioned **Spazio FE Repository** derives from repackaging of some modules of **Spazio FTFI** product line, for providing the file handling capability to **Spazio FE**. If you are interested in knowing the full functionality of **Spazio FTFI** please refer to the relevant manuals. In the following we will describe only the components embedded in **Spazio FE**.

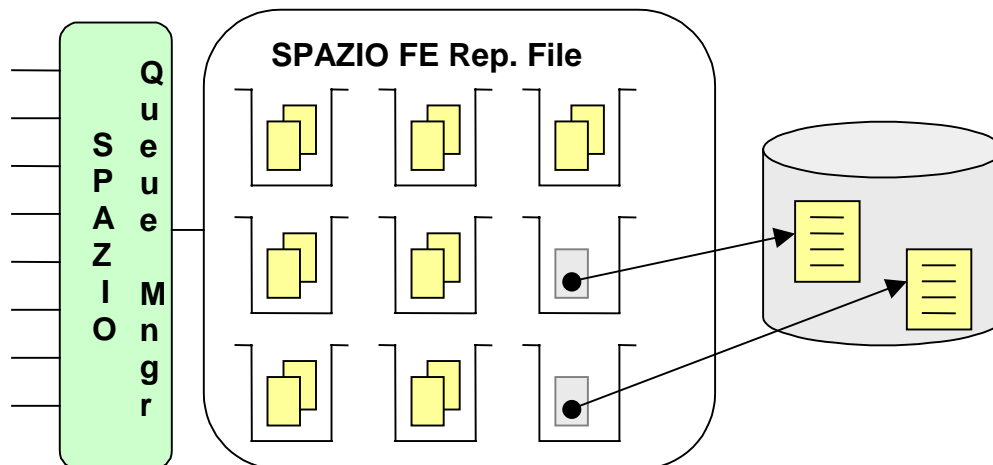
Spazio FE Repository Server is a key component of **Spazio FE**. In the context of **Spazio FE**, it is the managed file repository accessible for read and write operations from WMQ message flows.



As the above diagram illustrates, **Spazio FE Repository Server** provides a repository for storing files in an ordered way and a *Queue Manager* that provides applications with access to the files stored in the repository.

2.2.1 File management with SPAZIO FE Repository

The following diagram illustrates the basic characteristics of the **Spazio FE** repository:



Files are stored in a series of user-defined *queues* (just as WMQ messages are held in message queues). When you put a file in a **Spazio FE** Repository queue, you can choose to write the actual file data into the queue, or alternatively (for example in the case of very big files) to write only a pointer to the file in the filesystem. In this way, **Spazio FE Repository** provides an abstraction of the filesystem to which it refers, independently of the platform concerned. Access to files is handled by a *Queue Manager* module – again a familiar concept if you are used to dealing with WebSphere MQ.

2.2.2 SPAZIO FE Repository features and functionality

Here we will list some of the features of **Spazio FE Repository** that are particularly important in the context of **Spazio FE**.

- Virtual and physical storage of files in queues: as mentioned previously, **Spazio FE Repository** allows you to store files in queues *physically* (in which case the file records are actually written to the queue) and *virtually* (in this case, specific file information is stored as a pointer in the queue, while the actual file remains on disk in the file system)
- File attributes: once a file enters the **Spazio FE Repository** system, a wide range of attributes can be associated with it – many of which correspond in name and nature to WMQ message attributes (for example, *correlation ID*). These attributes can specify how the file is to be handled by **Spazio FE Repository**, and can often be passed as filters or search criteria by applications accessing files via the Queue Manager. In addition you can add user-specified attributes.

- File expiration management: **Spazio FE Repository** file attributes allow you to limit the lifetime of files in queues by specifying expiration times.
- File archiving and recovery: **Spazio FE Repository** allows you to request archiving and recovery of files put in queues
- Versioning: **Spazio FE Repository** automatically manages different versions of the same file, even with the same name
- Trigger management: the System Administrator is able to schedule on-line or batch procedures based upon the occurrence of one of the following events for a specified **Spazio FE Repository** queue:
 - Every time a new file is written to the queue
 - Every time a new file which has a specific identifier and belongs to a particular user class is written to the queue
 - Every time all files defined in a table and belonging to a common user class are written to the queue
- Communication capability: using **Spazio FE Repository**, files can be transferred using a wide range of standard or widely used protocols and products such as:
 - HTTP(S)
 - SMTP S/MIME
 - FTP(S)
 - SNA
 - WMQSeries
 - Connect:Direct
 - Netview FTP
 - Netmaster

2.2.3 User Perspective with Spazio FE

There are two main aspects to using **Spazio FE**:

- Configuring and maintaining the system (Queue Managers, queues etc) so that it can be accessed from WMQ message flows via the **Spazio FE** SPIN and SPOUT plug-in nodes. This is typically a system administrator task.
- Putting source files into **Spazio FE Repository** queues for consumption by WMQ messages flows, and retrieving files produced by WMQ message flows from the destination queues to which they have been routed (by the WBI Broker via the **Spazio FE** SPOUT node).

All the tasks related to the above can be performed via the **Spazio FE Repository** user and administrator interfaces that are supplied with **Spazio FE** (command line, character-based, GUI etc.). For more detailed information you can refer to the relevant user and system administration manuals.

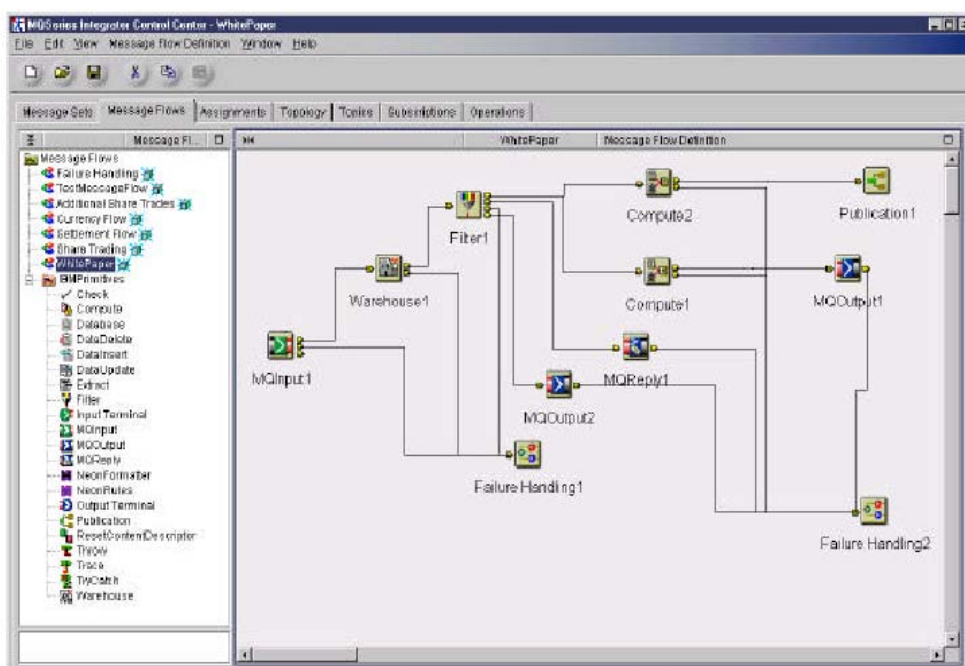
2.2.4 Workflow in Spazio FE

The **Spazio FE** plug-in nodes (SPIN, SPOUT, SPYN) provide the necessary interface for allowing WMQ message flows and brokers handle files. **Spazio FE Repository** Server is on the other hand the component that stores and manages access to the source and destination files consumed and produced by WMQ message flows via SPIN, SPOUT and SPYN nodes.

The combination of **Spazio FE** nodes and the **Spazio FE Repository** supporting technology and infrastructure allow you extend existing message flows to include file reading and writing capability.

Extending message flows to consume files from SPAZIO FE

Suppose you have a message flow that takes WMQ messages in input, via an MQ Input node. The format of these messages and the transformation and routing is defined for the WBI Broker. Suppose that for whatever reason you also have, or wish to have, a file-based application that outputs files with a record structure identical to the structure of these MQ messages. What you would like to achieve is to feed these file records as messages to your already designed message flow, and have the broker deal with them according to the specifications of the message flow, exactly as though it was handling the WMQ messages. **Spazio FE** allows you to implement exactly this. Typically your application will write its file to a **Spazio FE Repository** queue. Your message flow gets its input from the file via a **Spazio FE "SPIN"** node (instead of the MQ Input node it used for reading WMQ queues). From this point on, everything proceeds as with the WMQ messages. All the relevant existing business rules and transformation, routing etc. are applied. All that has changed is the source of the data: instead of a WMQ message queue, the source is a **Spazio FE Repository** file queue.



Extending message flows to produce files for Spazio FE

Supposing you wish to route the output of an existing message flow to a **Spazio FE Repository** file, **Spazio FE** handles this in an equally smart way. Instead of routing the data to an MQ Output node, you route it to a **Spazio FE “SPOUT”** node. Up to this point, the source MQ messages are read and processed as specified in the original message flow. All the relevant existing business rules and transformation, routing etc are applied. All that has changed is the destination (or one destination) of the data: instead of a WMQ message queue, the destination is a **Spazio FE Repository** file queue.

Extending message flows to read and write files to/from Spazio FE

Combining both SPIN and SPOUT nodes, you can easily adapt an existing message flow to both read from a source file and write to a destination file (as long as both source and destination are queued in the **Spazio FE Repository** system).

Creating new message flows that access Spazio FE

As well as extending existing message flows, **Spazio FE** also of course allows you to program new message flows, simply by including the relevant SPIN, SPOUT and SPYN nodes.

2.2.5 What you require

The conditions for all the above to work are:

- **Spazio FE Repository** Server installed on each broker machine
- **Spazio FE** SPIN, SPOUT and SPYN plug-in nodes installed in each broker
- **Spazio FE** nodes configured to read/write the appropriate **Spazio FE Repository** queues
- **Spazio FE Repository** system configured correctly (queues etc)

Please note that all the above components come bundled within the base Spazio FE system.

- The message flow and the rules in the metadata repository are applicable to the file in question (i.e. the format of the file records is compliant with the format of the WMQ messages handled by the message flow in question).

