WebSphere® Message Broker File Extender

Version 5 Release 0

Installation, Configuration & User Reference Guide

GC34-6666-01

WebSphere. Message Broker File Extender

Version 5 Release 0

Installation, Configuration & User Reference Guide

GC34-6666-01

Note

Before using this information and the product it supports, read the information in “Notices,” in Appendix A.

Second Edition (December 2005)

This edition applies to Version 5 Release 0 Fixpack 1of IBM WebSphere Message Broker File Extender (product number 5724-M01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GC34-6666-00.

Contents

Contents.. iii

About this book................................. 1

Who this book is for 1

What you need to know to understand this book1

What's new for this release 1

What's new for this edition of the manual 2

Structure of this book... 2

Making comments on this book............................ 3

Chapter 1. Product Information 4

Overview 4

Why *WebSphere Message Broker File Extender?* 4

Supported Operating Environment 5

Function Overview.. 7

Ability to read from and write to a file from a message flow 7

Ability to operate on file information with ESQL and built-in IBM WebSphere Business Integration Message Broker nodes 7

Support for File to File message flows 7

Support for File to Message and Message to File message flows 7

Easy installation process and extreme ease of use 8

Architectural Overview 8

Planning considerations 10

Files on the file system 10

Transactionality.. 12

F2F Message Flows and F2M Message Flows design checklist 12

M2F Message Flows design checklist......... 14

Constraints on message flow design and deployment... 15

An example business scenario 16

Business requirements 16

Solution overview.. 16

Chapter 2. Installation..................... 19

Preparing for installation..................................... 19

Checking software prerequisites 19

Checking the operating environment 19

Installing 21

Windows... 21

AIX... 23

Linux.. 25

Solaris 27

Configuring for first use 29

Importing node definitions into the broker toolkit on Windows or Linux 29

Preparation of the runtime environment... 30

Preparation of the broker queue manager. 31

Broker restart 32

Migrating message flows to WebSphere Message Broker V6 32

Installation verification procedure 32

Uninstalling 33

Windows 33

AIX 33

Linux.. 33

Solaris 34

Chapter 3. Configuration 35

FileInput Node Reference.................................... 35

Terminals of a FileInput Node 35

Properties of a FileInput Node................... 36

FileOutput Node Reference................................. 43

Terminals of a FileOutput Node 43

Properties of a FileOutput Node................ 44

FileInput error handling.............................. 49

FileProxy Node Reference 50

Terminals of a FileProxy node.................... 50

Properties of a FileProxy Node 50

Message Tree Structures 53

File Descriptor 53

File Destination List Descriptor.................. 55

File Action Tag 56

Advanced tasks... 56

Altering file nodes behavior by editing mbfe.properties file...................................... 56

Turning on service trace.............................. 59

Handling poison files using mbfesolve tool59

Chapter 4. Using file nodes in a message flow.................................... 61

Usage examples in simple scenarios 61

Preparing to execute the samples 61

Sample files.. 62

Sample 1 – File to File message flow 62

Sample 2 – File to Message message flow 64

Sample 3 – Message to File message flow 66

Chapter 5. Diagnosis....................... 69

Installation problems.. 69

First use problems... 69

Generic runtime problems................................... 69

Contacting IBM Software Support 70
Determine the business impact of your problem 71
Describe your problem and gather background information 71
Submit your problem 72
Chapter 6. Error Messages and error codes 75
Chapter 7. Related information 83
Appendix A. Notices......................... 85
Trademarks.. 87

About this book

Who this book is for

This manual describes how to install, configure and use WebSphere® Message Broker File Extender.

It defines the characteristics and functionality of WebSphere Message Broker File Extender, a series of components that can be used to enable file handling within WebSphere Business Integration Brokers.

This manual is aimed primarily at those whose task it is to install or manage the product.

What you need to know to understand this book

A detailed knowledge of WebSphere Business Integration Message Broker 5 and/or WebSphere Message Broker 6 is required.

What's new for this release

This release is a manufacturing refresh and a fixpack for WebSphere Message Broker File Extender 5.0.0 that delivers full support for WebSphere Message Broker 6.

The difference between the manufacturing refresh bundle and the fixpack bundle is that the former can be used for a new installation of WebSphere Message Broker File Extender, while the latter can be used only to upgrade an existing installation.

Both the manufacturing refresh bundle and the fixpack bundle contain fixes for the following APARs:

IY73729 -- MBFE Output node writes to the default directory even if set
IY76434 -- Error reading and writing file with non default codepages
IY76460 -- MBFE Poor performance when using sync terminal
IY76463 -- MBFE - returns 'JMS_PRIMEUR_SPFFMT_EorEofType' is NULL
IY77482 -- MBFE DestinationList not honored at runtime
IY77594 -- MBFE Properties.RetryCount honored but not refreshed when propagating 'whole file'

On top of introducing full support for WebSphere Message Broker 6, the following minor new features are also introduced:

- Support for silent mode installation

- Support of new options for mbfecfgqmgr and mbfesolve commands
- Support of broker specific and execution group specific mbfe.properties configuration files
- Support for circular tracing
- Backup of installation log file in <mbfeopt_root>/mdv at the end of the installation .

What's new for this edition of the manual

The following additions and modifications have been applied to the GC34-6666-01 edition of this manual

- Description of new installation wizard externals related to WebSphere Message Broker 6 support and silent install.
- Description of new options for mbfecfgqmgr and mbfesolve commands
- Description of new parameters mbfe.properties
- Description of broker specific and execution group specific mbfe.properties configuration files
- Detailed description of FileInput error handling.

Structure of this book

The opening chapter of this book, *Product Information*, describes the purpose and functionality of WebSphere Message Broker File Extender. It introduces the types of WebSphere Business Integration Message Broker message flows that can be constructed or modified by integrating WebSphere Message Broker File Extender nodes, and illustrates this with a sample scenario.

The second chapter , *Installation*, provides information on installation requirements and documents the installation procedure on the various platforms supported.

The next chapter, *Configuration*, describes in detail how to configure WebSphere Message Broker File Extender.

The fourth chapter of this book, *Using MBFE nodes in a message flow*, describes how the message flow developer can integrate WebSphere Message Broker File Extender nodes into a new or modified message flow, and the types of functionality that can be achieved through the use of these nodes.

Two further chapters deal with problem resolution: *Diagnosis* helps the user to make informed decisions on the possible sources of problems, while *Error Messages and error codes* provides a full list of Informational, Warning and Error messages that are written by WebSphere Message Broker File Extender using the standard platform-specific system log.

Related Information lists any further sources of related information, including relevant web sites.

Appendix A: Notices contains relevant information on copyright, licenses and trademarks.

Making comments on this book

If you especially like or dislike anything about this book, feel free to send us your comments.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information that is in this book and to the way in which the information is presented. Speak to your IBM representative if you have suggestions about the product itself.

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

You can get your comments to us quickly by sending an e-mail to **idrcf@hursley.ibm.com**. Alternatively, you can mail your comments to:

User Technologies
IBM United Kingdom Laboratories,
Mail Point 095, Hursley Park,
Winchester, Hampshire, SO21 2JN, United Kingdom

Please ensure that you include the book title, order number, and edition date.

Chapter 1. Product Information

Overview

WebSphere Message Broker File Extender provides file handling capability for IBM WebSphere Business Integration Brokers, enhancing WebSphere Business Integration beyond pure messaging into a more complete brokering solution.

WebSphere Message Broker File Extender is made up of plug-in nodes that are tightly coupled with the broker runtime and extend the palette of nodes available in the broker Eclipse-based tooling. This approach is unique to WebSphere Message Broker File Extender because most alternative solutions are based on more complex adapter frameworks.

By using WebSphere Message Broker File Extender , the message broker can handle incoming files directly via a file input node, without having to convert the data into WebSphere MQ™ messages, and place them on a WebSphere MQ input queue to drive the message flow.

A file output node is also provided to write message flow data directly to files, rather than to a WebSphere MQ output queue.

The file routing, content enrichment and transformation can all be performed within an existing WebSphere Business Integration Message Broker installation.

Why *WebSphere Message Broker File Extender?*

WebSphere Message Broker File Extender and WebSphere Business Integration Message Broker are an ideal platform for business integration. They offer you the essential flexibility to start small and grow with your customer demands.

Ease of implementation and ease of use in production ensures that time to value can be slashed and return on investment maximized.

These integration offerings help you increase control over your business, allowing rapid response to events as they occur, and enabling timely delivery of business information that supports key business decisions.

The use of WebSphere Message Broker File Extender means that useful and timely information extracted from files can be provided through the WebSphere Business Integration Message Broker for customers, suppliers, and partners in order to improve customer satisfaction and competitiveness.

It can help reduce delivery cycle times and error rates, improving operational performance, including reducing inventory and improving cash flow to increase return on capital employed.

Chief executives can consider improving customer service and business agility while reducing operational and IT costs by:

- Making their organization work more smartly and more productively
- Giving them better visibility and control over file related operations
- Giving their people the information they need to be successful, including file information.

Corporate information officers can consider reducing system development and maintenance costs by:

- Reusing existing file based applications and file data, rather than purchasing, rebuilding or setting up complex adapter frameworks.
- Extend the benefits of a managed mediation framework, such as that provided by IBM WebSphere Business Integration Message Broker, to file based applications and file data.
- Improving system flexibility and adaptability.

Operations Executives can consider reducing operational costs by:

- Reducing inventory and improving cash flow
- Improving workforce productivity through automation
- Reducing the cost of rectifying process errors

Customer-facing executives can consider improving customer service by:

- Supplying useful and timely information to customers, suppliers, and partners
- Reducing delivery cycle times
- Reducing error rates.

WebSphere Message Broker File Extender extends the value of previous investments by supporting and transforming information between legacy file based applications and new applications based on different protocols, standards and formats.

Supported Operating Environment

WebSphere Business Integration Message Broker 5 broker runtime component is supported on:

- IBM AIX® Version 5.1 (maintenance level 3) or IBM AIX Version 5.2 (maintenance level 2) or later 5.x on IBM pSeries™, RS/6000® processor machines or equivalent
- Microsoft® Windows® 2000 Server (with SP3) or Microsoft Windows 2000 Advanced Server (with SP3) or Microsoft Windows Server 2003 (Standard and Enterprise editions) on IBM xSeries™ and Intel® based systems (or equivalent), various manufacturers
- Sun Solaris 8 or Sun Solaris 9 (both with the SunSolve recommended patch level) on Sun Microsystems SPARC processor machines

- Red Hat Enterprise Linux™ AS 2.1 or Red Hat Enterprise Linux AS 3.0 or SuSE Linux Enterprise Server (SLES) 8 or SuSE Linux Enterprise Server (SLES) 9 on IBM xSeries and Intel based systems (or equivalent), various manufacturers.

WebSphere Business Integration Message Broker 5 Toolkit design time component is supported on IBM xSeries and Intel based systems (or equivalent), from various manufacturers, running one of the following operating systems:

- Microsoft Windows 2000 Professional (with SP3)
- Microsoft Windows 2000 Server (with SP3)
- Microsoft Windows 2000 Advanced Server (with SP3)
- Microsoft Windows XP Professional (with SP1) – for development and test purposes only

WebSphere Business Integration Message Broker 6 broker runtime component is supported on:

- IBM AIX V5.2 (maintenance level 3) or IBM AIX V5.3 (maintenance level 2) on IBM pSeries, RS/6000 processor machines or equivalent
- Microsoft Windows Server 2003 (Standard and Enterprise editions) , Microsoft Windows XP (with SP1) (for development and test purposes only)
- Sun Solaris 8 (with the SunSolve recommended patch level and Patch ID 111308-05) or Sun Solaris 9 (with the SunSolve recommended patch level) on Sun Microsystems SPARC processor machines
- Linux Red Hat Enterprise Advanced Server V3.0 plus Update 2 (Kernel Version 2.4.21) or SUSE Linux Enterprise Server (SLES) 9 (Kernel Version 2.6) on IBM xSeries and Intel based systems (or equivalent), various manufacturers.

WebSphere Message Broker 6 Toolkit design time component is supported on IBM xSeries and Intel based systems (or equivalent), from various manufacturers, running one of the following operating systems:

- Microsoft Windows Server 2003 Standard Edition
- Microsoft Windows Server 2003 Enterprise Edition
- Microsoft Windows XP Professional (with SP1) – for development and test purposes only
- Linux Red Hat Enterprise Advanced Server V3.0 plus Update 2 (Kernel Version 2.4.21)
- SUSE Linux Enterprise Server (SLES) 9 (Kernel Version 2.6)

Notes

Where patches for a product are specified, this means that these patches were required to resolve problems found during testing, and should, therefore, be considered as the minimum necessary maintenance level. Further maintenance should be applied following the recommendations of the relevant software manufacturer.

Up-to-date information about system requirements for WebSphere Message Broker File Extender and prerequisite broker products is available at the following website:

http://www.ibm.com/software/integration/wbimessagebroker/requirements/

Function Overview

Below is a short overview of the functions and capabilities of WebSphere Message Broker File Extender:

Ability to read from and write to a file from a message flow

WebSphere Message Broker File Extender FileInput and FileOutput nodes allow you to write a message flow that takes input from files stored in a local file system source directory (with the possible use of wildcards to select specific file subsets) and to produce output in the form of files in a local file system target directory.

When reading from a file you have the option of propagating the file content to the message flow either record by record or as a single buffer of information; in both cases the information read from files can be parsed exploiting the native WebSphere Business Integration Message Broker MRM definitions.

Ability to operate on file information with ESQL and built-in IBM WebSphere Business Integration Message Broker nodes

The information read from a file or written to a file can be handled from within a message flow using the standard facilities offered by IBM WebSphere Business Integration Message Broker, such as Compute nodes with ESQL language, Mapping nodes, Filter nodes and MRM parsing.

Support for File to File message flows

When a FileInput node is used in conjunction with a FileOutput node, the resulting message flow can be used to perform pure file brokering functions such as: file format conversion, file content augmentation and filtering.

Support for File to Message and Message to File message flows

When an FileInput node is used in conjunction with an MQOutput node, or vice versa, when an MQInput node is used in conjunction with an FileOutput node, the resulting message flow can be used to implement two-way interfaces between message based applications and file based applications, possibly including value added functions such as on the fly format conversion (for example, from a COBOL copybook-like format to XML) and routing.

Easy installation process and extreme ease of use

WebSphere Message Broker File Extender features an easy installation process – a key attribute of IBM's increased focus on supreme ease of use. Other usability features include: defaults for most configuration parameters and several working sample message flows.

Architectural Overview

WebSphere Message Broker File Extender architecture is very simple (see also Figure 1), the product consists of two macro components:

- a *run time component* (the effective *plug-in nodes*) that must be installed on all brokers where file handling capabilities are required
- a *design time component* that must be installed on all workstations featuring the Broker Toolkit GUI that will be used to design message flows including file handling.

No WebSphere Message Broker File Extender components are installed on WebSphere Business Integration Message Broker Configuration Manager.

Let’s now have a closer look at the broker components. WebSphere Message Broker File Extender comes with three nodes:

- *FileInput*: this node reads business information from a file and passes its contents to a message flow.
- *FileOutput*: this node receives business information from a message flow and stores it into a file.
- *FileProxy*: this is a helper node that must be used in conjunction with FileOutput in all message flows rooted in an MQInput node, for example, in message to file message flows.

WebSphere Message Broker File Extender users will exploit file handling capabilities from the broker by modeling message flows using one or more file nodes in conjunction with broker built-in nodes and deploying them to one or more target brokers.

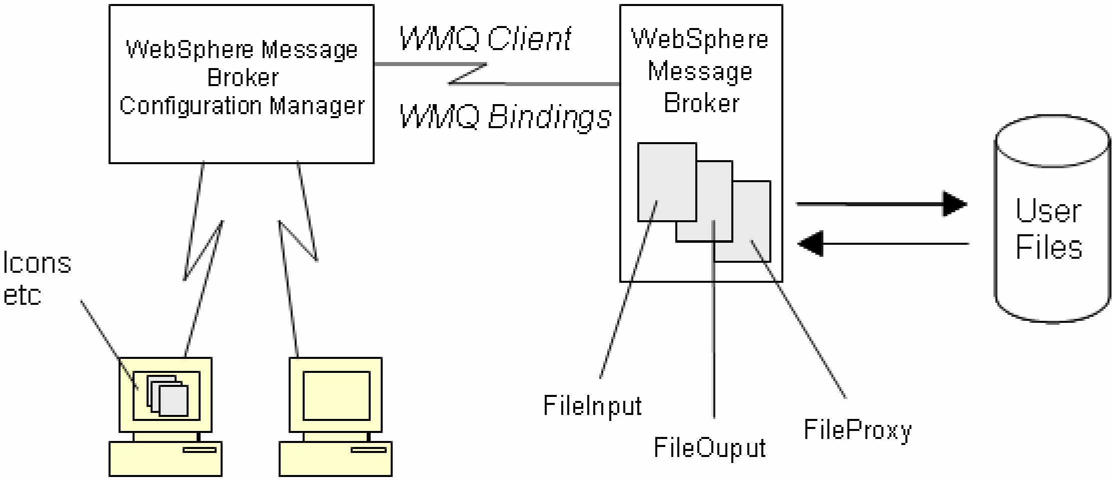


Figure 1 – WebSphere Message Broker File Extender components

WebSphere Message Broker File Extender enables message flow programmers to write new types of message flows that have a file source as input and/or a file destination as output, with several file handling options to choose from.
The extended set of message flows that can be written using file nodes can be divided into three categories:

- *File to file* (*F2F*) *message flows*: these are pure file transformation and or routing message flows, no WebSphere MQ messages are actually involved in the computation. Both FileInput and FileOutput nodes are needed in this scenario.
- *File to message* (*F2M*) *message flows*: these are message flows than can be used to split file contents into many records that are passed to the message flow for transformation and routing; the resulting information will typically be put on an output WebSphere MQ queue using the built-in MQOutput node (F2M message flows can also include FileOutput nodes if needed).
  Only a FileInput node is typically needed in this scenario.
- *Message to file* (*M2F*) *message flows*: these are message flows that aggregate the contents of WebSphere MQ messages over files; messages are read from a queue using an IBM provided MQInput node.
  Both FileProxy and FileOutput nodes are needed in this scenario.

In the architecture of a message flow, the role of the input node is pivotal, in that it is the input node's responsibility to drive commit and implement appropriate error management/recovery logic.

For this reason WebSphere Message Broker File Extender requires that when implementing a M2F message flow (that is, a message flow rooted in an MQInput node) you wire a FileProxy node immediately downstream of each MQInput node. This node will act as a transparent proxy for all FileOutput nodes contained in the flow, preserving any existing custom error management logic.

Planning considerations

When planning the adoption of WebSphere Message Broker File Extender there are several aspects that must be considered during the planning phase. This section outlines the main ones, providing annotated checklists when applicable.

Files on the file system

Availability of input files on a file system

WebSphere Message Broker File Extender users must make input files available to FileInput nodes by placing them in directories of the **local** file system where the user running the broker processes has adequate permissions to access them.

FileInput does not process files directly from the directory where they were stored, but rather moves them to an internal (*staging*) directory and processes them from there.

As a consequence of this approach, files are removed from where the user originally placed them; in this respect, input directories monitored by WebSphere Message Broker File Extender can be considered as input *file queues* where file workload is queued up by users and is dequeued by FileInput node instances.

Files are **moved** by WebSphere Message Broker File Extender from the input directory to the staging directory very quickly, so if the quantity of input file workload is high it can happen that an input file **disappears** immediately from the input directory but is processed by the broker only at a later time (for example, when a FileInput instance becomes available).

The above behavior is intended to alleviate a problem that is typical of some file based scenarios, where a file with a fixed name is created in a fixed directory with a certain frequency (for example, hourly or daily); the problem with this approach is that if the file is not removed quickly enough from its original position, it may be overwritten.

FileInput has a built-in mechanism that avoids processing of open files (for example, files that are not yet fully formed because the producing application is still writing to them).

While this feature protects against the processing of incomplete files in most scenarios, there is no way of avoiding the processing of a bad file in extreme cases, for instance where a corrupt file is created by an application that crashed, leaving the file closed but incomplete. The handling of such cases is delegated to custom error checking and exception recovery logic that the user should have embedded in the message flow.

A message flow rooted in a FileInput node (either F2F type or F2M type) can be deployed in multiple instances to a target broker. This enables parallel processing of the same set of files, for example, the contents of a directory, by several message flow instances.

Similarly, distinct message flows (with different underpinning business logic) may read from the same input directory, provided that the file filters in the respective FileInput nodes are configured in such a way that each file matches at most one of the filter clauses specified.

This latter aspect is a very important one to consider in the planning phase because it may require a revision of existing file placement policies and file naming conventions.

Production of output files on the file system

FileOutput node instances wired in an F2F type or M2F type message flow produce output files in the specified directory according to a configurable naming scheme.

FileOutput has a built-in mechanism that retains all output files not yet fully formed in an internal staging directory. As a consequence of this approach, external applications will see output files only when they are completely closed. Depending on the size of the files this may imply a certain latency.

Unfortunately, some file-based applications require that their input files are created with a fixed name in a fixed directory; the problem with this approach is that if the file is not consumed quickly enough by the application it may be overwritten.

In order to alleviate this problem, the FileOutput node features flexible file *disposition* options that let you choose whether to overwrite the existing file, append to it or let FileOutput generate a unique name for it (whenever possible, this is the recommended option).

Where it is feasible to configure FileOutput to generate unique file names, output directories handled by FileOutput node instances can be thought of as output *file queues* where file workload is queued up by FileOutput and dequeued by user file based applications.

Automatic clean-up of unprocessed files

WebSphere Message Broker File Extender includes an automatic clean up of unprocessed files that have been stored in the product internal directories for longer than **60 days**.

A file can be left unprocessed in WebSphere Message Broker File Extender internal directories for several reasons, the most common of which is that a message flow that caused the files to be moved to the internal directories has subsequently been undeployed from the broker execution group.

To learn how to modify this expiration time or completely disable the automatic clean-up feature please refer to "Advanced tasks" section.

Transactionality

The implementation of transactional integrity in WebSphere Message Broker File Extender is based on the use of control queues in WebSphere MQ.

Given that file handling has some unique requirements (mostly stemming from potentially huge file size), WebSphere Message Broker File Extender gives the user a flexible set of mechanisms, in order to implement the transaction demarcation policy best suited for the target environment; for example, in a F2M flow you can choose to release (commit) file records as WebSphere MQ messages in batches of a predetermined size (for example, 50), or to force the whole file content to correspond to a single big WebSphere MQ LUW (of course this can cause problems with your queue manager max uncommitted messages MAXUMSGS settings).

If all the safeguards discussed above fail, and one of the nodes goes into an "in doubt" state (files not being transactional in the strict sense of the term), you can configure WebSphere Message Broker File Extender to automatically undo/redo in-doubt work, or simply propagate exception information to the node failure terminal for an administrator to establish what the root cause of the problem was (all this is very similar to what happens on the rare occasions when a WebSphere MQ MCA channel goes in doubt and you have to work with the RESOLVE CHANNEL command).

F2F Message Flows and F2M Message Flows design checklist

Identification of input files to be processed

As discussed in the previous sections, the first step in designing a new F2F or F2M is that of uniquely identifying a set of input files on the file systems that the message flow is to *consume* (that is, to process and subsequently *delete*).

Selection of file content propagation method

Once a valid set of input files has been identified, the message flow programmer must decide how to propagate the file content to the message flow, this decision will be based upon factors such as: file size, reusability of existing MRM definitions, kind of business logic that the message flow is to apply to the file, and so on.

There are three options that can be used:

- *Propagation of the whole file content in a single iteration*: this option can only be used when the file size falls below the limits of the hardware and the broker itself to manage the whole file content in memory (for example, a 10K file is not a problem, while a 10 MB may cause some problems on heavily used low end servers);

- *Propagation of the file content record by record*: this option is suitable for big files and when each CR / CR-LF terminated or fixed length record in the file makes some business sense on its own (for example, an item sold in a supermarket or a money transfer);
- *Propagation of just the file descriptor but not the file body*: this third option is intended to be used when the broker does not have to change the file contents but only to perform auditing/tracking tasks or move files from one directory to another based on complex rules (for example, all files with a certain name are subject to special handling on bank holidays).

Revision of parsing requirements

Once a suitable unit of information is read from the file (for example, a single record or the whole file), you must consider if all the required metadata is explicitly (MRM definition) or implicitly (generic self-describing XML, well known MQ headers, etc) available to the message flow, in order to correctly handle the parsing of the information retrieved from the file.

Revision of transactionality requirements

For F2F and F2M transactional message flows the main transaction-related decisions to be taken are:

- *Transaction demarcation*: you have the ability to choose whether to have a unit-of-work (UOW) that spans the whole file or commit resources after a certain batch of records is processed; for big files it is typically better to commit each batch in order to release locks on resources (for example, the in-flight MQ UOW)
- *Coordination with other broker resources*: if the message flow alters some other non-file resource such as WebSphere MQ (always the case for F2M) or a database, and the user requires transacted coordination with this resource, an extra terminal, *Synch,* must be connected on the FileInput node; please refer to the FileInput reference section for details.

Preparation for output on WebSphere MQ (F2M only)

In F2M message flow the message flow design must explicitly include a Compute node step that handles the construction of a suitable WebSphere MQ Message Descriptor before performing the output on a queue via MQOutput node.

For extra details and ESQL code snippets please refer to the section discussing constraints on message flow programmers.

M2F Message Flows design checklist

FileProxy Node wiring

As previously discussed, each M2F message flow must include a FileProxy node wired right after the MQInput node, featuring a set of connected input/output terminals exactly as the set of terminals that the MQInput node would have needed for error/exception management if the FileProxy node were not there. FileProxy is just a helper node that must be included in *all* M2F flows.

Determination of output file attributes

The main attributes of output files that must be configured in the node properties or coded in ESQL are:

- *File location*: directory where the files must be written
- *File name*: name of the file to be created (can be automatically altered to make it unique depending on file disposition settings)
- *File disposition*: option that controls whether to overwrite/append existing files or create uniquely named files in case of file name collisions.

Selection of file closure policy

In a M2F scenario, where input WebSphere MQ messages are being accumulated in a file, there is no immediate way of determining when it is appropriate to close the output file under construction and release it into the external user file system.

WebSphere Message Broker File Extender provides three different approaches that can be used singularly or combined together in order to control output file closure time:

- *Time threshold*: when no new input WebSphere MQ message is received for a configurable number of seconds, the output file is closed.
- *Record threshold*: when a configurable number of records have been written to the output file, the file is closed.
- *Explicit*: when the message flow program determines (for example, by inspecting input messages content) that the file must be closed, it creates a special WebSphere Message Broker File Extender honored element in *LocalEnvironment* that in turn causes the file to be closed.

Revision of transactionality requirements

When creating a transactional M2F flow, an extra terminal, *Synch,* must be connected to the FileProxy node; please refer to the FileProxy reference section for details.

Constraints on message flow design and deployment

LocalEnvironment tree propagation

WebSphere Message Broker File Extender stores several attributes in the *LocalEnvironment* tree (see "Message Tree Structures" section for reference information); in order to ensure that these attributes do not get lost during the traversal of a message flow graph, you must set the *Advanced.Compute Mode* attribute of Compute nodes to **LocalEnvironment and Message**.

Creation of a WebSphere MQ message descriptor for F2M message flows

In F2M message flows you need to create a valid ReplyIdentifier (required by WebSphere Business Integration Message Broker even if no reply needs to be created) and a WebSphere MQ message descriptor before propagating to an MQOutput node, as no such descriptor will be provided by FileInput.

Below is a snippet of code you can use to create a ReplyIdentifier and minimal MQMD.

```
CALL CopyMessageHeaders();
SET OutputRoot.Properties.ReplyIdentifier = MQCI_NONE;
SET OutputRoot.MQMD.Version=MQMD_CURRENT_VERSION;
SET OutputRoot.MQMD.StrucID=MQMD_STRUC_ID;
```

Typically you also need to provide the ESQL code for content propagation (file content in this specific case) right after MQMD creation; the resulting ESQL snippet is shown below.

```
SET OutputRoot.Properties.ReplyIdentifier = MQCI_NONE;
SET OutputRoot.MQMD.Version=MQMD_CURRENT_VERSION;
SET OutputRoot.MQMD.StrucID=MQMD_STRUC_ID;

SET OutputRoot.BLOB = InputBody;

SET OutputLocalEnvironment = InputLocalEnvironment;
```

Revision of MQOutput node settings for F2M message flows

In F2M message flows you need to set the attribute *Advanced.Message Context* to **default**.

Typically this attribute will be set to **Pass All**, but this value is not valid when the message being written by MQOutput node was not originally read from a queue via an MQInput node.

Failing to correctly set *Advanced.Message Context* will cause the MQOutput node to fail with this WebSphere MQ reason code: MQRC_CONTEXT_HANDLE_ERROR (MQRC=2097).

Revision of parallelism constraints for M2F message flows

Only the following settings are currently supported for assigned M2F message flows properties:

- *Commit Count* = **1**
- *Additional Instances* = **0**

More generally, the product does not currently support more than one M2F message flow instance serving the same WebSphere MQ queue either from the same execution group or a different one.

To overcome some of the scalability limitations stemming from this constraint, you can deploy multiple instances of the same identical M2F message flow with each one reading messages from a different WebSphere MQ queue (as a best practice you can deploy as many instances as the physical processors available on the broker machine) and then deploy an extra service flow reading messages from the original local queue and distributing them in a balanced way over the pool of queues served by the M2F message flows.

An example business scenario

Business requirements

A nightly scheduled proprietary application produces a CSV file representing the last business day accounts receivable.

For a subset of the files, the contents must be archived and the main file attributes must be persisted in a relational audit trail to help with Sarbanes-Oxley (SOX) compliance before being fed into the mySAP.com ERP system.

Solution overview

The existing proprietary application is run unchanged, while the rest of the processing is driven by two message flows.

The first message flow (“SOX message flow”) performs the following steps:

- A FileInput node reads the file descriptor
- A Filter node selectively (based on file name) drives:
  - A FileOutput node that creates a copy of the file on a directory “guarded” by Tivoli Storage Manager in order to feed the corporate file warehouse.
  - A Database node that inserts a new entry in a database table, storing an audit trail of the file attributes of all the processed files.
- The file is then copied in the input directory of the second message flow

The second message flow (“SAP message flow) performs the following steps:

- A FileInput node reads the file content record by record

- Each record that is parsed using an MRM definition is processed by a mapping node in order to create a business object structure
- The business object is serialized as an XML message and sent via an MQOutput node to WebSphere Business Integration Adapter for mySap.com, which will take care of the actual interaction with SAP.

This scenario shows how a very simple broker-centric architecture can help in coping with stringent business requirements in complex and heterogeneous IT environments.

Chapter 2. Installation

Preparing for installation

Checking software prerequisites

WebSphere Message Broker File Extender requires one of these broker products:

- WebSphere Business Integration Message Broker 5.0 with Fix Pack 05
- WebSphere Message Broker 6.0 with fix for APAR IY77292.

In addition, Java classes for MQ feature for the prerequisite WebSphere MQ must also be installed.

Up-to-date information on system requirements for WebSphere Message Broker File Extender and prerequisite broker products is available at the following website:

http://www.ibm.com/software/integration/wbimessagebroker/requirements/

Checking the operating environment

The WebSphere Business Integration Message Broker 5 broker runtime component is supported on:

- IBM AIX Version 5.1 (maintenance level 3) or IBM AIX Version 5.2 (maintenance level 2) or later 5.x on IBM pSeries, RS/6000 processor machines or equivalent
- Microsoft® Windows® 2000 Server (with SP3) or Microsoft Windows 2000 Advanced Server (with SP3) or Microsoft Windows Server 2003 (Standard and Enterprise editions) on IBM xSeries and Intel® based systems (or equivalent), various manufacturers
- Sun Solaris 8 or Sun Solaris 9 (both with the SunSolve recommended patch level) on Sun Microsystems SPARC processor machines
- Red Hat Enterprise Linux AS 2.1 or Red Hat Enterprise Linux AS 3.0 or SuSE Linux Enterprise Server (SLES) 8 or SuSE Linux Enterprise Server (SLES) 9 on IBM xSeries and Intel based systems (or equivalent), various manufacturers.

The WebSphere Business Integration Message Broker 5 Toolkit design time component is supported on IBM xSeries and Intel based systems (or equivalent), from various manufacturers, running one of the following operating systems:

- Microsoft Windows 2000 Professional (with SP3)
- Microsoft Windows 2000 Server (with SP3)
- Microsoft Windows 2000 Advanced Server (with SP3)
- Microsoft Windows XP Professional (with SP1) – for development and test purposes only

The WebSphere Business Integration Message Broker 6 broker runtime component is supported on:

- IBM AIX V5.2 (maintenance level 3) or IBM AIX V5.3 (maintenance level 2) on IBM pSeries, RS/6000 processor machines or equivalent
- Microsoft Windows Server 2003 (Standard and Enterprise editions) , Microsoft Windows XP (with SP1) (for development and test purposes only)
- Sun Solaris 8 (with the SunSolve recommended patch level and Patch ID 111308-05) or Sun Solaris 9 (with the SunSolve recommended patch level) on Sun Microsystems SPARC processor machines
- Linux Red Hat Enterprise Advanced Server V3.0 plus Update 2 (Kernel Version 2.4.21) or SUSE Linux Enterprise Server (SLES) 9 (Kernel Version 2.6) on IBM xSeries and Intel based systems (or equivalent), various manufacturers.

The WebSphere Message Broker 6 Toolkit design time component is supported on IBM xSeries and Intel based systems (or equivalent), from various manufacturers, running one of the following operating systems:

- Microsoft Windows Server 2003 Standard Edition
- Microsoft Windows Server 2003 Enterprise Edition
- Microsoft Windows XP Professional (with SP1) – for development and test purposes only
- Linux Red Hat Enterprise Advanced Server V3.0 plus Update 2 (Kernel Version 2.4.21)
- SUSE Linux Enterprise Server (SLES) 9 (Kernel Version 2.6)

Notes

Where patches for a product are specified, this means that these patches were required to resolve problems found during testing, and should, therefore, be considered as the minimum necessary maintenance level. Further maintenance should be applied following the recommendations of the relevant software manufacturer.

Up-to-date information about system requirements for WebSphere Message Broker File Extender and prerequisite broker products is available at the following website:

http://www.ibm.com/software/integration/wbimessagebroker/requirements/

Important Note for Solaris Users

WebSphere Business Integration Message Broker 5.0 for Solaris requires Sun Java™ Runtime Environment (*JRE*) 1.3.1 (SR3 or later).

When using WebSphere Message Broker File Extender you must upgrade the Sun JRE used by the broker to version IBM JRE **1.3.10**.

Important Note for Linux Users

WebSphere Business Integration Message Broker 5.0 for Linux requires IBM JRE 1.4.x on some Linux distributions (for example, Redhat EL AL 3.0).

In these cases WebSphere Message broker File Extender will work correctly only if this JRE related environment variable definition is available to the broker at runtime:

```
IBM_JVM_SR_SIGNUM = 64
```

This setting prevents broker hangs that could be caused by operating system specific and JRE specific issues in the area of signal handling.

For more detailed information about supported JREs on Linux and Solaris platforms for WebSphere Business Integration Message Broker 5.0 see also this technote:
http://www-1.ibm.com/support/docview.wss?rs=849&uid=swg27006541

Installing

Windows

GUI Wizard Driven Installation

1. Make sure you are logged in as an administrator
2. Run `setupwin32.exe`
3. Wait while a JRE suitable to continue to run the installation wizard is found.
   If you get an error message stating that no JRE has been found this means that no prerequisite broker product is installed on the machine (as it includes a JRE).
4. Read the welcome screen
5. Read and accept the license agreement
6. Select the installation directory for constant data (for example, executable programs and documentation) which will be referred to as <*mbfeopt_root*> in the rest of this manual – this step is skipped in the case of fixpack installation

7. Select the installation directory for variable data (for example, configuration files) which will be referred to as <*mbfevar_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
8. Complete the wizard and wait for the installation process to complete.

Console Mode Wizard Driven Installation

1. Make sure you are logged in as an administrator
2. Open a command prompt window or telnet to a remote machine where an installable product image is available
3. Run `setupwin32console.exe -console`
4. Wait while a JRE suitable to continue to run the installation wizard is found.
   If you get an error message stating that no JRE has been found this means that no prerequisite broker product is installed on the machine (as it includes a JRE).
5. Read the welcome screen
6. Read and accept the license agreement
7. Select the installation directory for constant data (for example, executable programs and documentation) which will be referred to as <*mbfeopt_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
8. Select the installation directory for variable data (for example, configuration files) which will be referred to as <*mbfevar_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
9. Complete the wizard and wait for the installation process to complete.

Silent installation

1. Make sure you are logged in as an administrator
2. Review the contents of the response file template setupwin32.opt and modify it to reflect your requirements (if you are installing from a CD you need to copy that file to a non read-only file system location before editing it).
3. Run `setupwin32Console -options setupwin32.opt –silent`

Solving installation problems

In case of installation problems please review the installation log file `mbfeinstlog.txt` that is created in the system temporary directory (if you are not sure of where that is, open a command prompt window and inspect the output of this command: `echo %TEMP%`). At the end of the installation this file is copied to the <mbfeopt_root>\mdv directory for future reference.
If the problem persists, or if you suspect that the problem lies with the installation program itself, you can try to rerun the installation program with the additional command line parameter `–is:log c:\ishield.log` and then inspect the `ishield.log` file produced.

If the prerequisite WebSphere Business Integration Message Broker 5.0 with Fix Pack 05 product was not found during the setup and you decided to carry on with the installation, and you install the broker at a later time, you must remember to copy these two files under `<wbimb5_root>/jplugin` directory (where `<wbimb5_root>` is the WebSphere Business Integration Message Broker 5.0 installation directory):

```
<mbfeopt_root>\bin\mbfe.jar
<mbfeopt_root>\bin\mbfe.path.jar
```

If the prerequisite WebSphere Message Broker 6 product was not found during the setup and you decided to carry on with the installation, and you install the broker at a later time, you must remember to copy this file under the `%MQSI_WORKPATH%\common\profiles` directory:

`<mbfeopt_root>\bin\mbfeprofile.cmd`.

If the optional prerequisite WebSphere Business Integration Message Broker 5 Toolkit component was not found during the setup and you decided to carry on with the installation, and you install the toolkit at a later time, you must remember to copy this file under `<wbimb5t_root>/eclipse/links` (where `<wbimb5_root>` is the WebSphere Business Integration Message Broker 5.0 installation directory):

```
<mbfeopt_root>\eclipse\wbimb5\eclipse\com.ibm.mbfe.link
```

If the optional prerequisite WebSphere Business Integration Message Broker 6 Toolkit component was not found during the setup and you decided to carry on with the installation, and you install the toolkit at a later time, you must remember to copy this file under `<wmb6t_root>/eclipse/links` (where `<wmb6_root>` is the WebSphere Message Broker 6 Toolkit installation directory):

```
<mbfeopt_root>\eclipse\wmb6\eclipse\com.ibm.mbfe.link
```

AIX

GUI Wizard Driven Installation

1. Ensure you are logged in as root
2. Run `setupaix`
3. Wait while a JRE suitable to continue to run the installation wizard is found.
   If you get an error message stating that no JRE has been found, this means that no prerequisite broker product is installed on the machine (as it includes a JRE).
4. Read the welcome screen
5. Read and accept the license agreement

6. Select the installation directory for constant data (for example, executable programs and documentation) which will be referred to as <*mbfeopt_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
7. Select the installation directory for variable data (for example, configuration files) which will be referred to as <*mbfevar_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
8. Complete the wizard and wait for the installation process to complete.

Console Mode Wizard Driven Installation

1. Ensure you are logged in as root
2. Open a shell window or telnet to a remote machine where an installable product image is available
3. Run `setupaix -console`
4. Wait while a JRE suitable to continue to run the installation wizard is found.
   If you get an error message stating that no JRE has been found, this means that no prerequisite broker product is installed on the machine (as it includes a JRE).
5. Read the welcome screen
6. Read and accept the license agreement
7. Select the installation directory for constant data (for example, executable programs and documentation) which will be referred to as <*mbfeopt_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
8. Select the installation directory for variable data (for example, configuration files) which will be referred to as <*mbfevar_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
9. Complete the wizard and wait for the installation process to complete.

Silent installation

1. Make sure you are logged in as an administrator
2. Review the contents of the response file template `setupaix.opt` and modify it to reflect your requirements (if you are installing from a CD you need to copy that file to a non read-only file system location before editing it).
3. Run `setupaix -options setupaix.opt –silent`

Solving installation problems

In case of installation problems please review the installation log file `/tmp/mbfeinstlog.txt`. At the end of the installation this file is copied to the <mbfeopt_root>/mdv directory for future reference.

If the problem persists, or if you suspect that the problem lies with the installation program itself, you can try to rerun the installation program with the additional command line parameter `–is:log /tmp/ishield.log` and then inspect the `ishield.log` file produced.

If the prerequisite WebSphere Business Integration Message Broker 5.0 with Fix Pack 05 product was not found during the setup and you decided to carry on with the installation, and you install the broker at a later time, you must remember to link these two files from the broker `/usr/opt/mqsi/jplugin` directory:

```
<mbfeopt_root>/bin/mbfe.jar
<mbfeopt_root>/bin/mbfe.path.jar
```

If the prerequisite WebSphere Message Broker 6 product was not found during the setup and you decided to carry on with the installation, and you install the broker at a later time, you must remember to link this file from under the `$MQSI_WORKPATH/common/profiles` directory:

`<mbfeopt_root>/bin/mbfeprofile.sh`.

Linux

GUI Wizard Driven Installation

1. Make sure you are logged in as root
2. Run `setuplinux`
3. Wait while a JRE suitable to continue to run the installation wizard is found.
   If you get an error message stating that no JRE has been found, this means that no prerequisite broker product is installed on the machine (as it includes a JRE).
4. Read the welcome screen
5. Read and accept the license agreement
6. Select the installation directory for constant data (for example, executable programs and documentation) which will be referred to as <*mbfeopt_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
7. Select the installation directory for variable data (for example, configuration files) which will be referred to as <*mbfevar_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
8. Complete the wizard and wait for the installation process to complete.

Console Mode Wizard Driven Installation

1. Ensure you are logged in as root
2. Open a shell window or telnet to a remote machine where an installable product image is available
3. Run `setuplinux -console`
4. Wait while a JRE suitable to continue to run the installation wizard is found.

If you get an error message stating that no JRE has been found, this means that no prerequisite broker product is installed on the machine (as it includes a JRE).

5. Read the welcome screen
6. Read and accept the license agreement
7. Select the installation directory for constant data (for example, executable programs and documentation) which will be referred to as <*mbfeopt_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
8. Select the installation directory for variable data (for example, configuration files) which will be referred to as <*mbfevar_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
9. Complete the wizard and wait for the installation process to complete.

Silent installation

1. Make sure you are logged in as an administrator
2. Review the contents of the response file template `setuplinux.opt` and modify it to reflect your requirements (if you are installing from a CD you need to copy that file to a non read-only file system location before editing it).
3. Run `setuplinux -options setuplinux.opt –silent`

Solving installation problems

In the case of installation problems please review the installation log file `/tmp/mbfeinstlog.txt`. At the end of the installation this file is copied to the <mbfeopt_root>/mdv directory for future reference.

If the problem persists, or if you suspect that the problem lies with the installation program itself, you can try to rerun the installation program with the additional command line parameter `–is:log /tmp/ishield.log` and then inspect the `ishield.log` file produced.

If the prerequisite WebSphere Business Integration Message Broker 5.0 with Fix Pack 05 product was not found during the setup and you decided to proceed with the installation, and you install the broker at a later time, you must remember to link these two files from the broker `/opt/mqsi/jplugin` directory:

```
<mebfeopt_root>/bin/mbfe.jar
<mebfeopt_root>/bin/mbfe.path.jar
```

If the prerequisite WebSphere Message Broker 6 product was not found during the setup and you decided to carry on with the installation, and you install the broker at a later time, you must remember to link this file from under the `$MQSI_WORKPATH/common/profiles` directory:

`<mbfeopt_root>/bin/mbfeprofile.sh`.

If the optional prerequisite WebSphere Business Integration Message Broker 6 Toolkit component was not found during the setup and you decided to carry on with the installation, and you install the toolkit at a later time, you must remember to link this file from under `<wmb6t_root>/eclipse/links` (where `<wmb6t_root>` is the WebSphere Message Broker 6 Toolkit installation directory):

```
<mbfeopt_root>/eclipse/wmb6/eclipse/com.ibm.mbfe.link
```

Solaris

GUI Wizard Driven Installation

1. Make sure you are logged in as root
2. Run `setupsolaris`
3. Wait while a JRE suitable to continue to run the installation wizard is found.
   If you get an error message stating that no JRE has been found, this means that no prerequisite broker product is installed on the machine (as it includes a JRE).
4. Read the welcome screen
5. Read and accept the license agreement
6. Select the installation directory for constant data (for example, executable programs and documentation) which will be referred to as <*mbfeopt_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
7. Select the installation directory for variable data (for example, configuration files) which will be referred to as <*mbfevar_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
8. Complete the wizard and wait for the installation process to complete.

Console Mode Wizard Driven Installation

1. Ensure you are logged in as root
2. Open a shell window or telnet to a remote machine where an installable product image is available
3. Run `setupsolaris -console`
4. Wait while a JRE suitable to continue to run the installation wizard is found.
   If you get an error message stating that no JRE has been found, this means that no prerequisite broker product is installed on the machine (as it includes a JRE).
5. Read the welcome screen
6. Read and accept the license agreement
7. Select the installation directory for constant data (for example, executable programs and documentation) which will be referred to as <*mbfeopt_root*> in the rest of this manual – this step is skipped in the case of fixpack installation

8. Select the installation directory for variable data (for example, configuration files) which will be referred to as <*mbfevar_root*> in the rest of this manual – this step is skipped in the case of fixpack installation
9. Complete the wizard and wait for the installation process to complete.

Silent installation

1. Make sure you are logged in as an administrator
2. Review the contents of the response file template `setupaix.opt` and modify it to reflect your requirements (if you are installing from a CD you need to copy that file to a non read-only file system location before editing it).
3. Run `setupsolaris -options setupsolaris.opt –silent`

Solving installation problems

In the case of installation problems, please review the installation log file `/tmp/mbfeinstlog.txt`. At the end of the installation this file is copied to the <mbfeopt_root>/mdv directory for future reference.

If the problem persists, or if you suspect that the problem lies with the installation program itself, you can try to rerun the installation program with the additional command line parameter `–is:log /tmp/ishield.log` and then inspect the `ishield.log` file produced.

If the prerequisite WebSphere Business Integration Message Broker 5.0 with Fix Pack 05 product was not found during the setup and you decided to proceed with the installation, and you install the broker at a later time, you must remember to link these two files from the broker `/opt/mqsi/jplugin` directory:

```
<mebfeopt_root>/bin/mbfe.jar
<mebfeopt_root>/bin/mbfe.path.jar
```

If the prerequisite WebSphere Message Broker 6 product was not found during the setup and you decided to carry on with the installation, and you install the broker at a later time, you must remember to link this file from under the `$MQSI_WORKPATH\common\profiles` directory:

`<mbfeopt_root>/bin/mbfeprofile.sh`.

Configuring for first use

Importing node definitions into the broker toolkit on Windows or Linux

In order to be able develop new message flows including file nodes, the nodes palette included broker toolkit must be extended with the file nodes definitions.

The steps needed to extend the palette of nodes available in the tooling are automatically performed during the installation.

For WebSphere Business Integration Message Broker 5 Toolkit you only need to allow any active instances of the toolkit to restart in order to pick up the changes.

For some WebSphere Message Broker 6 Toolkit installations just allowing any active instances of the toolkit to restart is not sufficient to refresh the broker nodes palette; in those cases these extra steps are needed:

1. Windows
    a. Locate your toolkit installation directory (you can do this by inspecting the associated Windows Start menu shortcut
    b. Open a command prompt windows
    c. Make your toolkit installation directory current
    d. Run `wmbt.exe -clean`
2. Linux
    a. Locate your toolkit installation directory (if you accepted the installation defaults this will be `/opt/ibm/MessageBrokersToolkit/6.0/wmbt.bin`
    b. Open a shell window
    c. Make your toolkit installation directory current
    d. Run `wmbt.bin -clean`

Notes

Using the –clean option when starting the toolkit is required only the first time after WebSphere Message Broker File Extender is installed.

Starting the toolkit with the –clean option normally requires several minutes before the toolkit main window is displayed.

For more details on why the –clean option is needed you can browse this technote: http://www-1.ibm.com/support/docview.wss?uid=swg21220271

Preparation of the runtime environment

WebSphere Business Integration Message Broker 5

On Windows, the environment variables such as PATH are updated by the installation program; the exact updates made depend on the components that you have installed and the directories into which you have installed them.

On UNIX™ systems, the installation program **does not** update environment variables.

Sample profile files are provided with WebSphere Message Broker File Extender, and you must check their content to ensure the variables are set correctly for your environment before you use the product or configure any resources. The path and filename of the profile is dependent on your platform

For AIX: `<mbfeopt_root>/profiles/profile.aix`

For Linux: `<mbfeopt_root>/profiles/profile.lnx`

For Solaris: `<mbfeopt_root>/profiles/profile.sol`

WebSphere Message Broker 6

With this version of the broker, on both Windows and on UNIX systems you need to initialize the environment before you can invoke broker and file extender commands to broker and file extender runtime components, such as:

```
mqsilist
mbfecfgqmgr
```

This initialization ensures that all the commands you issue interact with the correct installation of the code.

Here is how to set up a command environment:

1. Windows , open a command console. Click **Start > IBM > WebSphere Message Brokers 6.0 > Command Console** for the appropriate broker installation.
2. On Linux and UNIX systems, run the environment profile as shown below:
   ```
   . <brokerinstalldir>/bin/mqsiprofile
   ```
   Please note that you must include the period and space preceding the location for this invocation to work correctly.

For more information on this subject you can refer to WebSphere Message Broker 6 infocenter topic http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp?topic=/com.ibm.etools.mft.doc/an26190_.htm

Preparation of the broker queue manager

Before being able to run the WebSphere Message Broker File Extender installation verification procedure, samples or custom message flows using the file nodes on a broker, several queues and other structures must be defined.

WebSphere Message Broker File Extender provides a special command to assist in creating these definition (see details below.) This command is located in the <mbfebin_home>/bin directory, and administrative access rights to the parameter queue manager are needed to run it.

This command must be run against **each** broker queue manager where message flows containing file nodes need to be deployed.

Windows syntax

```
mbfecfgqmgr brokerQMgr [-redo | -reset] [-silent] [-nosamples]
```

Unix syntax

```
mbfecfgqmgr.sh brokerQMgr [-redo | -reset] [-silent] [-nosamples]
```

Where:

- `brokerQMgr` is the name of the .queue manager hosting the broker that will run WebSphere Message Broker File Extender based message flows.
- `-reset` is the optional switch that can be used to completely reset the WebSphere Message Broker File Extender internal file repository and operational configuration data, undoing the effects of a previous successful or partially failed execution of the **mbfecfgqmgr** command; this option must be used with care on production systems because it will cause the deletion of in flight business file data
- `-redo` is the optional switch that can be used to completely reset and recreate the WebSphere Message Broker File Extender internal file repository and operational configuration data, undoing the effects of a previous successful or partially failed execution of the **mbfecfgqmgr** command; this option must be used with care on production systems because it will cause the deletion of in flight business file data
- `-silent` is the optional switch that will toggle off the interactive confirmation prompt associated with `-reset` and `-redo` switches; `-silent` is intended to be used when **mbfecfgqmgr** is invoked from within non-interactive shell scripts.
- `-nosamples` is the optional switch that can be used to avoid the creation of WebSphere MQ queues used by the product samples. By default these queues will be created.

Please note that given the wide impact of `-reset` and `-redo` switches on established WebSphere Message Broker File Extender installations, they cannot be executed when the broker is running.

Broker restart

As a final step before being able to successfully deploy your first WebSphere Message Broker File Extender-based flow to a newly configured broker, you need to allow the broker to restart.

Migrating message flows to WebSphere Message Broker V6

Migrating a message flow containing FileInput, FileOutput or FileProxy nodes from WebSphere Business Integration Message Broker 5 to WebSphere Message Broker 6 does not involve any particular WebSphere Message Broker File Extender related action.

Please refer to WebSphere Message Broker 6 for general guidance on how to migrate existing message flows to this version of the broker.

Installation verification procedure

To ensure your WebSphere Message Broker File Extender system has been correctly installed configured, an Installation Verification Procedure (*IVP*) is provided.

The IVP is made of two components:

- An extremely simple F2F message flow that simply changes the name of the input file received on the FileInput node and forwards it to the FileOutput node without touching the file content.
- A script that creates a small file that will be read by the IVP and waits for the same file to appear with a different name on the output directory.

To run the IVP, deploy the provided broker archive file named <mqbfeopt_root>/bin/mbfeivp.bar to the selected broker using the WebSphere Business Integration Message Broker toolkit or WebSphere Business Integration Message Broker *mqsideploy* command.

On the target broker machine you can then run the script driving the deployed IVP message flow, as shown below.

Windows syntax

```
mbfeivp
```

Unix syntax

```
mbfeivp.sh
```

Below is an example of a typical session transcript:

```
C:\Program Files\IBM\MBFE\bin>mbfeivp.bat

-----------------------------------------------------------
start Test
----------------------------------------------------------

output file mbfeivp_out successfully created

Fri 05/06/2005    17:59:18.81

-----------------------------------------------------------
Test successfully completed
-----------------------------------------------------------
```

Uninstalling

Windows

1. Stop all running processes that use any part of WebSphere Business Integration Message Broker before you uninstall
2. Navigate to **Start > Settings > Control Panel > Add/Remove Programs**
3. Select the WebSphere Message Broker File Extender feature and follow the wizard panels to completely remove the product.

AIX

1. Stop all running processes that use any part of WebSphere Business Integration Message Broker before you uninstall
2. Run the `<mbfeopt_root>/_uninst/uninstaller` program to completely remove the product; optionally add the `-console` command line switch if you want to perform a console based uninstall.

Linux

1. Stop all running processes that use any part of WebSphere Business Integration Message Broker before you uninstall
2. Run the `<mbfeopt_root>/_uninst/uninstaller.bin` program to completely remove the product; optionally add the `-console` command line switch if you want to perform a console based uninstall.

Solaris

1. Stop all running processes that use any part of WebSphere Business Integration Message Broker before you uninstall
2. Run the `<mbfeopt_root>/_uninst/uninstaller.bin` program to completely remove the product; optionally add the `-console` command line switch if you want to perform a console based uninstall.

Chapter 3. Configuration

This section contains reference information on WebSphere Message Broker File Extender node terminals, attributes and on the message tree structure created/managed by the nodes at runtime.

FileInput Node Reference

Terminals of a FileInput Node

Input terminals

The FileInput node has no input terminals.

Output terminals

Terminal	**Description**
Out	A terminal through which a message is routed each time FileInput Node reads data from a file.
Catch	A terminal through which an exception list and possibly some data read from the file are routed, if an exception is thrown, and not handled, during propagation to the *Out* terminal.
Failure	A terminal through which a failure message is routed if there is a failure within a node.

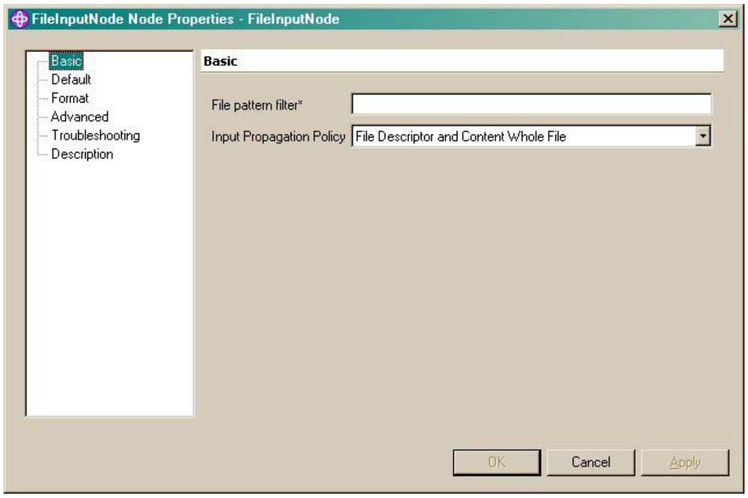
Synch	A terminal that is used only when you wish your transaction to be synchronized with the transaction involving other broker resources such as WebSphere MQ and/or the database.
	In the above mentioned cases the terminal must be connected to a Compute node containing this fixed ESQL:
	`SET OutputRoot.MQMD = InputRoot.MQMD;`
	`SET OutputRoot.BLOB = InputBody;`
	and the compute node output must be connected to an MQReply node (with all properties left at their default values).

Properties of a FileInput Node

The column headed M indicates whether the property is mandatory; that is, whether you must enter a value if no default value is defined.

The column headed C indicates whether the property is configurable; that is, whether you can change the value in the .bar file.

Basic properties



File pattern filter*

Input Propagation Policy: File Descriptor and Content Whole File

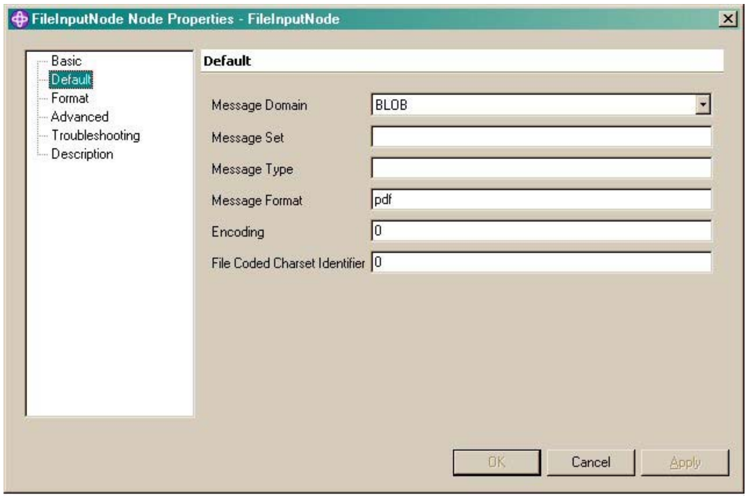
Property	M	C	Default	Description

File Pattern Filter	Yes	Yes	Blank	Path name and optional file filter ("*" and "?" wildcards allowed) within the specified directory where input files are read.
				To read all files contained in a directory specify the **full path name followed by a slash and a "*" character**.
				Examples
				`/var/mydata/*` will process all files contained in the directory named `/var/mydata`
				`/var/mydata/*.txt` will process all files with extension txt contained in the directory named `/var/mydata`
				`/var/mydata/mydata1` will

Property	M	C	Default	Description
				process **only one file** named mydata1 contained in the directory named /var/mydata /var/mydata/mydata1.dat will process **only one file** named mydata1.dat contained in the directory named /var/mydata If the specified path is not an absolute one, <mbfevar_root>/filedata/inbox is assumed as the root directory for the partial path name specified. This default value can be changed by altering the mbfe.properties configuration file.
Input Propagation Policy	Yes	No	**File Descriptor and Content Whole File**	Specifies how the FileInput node will propagate the input file content to the message flow. Valid values are: **File Descriptor and Content Whole File** - in this case, the FileInput node will propagate the file descriptor and the whole file content. We do not recommend using this option for very big files **File Descriptor and Content Record by Record** - in this case, on each iteration, the FileInput node will propagate the file descriptor and the content of a record read sequentially from the input file. **File Descriptor only** - in this case, the FileInput node will propagate just the file descriptor. The file content is not propagated. This option can be used to implement rule-based file

Property	M	C	Default	Description
				routing using WebSphere Business Integration Message Broker.

Default properties



FileInputNode Node Properties - FileInputNode

Default

Message Domain: BLOB

Message Set:

Message Type:

Message Format: pdf

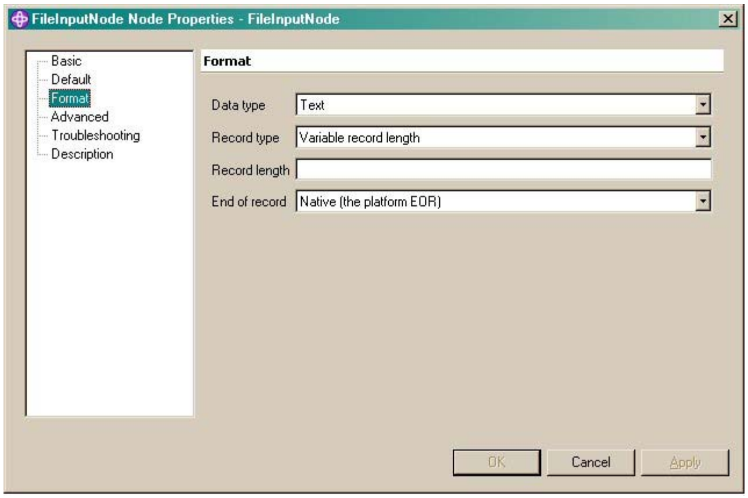
Encoding: 0

File Coded Charset Identifier: 0

Property	M	C	Default	Description
Message Domain	No	Yes	BLOB	Specifies the message domain to be used in parsing the message.
Message Set	No	Yes	Blank	Specifies the message set to be used in parsing the message.
Message Type	No	Yes	Blank	Specifies the message type to be used in parsing the message.
Message Format	No	Yes	Blank	Specifies the message format to be used in parsing the message.
File Coded Charset Identifier	No	No	0	Specifies the code page to be used when parsing the message. This is typically set to zero (the default value) in order to inherit the machine default code page settings.
Encoding	No	No	0	Specifies the encoding to be used when parsing the message. This is typically set to zero (the default value) in order

				to inherit the machine default encoding settings

Format properties



Format

Data type: Text

Record type: Variable record length

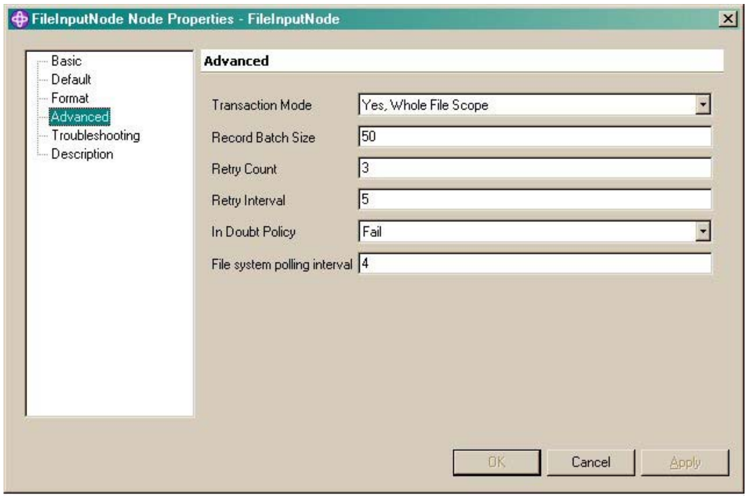
Record length:

End of record: Native (the platform EOR)

Property	M	C	Default	Description
Data Type	Yes	No	**Text**	Data type contained in the file being read. Valid values are: **Binary** - the file is read by FileInput as binary (use this option also when dealing with files without a record structure). **Text –** the file is read by FileInput as text (use this option also when dealing with files with a record structure).
Record Type	Yes	No	**Variable Record Length**	In case of record structured files, this parameter specifies the type of record contained in the file being read. Valid values are: **Fixed Record Length** - Records will be read from the file as fixed length records. The length of each record is determined by the *Record Length* property value.

Property	M	C	Default	Description
				Variable Record Length - Records will be read from the file as variable length records.
Record Length	No	No	0	When *Record Type* is set to **Fixed Record Length**, the value of this property is used to drive the reading of the file.
End of Record Type	Yes	Yes	**Native**	In case of record structured files this parameter specifies the type of record terminator Valid values are: **Native** - Default native record separator for the platform (for example, **CR+LF** on Windows and **CR** on Unix) **None** -No record separator **CR -** Carriage Return character **LF -** Line feed character **CR+LF -** Combination of Carriage Return and Line Feed characters

Advanced properties



Transaction Mode: Yes, Whole File Scope

Record Batch Size: 50

Retry Count: 3

Retry Interval: 5

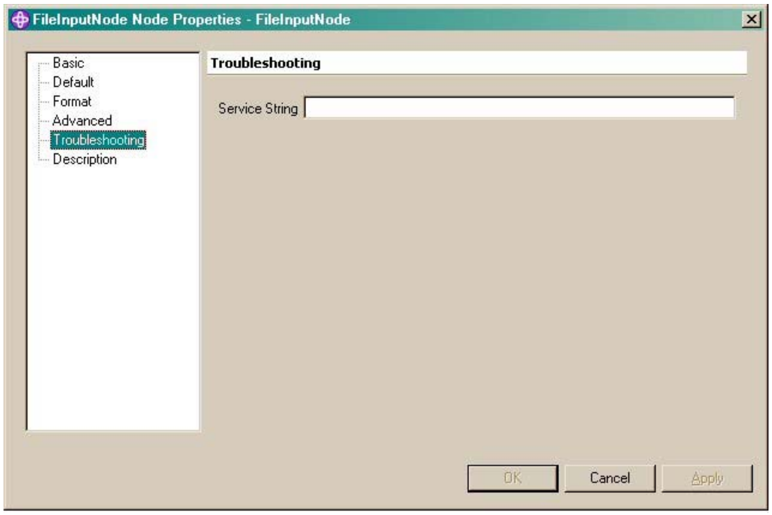
In Doubt Policy: Fail

File system polling interval: 4

Property	**M**	**C**	**Default**	**Description**
Transaction Mode	Yes	Yes	**Yes, Whole File Scope**	Specifies how the input file is to be handled by FileInput node Valid values are: **Yes, Whole File Scope** In this case, the entire input file will be handled within a single unit of work. **Yes, Record Batch Scope** In this case a specified number of file records will be handled within a single unit of work. If you select this option, you must specify the number of records in the *Record Batch Size* field. **No** In this case no unit of work logic is applied to file content handling.
Record Batch Size	No	Yes	50	Required only if *Transaction Mode*= *Yes, Record Batch Scope*, it specifies the number of records to be included in each batch.
Retry Count	No	Yes	3	Number of retries to be attempted if the FileInput Node detects a problem during the message flow execution
Retry Interval	No	Yes	5	Specifies the interval in seconds between successive retries.
In-Doubt Policy	Yes	Yes	**Fail**	Specifies how transaction-related problems are to be handled. Valid values are: **Fail** The in-doubt transaction will fail, with the error data being written in the FileInput node failure terminal for inspection by human users or recovery routines. **Redo** The in-doubt transaction will be re-done,

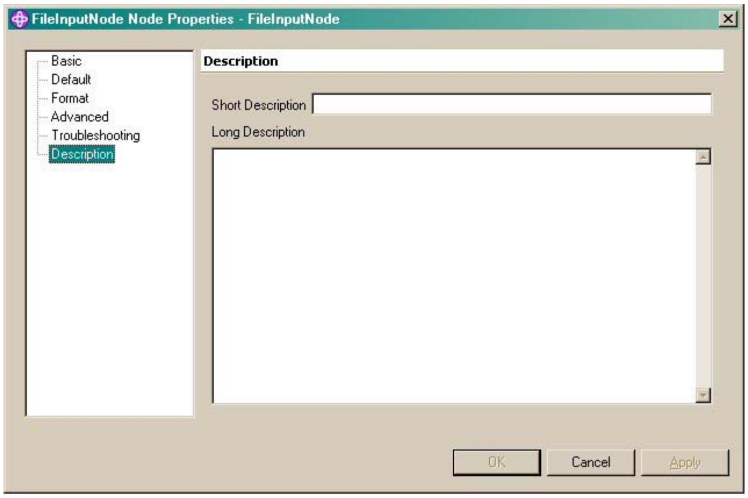
Property	M	C	Default	Description
				potentially implying double processing for some of the input data. **Skip** The in-doubt transaction will be skipped, potentially implying skipping processing for some of the input data.
File system polling interval	Yes	Yes	4	Number of seconds the nodes sleeps when polling the input directory

Troubleshooting properties



Property	M	C	Default	Description
Service String	No	Yes	Blank	Reserved. Use this field only if instructed to do so by IBM support personnel.

Description properties



Property	M	C	Default	Description
Short Description	No	Yes	Blank	Provides a short user description of the role of the node within the message flow.
Long Description	No	Yes	Blank	Provides a longer user description of the rode of the node within the message flow.

FileOutput Node Reference

Terminals of a FileOutput Node

Input terminals

Terminal	Description
in	The input terminal that receives the input content (for example, a WebSphere MQ message or a file record read by FileInput node).

Output terminals

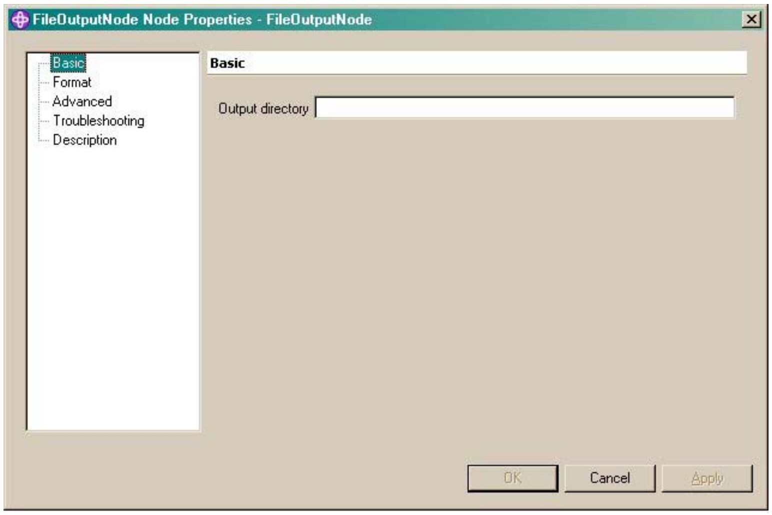
The FileOutput node has no output terminals

Properties of a FileOutput Node

The column headed M indicates whether the property is mandatory; that is, whether you must enter a value if no default value is defined.

The column headed C indicates whether the property is configurable; that is, whether you can change the value in the .bar file.

Basic properties

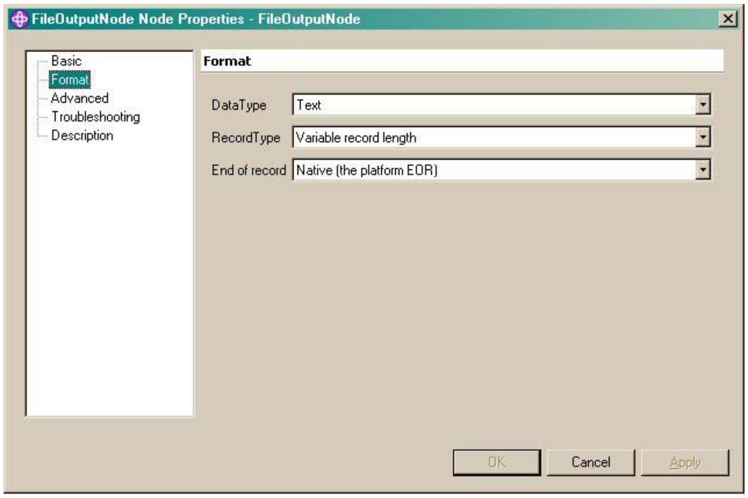


Basic

Output directory

Property	M	C	Default	Description
Output Directory	Yes	Yes	Blank	Output directory where files are written. In case the specified path is not an absolute one, <mbfevar_root>/filedata/outbox is assumed as the root directory for the partial path name specified. This default value can be changed by altering the mbfe.properties configuration file.

Format properties



FileOutputNode Node Properties - FileOutputNode

Format

DataType: Text

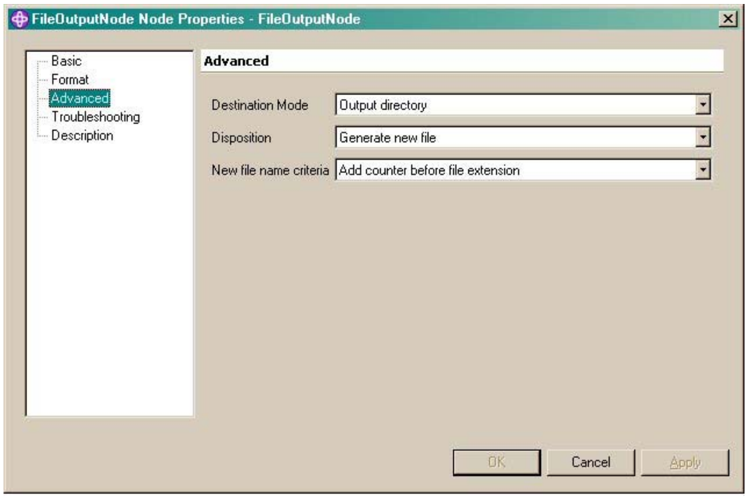
RecordType: Variable record length

End of record: Native (the platform EOR)

Property	M	C	Default	Description
Data Type	Yes	No	**Text**	Data type contained in the file being written. Valid values are: **Binary** - the file written by FileOutput node will be written as binary. **Text -** the file written by FileOutput node will be written as text **Inherited** - the Data Type attribute is inherited from the LocalEnvironment settings using the property DataType in the MBFEProperties subfolder.
Record Type	Yes	No	**Variable Record Length**	Type of record contained in the file being written. Valid values are: **Fixed Record Length** - Records will be written to the file as fixed length records. The length of each record is determined by the length of the first record written to the file. Subsequent records showing a different length will cause a recoverable exception to be thrown by the node **Variable Record Length -** Records will be written to the file as variable length records

Property	M	C	Default	Description
				(no record length check performed). **Stream (no record)** - The file being written will not contain any particular record structure.
End of Record Type	Yes	Yes	**Native**	Type of record terminator Valid values are: **Native** - Default native record separator for the platform (for example, **CR+LF** on Windows and **CR** on Unix) **None** -No record separator **CR -** Carriage Return character **LF -** Line feed character **CR+LF -** Combination of Carriage Return and Line Feed characters **Inherited** - the *End of Record Type* attribute is inherited from the LocalEnvironment settings using the property EorType in the MBFEProperties subfolder

Advanced properties



Destination Mode: Output directory

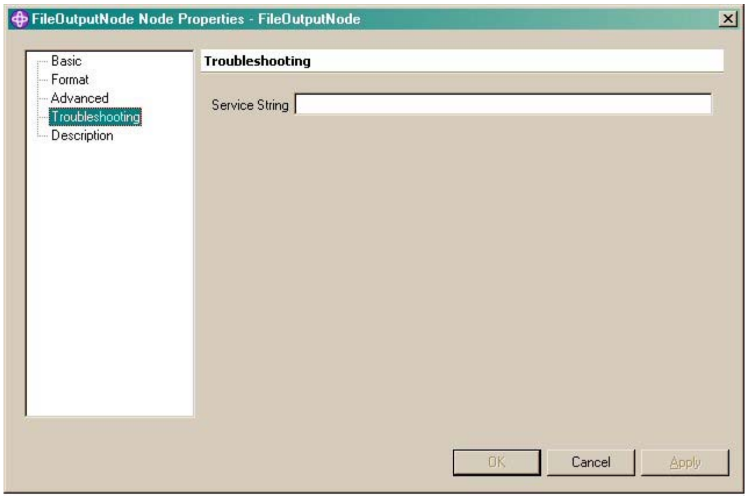
Disposition: Generate new file

New file name criteria: Add counter before file extension

Property	M	C	Default	Description
Destination Mode	Yes	No	**Output directory**	Defines the directories to which the file is written. Valid values are:

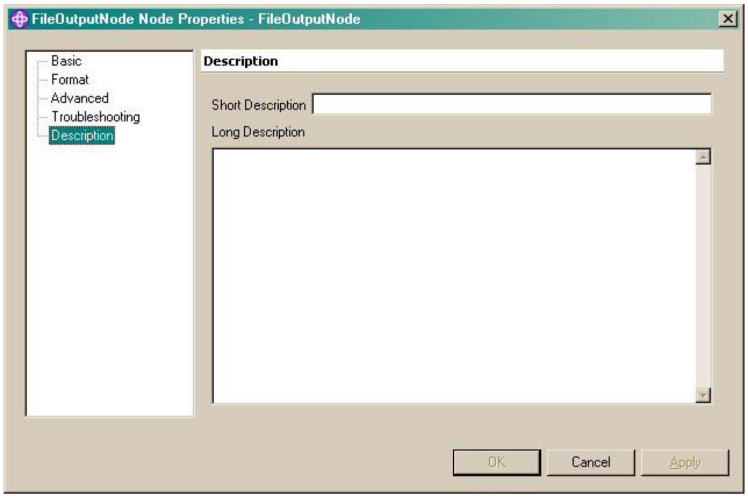
Property	M	C	Default	Description
				Output directory – when this value is selected, the directory name is taken from *Output Directory* node property **Output Directory List** – when this value is selected, the destination directories are dynamically taken from the DestinationList specified in LocalEnviroment.
Disposition	Yes	No	**Generate New File**	Defines how output file creation is to be handled. Valid values are: **Append** – when this value is selected and a file named as the current file already exists, the content is appended to that file. **Replace** – when this value is selected and a file named as the current file already exists, the already existing file is overwritten. **Generate New File** – when this value is selected a unique name for the file is generated.
New File Name Criteria	No	No	**Add counter before file extension**	If **Generate New File** has been selected for *Disposition*, this property controls the naming scheme adopted to make the file name unique. Valid values are: **Add counter at the beginning** **Add counter at the end** **Add counter before file extension**

Troubleshooting properties



Property	M	C	Default	Description
Service String	No	Yes	Blank	Use this field only if instructed to do so by IBM technical support personnel.

Description properties



Property	M	C	Default	Description
Short Description	No	Yes	Blank	Provides a short user description of the role of the node within the message flow.
Long Description	No	Yes	Blank	Provides a longer user description of the role of the node within the message flow.

FileInput error handling

Given that the FileInput is an input node, it drives error handling actions for the whole message flow.

The general error handling policies implemented by the FileInput node are similar but not identical to those implemented by the MQInput node (for MQInput node error handling information please check http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp?topic=/com.ibm.etools.mft.doc/ac00414_.htm**).**

The following paragraphs describe the system behavior in the event of an exception following an initial read attempt, in four scenarios that differ in the number and type of FileInput terminals that are actually used by the message flow programmer:

1. *FileInput with just an Out terminal attached*: the current batch of records or the whole file body (depending on File Propagation Policy attribute settings) will be re-propagated through the Out terminal for up to Retry Count times; then, if the error condition persists, the input file is marked as poison and will skip processing in subsequent message flow iterations; error messages are issued and the administrator can decide what to do with the problem file using the provided **mbfesolve** command.
2. *FileInput with Out and Catch terminals attached*: the current record or the whole file body (depending on File Propagation Policy attribute settings) will be propagated through the *Catch* terminal and then FileInput will commit and continue.
   If the Catch subflow itself hits an exception condition, the FileInput will be re-propagated through the Out and Catch terminals for up to Retry Count times; then, if the error condition persists, the input file is marked as poison and will skip processing in subsequent message flow iterations; error messages are issued and the administrator can decide what to do with the problem file using the provided **mbfesolve** command.
3. *FileInput with Out and Fail terminals attached*: the current batch of records or the whole file body (depending on File Propagation Policy attribute settings) will be re-propagated through the Out terminal for up to Retry Count times; then, if the error condition persists, the current batch of records or the whole file body is propagated through the Fail terminal. After this, the node will process subsequent records in the same file or the next file, depending on the File Propagation Policy attribute settings. In case another exception condition is encountered in the Failure subflow, the input file is marked as poison and will skip processing in subsequent message flow iterations; error messages are issued and the administrator can decide what to do with the problem file using the provided **mbfesolve** command.
4. *FileInput with Out, Fail and Catch terminals attached*: this case is a combination of the two previous scenarios; in the event of an exception,

Catch logic is executed first and Fail logic is executed only in the case of a repeated failure in the Catch, just before marking the file as poison.

FileProxy Node Reference

Terminals of a FileProxy node

Input terminals

Terminal	Description
From_out	This is the input terminal connected to the output terminal of an MQInput node.
From_catch	This is the input terminal connected to the *catch* terminal of a MQInput node. You must connect to this terminal if and only if the *catch* terminal of the corresponding MQInput node was originally connected.
From_failure	This is the input terminal connected to the *failure* terminal of an MQInput node. You must connect to this terminal if and only if the *failure* terminal at the corresponding MQInput node was originally connected.

Output terminals

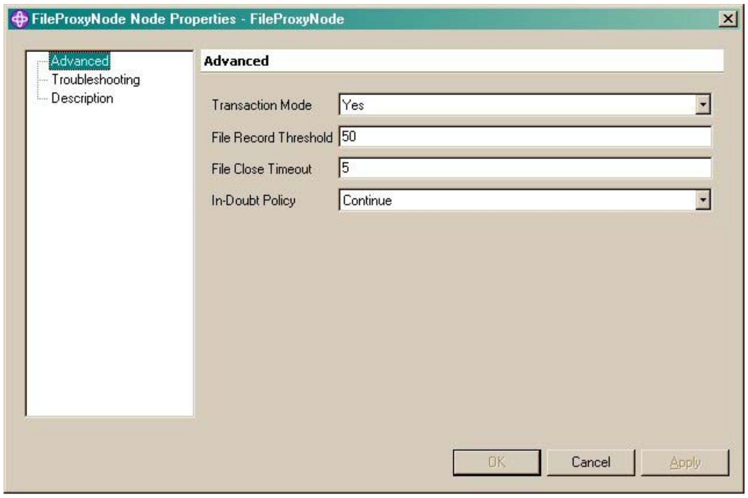
Terminal	Description
Out	This is the standard node output terminal, to which correct output is written.
Catch	This is a terminal that you can connect to an exception-handling subflow.
Failure	This is the terminal to which details of a failure are written.
Synch	An optionally used terminal that should be connected to an MQReply node (with all properties left at default values) when you wish your transaction to be synchronized with the transaction involving other broker resources such as WebSphere MQ and/or the database.

Properties of a FileProxy Node

The column headed M indicates whether the property is mandatory; that is, whether you must enter a value if no default value is defined.

The column headed C indicates whether the property is configurable; that is, whether you can change the value in the .bar file.

Advanced properties



FileProxyNode Node Properties - FileProxyNode

Advanced

Transaction Mode: Yes

File Record Threshold: 50

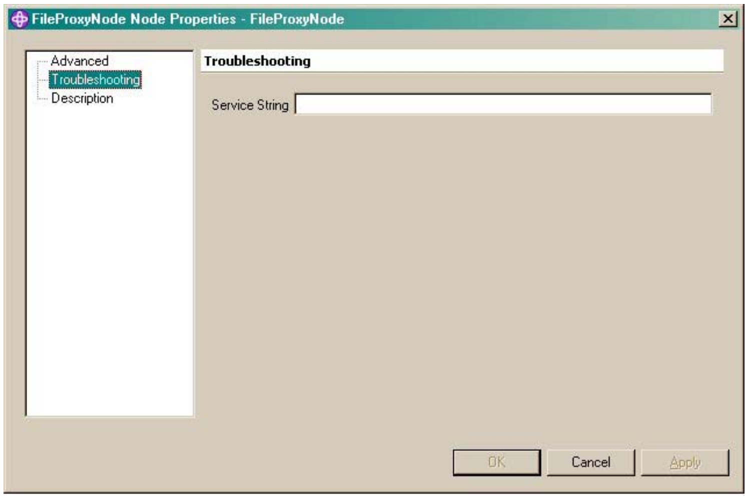
File Close Timeout: 5

In-Doubt Policy: Continue

Property	M	C	Default	Description
Transaction Mode	Yes	No	**Yes**	Specifies how the output file is to be handled by FileOutput nodes controlled by FileProxy node. Valid values are: **Yes** In this case, the input messages are appended to the output file in sync with the WebSphere MQ outstanding unit of work. Use this option only when the MQInput node corresponding to this instance of the FileProxy node is operating within syncpoint. **No** In this case, the input messages are appended to the output file regardless of WebSphere MQ unit of work settings.
File Record Threshold	No	Yes	50	Required only if *Transaction Mode* is **Yes**. Specifies the maximum number of records after which an output file is to be closed and a new one created.
File Close Timeout	No	Yes	5	Specifies the number of seconds that an output file is to

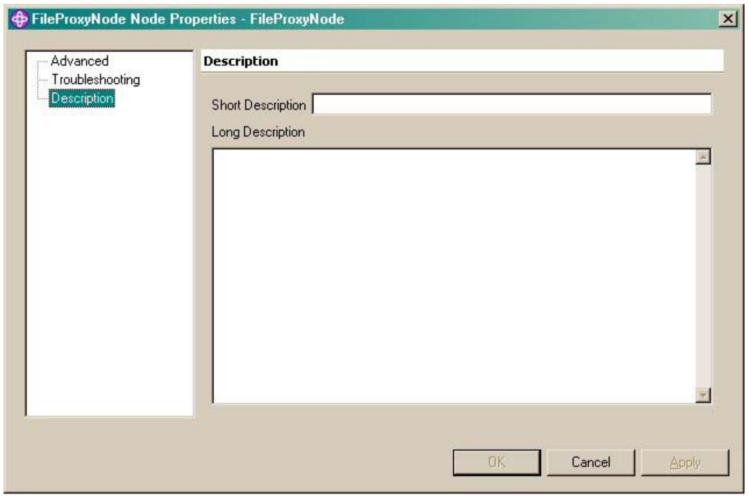
Property	M	C	Default	Description
				be kept open without appending new content to it. When this timeout expires, the file is closed and made available in the file system in the specified location.
In-Doubt Policy	Yes	No	**Continue**	Specifies how transaction-related problems are to be handled. Valid values are: **Continue** Output file records that are generated during in-doubt transaction execution are retained. Output data generation will proceed, appending records to those files. **Clear** A new transaction is started, without any attempt to recover any in-doubt work.

Troubleshooting properties



Property	M	C	Default	Description
Service String	No	Yes	Blank	Use this field only if instructed to do so by IBM technical support personnel.

Description properties



Property	M	C	Default	Description
Short Description	No	Yes	Blank	Provides a short user description of the role of the node within the message flow.
Long Description	No	Yes	Blank	Provides a longer user description of the role of the node within the message flow.

Message Tree Structures

File Descriptor

The file descriptor (MBFEProperties) is a set of attributes that are created by a FileInput when reading the file in order to identify characteristics.

A reduced version of this descriptor can be created/modified using ESQL in a Compute node before FileOutput in order to drive its behavior (for example, setting the file name to a context dependent value, like a timestamp or the result of a database lookup).

Properties

Root Element	LocalEnvironment.Variables.MBFEProperties			
Property	**Data type**	**FileInput Access**	**FileOutput Access**	**Description**
SessionId	Integer	Write	N/A	Unique identifier assigned by the system to the

				current file management session. Some error messages report this value, so it can be used to help with troubleshooting.
RetryCount	Integer	Write	N/A	Number of times a transient error operation has been tried unsuccessfully on the file.
DataType	Integer	Write	Read	It specifies the type of data contained in the file being handled. Valid values are: 0 – Text file 1 – Binary file
RecordType	Integer	Write	Read	It specifies the type of records contained in the file being handled. Valid values are: 1 – Fixed length records 2 – Variable length records 4 – No record structure
FileId	Character	Write	Read	Unique identifier assigned to the file by the system. It can be used in audit trails or for other tracking purposes.
FileName	Character	Write	Read	Name of the input file (for FileInput nodes) or output file (for FileOutput nodes)
Directory	Character	Write	Read	Directory where the file is read (for FileInput nodes) or written (for FileOutput nodes)
EorType	Integer	Write	Read	It specifies the type of record terminator (EOR) associated to the file. 0 – Platform native EOR 2 – LF character EOR 4 – CR+LF characters EOR 16 – No EOR character
CodedCharSetId	Integer	Write	Read	CodedCharSetId associated to the file, a zero value means that the platform default is inherited.
RecordNumber	Integer	Write	N/A	Position of the current record in the file
Size	Integer	Write	N/A	File size in bytes
Time	Timestamp	Write	N/A	Last modification timestamp of the file
Expiry	Character	Write	N/A	Timestamp after which the file is automatically removed from the system if still unprocessed.
Encoding	Character	Write	Read	Encoding associated to the file, a zero value

				means that the platform default is inherited.

File Destination List Descriptor

The Destination List descriptor is a set of possibly **repeated** DestinationData elements that are honored by a FileOutput node configured to use an **Output Directory List** as *Destination Mode,* in order to create the same file in multiple target directories.
Each DestinationData element contains the name of a target directory.

It is the user's responsibility to create an appropriate DestinationList at runtime using ESQL.

Properties

Root Element	LocalEnvironment.Destination.MBFE.DestinationData (where there may be several *DestinationData* children elements)			
Property	**Data type**	**FileInput Access**	**FileOutput Access**	**Description**
Directory	Character	N/A	Read	Name of the target directory where the file is to be written.

Usage example

The file Destination List feature can be used for creating an output file on one or more output directories. The actual name of the file and the name of the output directory can be dynamically decided at run time within a message flow.

A very simple Destination List that instructs a FileOutput node to create a file named myFile.dat in /var/myFiles can be established in a compute node by coding the following ESQL:

```
SET OutputRoot = InputRoot;
SET OutputLocalEnvironment = InputLocalEnvironment;
SET
OutputLocalEnvironment.Destination.MBFE.DestinationData.Directory
= '/var/myFiles';
SET OutputLocalEnvironment.Variables.MBFEProperties.FileName =
'myFile.dat';
```

Moreover you also need to ensure that:

1. The compute node is configured to deal with **LocalEnvironment and Message** and not just **Message** (see also the "Planning Considerations" section in Chapter 1 for other message flow development related constraints).

2. The FileOutput node *Destination mode* attribute is set to **Output Directory List**

File Action Tag

The file descriptor (MBFEProperties) also hosts a special attribute named *action* that can be created via ESQL in M2F flows in order to control when an output file can be closed, based on the content of an input message.

Properties

Root Element	LocalEnvironment.Variables.MBFEProperties			
Property	**Data type**	**FileInput Access**	**FileOutput Access**	**Description**
action	Character	N/A	Read	When present, this attribute causes all the FileOutput instances in the flow to close the file they are appending records to and release it to the directory specified by the user. Valid values are: **close** - The current message is appended to the file and the file is closed **skipAndClose** – The current message is skipped and the file is closed.

Advanced tasks

This section covers some advanced aspects of WebSphere Message Broker File Extender configuration and operation.

Altering file nodes behavior by editing *mbfe.properties* file

There are some aspects of WebSphere Message Broker File Extender runtime behavior that can only be controlled by editing the *mbfe.properties* configuration file.

A template of this file with reasonable values is shipped with the product. It is not expected that you should need to change these values under normal circumstances.

Any changes to this file are effective only after a broker restart.

mbfe.properties configuration file is located under the <mbfevar_home>/config installation directory and can be edited using a text editor.

Notes

By default *mbfe.properties* configuration settings are system wide; in case you need to limit their scope to a single broker or a single execution group within a broker you must create a copy of *mbfe.properties* in <mbfevar_home>/config named respectively *mbfe.<broker>.properties* or *mbfe.<broker>.<executionGroup>.properties*, where *broker* is the name of your broker and *executionGroup* is the name of the execution group.

For example, when a new message flow instance is started in broker MYBROKER and execution group MYEXEGRP, WebSphere Message Broker File Extender will read the whole configuration from the first valid properties file instance detected by looking up the file system for the following files in the following order:

1. <mbfevar_home>/config/mbfe.MYBROKER.MYEXEGRP.properties
2. <mbfevar_home>/config/mbfe.MYBROKER.properties
3. <mbfevar_home>/config/mbfe.properties

Once an eligible properties file is found, no other attempt at reading any other file is performed, so each of the properties files must contain **all** the mandatory properties.

Properties

Property	**Description**
trc.level	Switches on and off tracing Valid values are: **NO** – No trace files are created **YES** – Tracing is turned on. **VERBOSE** – Verbose tracing is turned on.
trc.filePath	Directory where trace files are created.
trc.maxFileSize	Maximum size in bytes that the current trace file can reach before it is renamed and a new empty trace file becomes current. If this parameter is absent or is not set to a value greater than zero, all traces will be appended to a single file without size limitations.
trc.maxBackupIndex	If trc.maxFileSize is set to a value greater than zero, this attribute indicates the maximum number of trace backup files that are to be saved in a circular tracing scenario.
baseInputDir	Base input directory to use in case a blank or relative (non absolute) path name is specified in a FileInput node *File Pattern Filter* attribute.

	This property defaults to: <mbfevar_home>/filedata/inbox. If you change this value the scripts used by the Product installation verification procedure and samples will need to be altered accordingly.
baseOutputDir	Base output directory to use in case a blank or relative (non absolute) path name is specified in a FileOutput node *Output Directory* attribute or *DestinationData.Directory* destination list member element. This property defaults to: <mbfevar_home>/filedata/outbox. If you change this value the scripts used by the Product installation verification procedure and samples will need to be altered accordingly.
fsRepository.newFilesExpiryInterval	Number of days after which unprocessed files are automatically removed from WebSphere Message Broker File Extender internal repository. This property defaults to **60 days**. To inhibit this feature delete or comment the line containing this property.
freezePoisonFiles	Parameter that controls the handling of poison files, that is, files for which WebSphere Message Broker File Extender has encountered an error that was not recovered within the maximum number of retries configured. Valid values are: **true** (default and recommended) – The poison file will be kept in the internal repository, no further attempts to process it will be performed and the normal processing will continue on the other files queued up behind it. **false** – the FileInput node will keep iterating on the poison file (even if no further attempts to process it are made), so in case the message flow has not been deployed with multiple instances, the poison file will effectively prevent processing of other correct files queued up behind it. Use this option only in case you need to preserve a strict FIFO order in the processing of files.

Turning on service trace

To create a service trace follow these steps:

1. Stop the broker
2. Set **trc.Level=VERBOSE** in the <mbfevar_root>/config/mbfe.properties configuration file
3. Start the broker
4. Run your use cases
5. Stop the broker
6. Set **trc.Level=NO** in the <mbfevar_root>/config/mbfe.properties configuration file
7. Start the broker
8. Collect trace files that were created in the <mbfevar_root>/log directory or in the non-default directory you specified in the **trc.filePath** key in the <mbfevar_root>/config/mbfe.properties configuration file

Handling poison files using *mbfesolve* tool

As discussed in the previous section, we use the term *poison file* to indicate a file for which WebSphere Message Broker File Extender has encountered an error that was not recovered within the maximum number of retries.

After some problem determination has been performed regarding the causes of the problem condition, there are typically three types of recovery actions that you may decide to perform:

1. *Extract the file from the internal repository*, for example, there may be cases when you need to correct the file content manually before resubmitting it
2. *Reset the retry counter associated with the file*, for example, because the error condition was transient, and is now corrected but it was not recovered in time,
3. *Remove the file from the internal repository,* for example, because the file was submitted by mistake and it is completely extraneous to the business message flow where it ended up.

WebSphere Message Broker File Extender provides a special command, located in the <mbfeopt_root>/bin directory, that can assist you in recovering from the scenarios described above.

Windows syntax

```
mbfesolve   –t session –a action
            –q brokerQMgr –b broker -e executionGroup
            [-p backupPath]
```

Unix syntax

```
mbfesolve.sh –t session –a action
             –q brokerQMgr –b broker -e executionGroup
             [-p backupPath]
```

Where:

- `session` is the identifier of the session containing the problem file, as indicated in the error message notifying the presence of the poison file; the special value **ALL** will apply the command action to all in flight sessions.
- `action` is one of the following values:
  - `view` to list the status of a specific session (or all in flight sessions)
  - `restore` to restore the poison file associated with `session` (or all in flight sessions) by moving it out of the internal repository, either onto the original file directory or onto the optionally specified `backupPath`
  - `remove to remove the poison file associated with session (or all in flight sessions)`
- `brokerQmgr` name of the queue manager hosting the broker
- `broker` name of the broker
- `executionGroup` name of the execution group within the broker.

Chapter 4. Using file nodes in a message flow

Usage examples in simple scenarios

In the next few sections we will go through three very simple WebSphere Message Broker File Extender based message flows in detail.

The message flows presented and all the companion scripts are provided in source format with the product.

The overall paradigm behind the samples is a fictional weather station that:

- Collects Celsius temperature readings from several cities, in file and message format
- Submits the collected data as a file or sequence of messages to the broker
- The broker applies some syntactic format transformation and semantic conversion to Fahrenheit, then exposes the output data as file or sequence of messages.

Each sample can be run using a script via the command line that assists in the creation of input WebSphere MQ messages or files and the subsequent inspection of the output messages or file.

Preparing to execute the samples

In order to run the samples you must:

- Ensure that **WebSphere MQ samples** have been correctly installed on the target broker machine
- Ensure that WebSphere Message Broker File Extender has been successfully installed and configured on the target broker machine
- Ensure that the *mbfecfgqmgr.sh* command has been successfully executed against the selected broker **without** specifying the –nosamples option (which inhibits the creation of WebSphere MQ queues used by the samples).
- Deploy the provided broker archive file named <mqbfeopt_root>/bin/mbfes.bar to the selected broker using WebSphere Business Integration Message Broker toolkit or WebSphere Business Integration Message Broker *mqsideploy* command.

The samples get their input from:

- WebSphere MQ queue named COM.IBM.BROKER.MBFE.SAMPLE.INPUT.QUEUE, or
- File system directory <mbfevar_root>/filedata/inbox

while the output is produced onto:

- WebSphere MQ queue named COM.IBM.BROKER.MBFE.SAMPLE.OUTPUT.QUEUE, or
- File system directory <mbfevar_root>/filedata/outbox

Sample files

Location: <mbfeopt_root>/bin	
FileName	**Description**
mbfesf2f.esql mbfesf2f.msgflow	F2F message flow source files that can be imported into a WebSphere Business Integration Message Broker toolkit workspace for viewing
mbfesf2f.bat (windows) mbfesf2f.sh (unix)	Shell script assisting in the creation of an input file for the F2F sample and the inspection of the output file created.
mbfesf2m.esql mbfesf2m.msgflow	F2M message flow source files that can be imported into a WebSphere Business Integration Message Broker toolkit workspace for viewing
mbfesf2f.bat (windows) mbfesf2f.sh (unix)	Shell script assisting in the creation of an input file for the F2M sample and the inspection of the output WebSphere MQ messages.
mbfesm2f.esql mbfesm2f.msgflow	M2F message flow source files that can be imported into a WebSphere Business Integration Message Broker toolkit workspace for viewing
mbfesm2f.bat (windows) mbfesm2f.sh (unix)	Shell script assisting in the creation of sequence of input WebSphere MQ messages for the M2F sample and the inspection of the output file created.
mbfes.bar	Bar file including all the sample flows ready for deployment on a target broker.

Sample 1 – File to File message flow

This sample shows how to process an input text file that contains several records consisting of two “$” separated variable length fields:

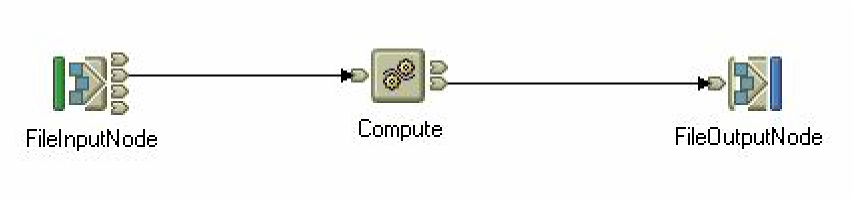
- City
- Temperature in Celsius

and transform it into a different output file of fixed positional records having the following fields:

- Temperature in Fahrenheit

- City
- Timestamp

Below is a picture of the very simple resulting message flow



An interesting feature demonstrated by the Compute node ESQL code is the possibility of dynamically deciding the name of the output file by modifying the *MBFEProperties* file descriptor created by FileInput:

```
SET OutputLocalEnvironment.Variables.MBFEProperties.FileName
    = OUT_NAME;
```

For more information about available file descriptor attributes see ”Message Tree Structures” section.

To actually try out the sample message flow, run <mbfeopt_root>/bin/mbfesf2f.sh (on unix) or <mbfeopt_root>\bin\mbfesf2f.bat (on Windows), then follow the interactive instructions and wait until the output file is displayed.

Below is an example of a typical session transcript:

```
C:\Program Files\IBM\MBFE\bin>mbfesf2f
-------------------------------------------------------------------------

start data acquisition
-------------------------------------------------------------------------

type the name of the city (press enter to finish)
Pisa
type the temperature of the city (press enter to finish)
27

type the name of the city (press enter to finish)
Toronto
type the temperature of the city (press enter to finish)
15

type the name of the city (press enter to finish)
Kuala Lumpur
type the temperature of the city (press enter to finish)
25

type the name of the city (press ctrl-d or enter to finish)

-----------------------------------------------------------------

end data acquisition
-----------------------------------------------------------------
```

```
input file mbfesf2f.Input.Celsius.txt successfully created

-----------------------------------------------------------------
please wait, processing in progress
-----------------------------------------------------------------

output file mbfesf2f.Output.Fahrenheit.txt successfully created


80.6        Pisa           Fri May 06 15:36:00 CEST 2005
59          Toronto        Fri May 06 15:36:00 CEST 2005
77          Kuala Lumpur   Fri May 06 15:36:00 CEST 2005

-----------------------------------------------------------------
test successfully completed
-----------------------------------------------------------------
```

The output file is kept in the <mbfevar_root>/filedata/outbox directory until the next run of this sample.

Sample 2 – File to Message message flow

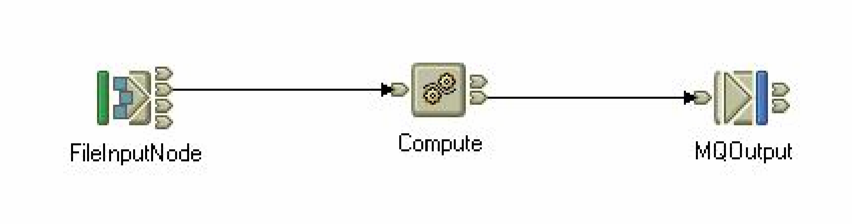
This sample shows how to process an input text file that contains several records consisting of two "$" separated variable length fields:

- City
- Temperature in Celsius

and transform it in a sequence of distinct WebSphere MQ messages (one for each record) in fixed positional format with the following fields:

- Temperature in Fahrenheit
- City
- Timestamp

Below is a picture of the very simple resulting message flow



An interesting feature demonstrated by the Compute node ESQL code is the dynamic creation of the WebSphere MQ message descriptor (*MQMD*) that will not be present in the message tree of non-WebSphere MQ sources of data such as a file.

The relevant snippet of code is:

```
SET OutputRoot.Properties.ReplyIdentifier=MQCI_NONE;
SET OutputRoot.MQMD.Version=MQMD_CURRENT_VERSION;
SET OutputRoot.MQMD.StrucID=MQMD_STRUC_ID;
```

To actually execute the sample message flow, run <mbfeopt_root>/bin/mbfesf2m.sh (on unix) or <mbfeopt_root>\bin\mbfesf2m.bat (on Windows), then follow the interactive instructions and wait until the output file is displayed.

Below is an example of a typical session transcript:

```
C:\Program Files\IBM\MBFE\bin>mbfesf2m.bat MQSI
------------------------------------------------------------------
start data acquisition
------------------------------------------------------------------

type the name of the city (press enter to finish)
Turin
type the temperature of the city (press enter to finish)
24

type the name of the city (press ctrl-d or enter to finish)
Detroit
type the temperature of the city (press enter to finish)
21

type the name of the city (press enter to finish)
Tokyo
type the temperature of the city (press enter to finish)
25

type the name of the city (press enter to finish)

------------------------------------------------------------------

end data acquisition
------------------------------------------------------------------

input file mbfesf2m.Input.Celsius.txt successfully created

------------------------------------------------------------------
please wait, processing in progress
------------------------------------------------------------------

Sample AMQSGET0 start
message <75.2       Turin         Fri May 06 15:56:04 CEST 2005>
message <69.8       Detroit       Fri May 06 15:56:04 CEST 2005>
message <77         Tokyo         Fri May 06 15:56:04 CEST 2005>
no more messages
Sample AMQSGET0 end
------------------------------------------------------------------
test completed
------------------------------------------------------------------
```

The output messages are read with a destructive get, so the queue COM.IBM.BROKER.MBFE.SAMPLE.OUTPUT.QUEUE will be empty after successfully running the sample.

Sample 3 – Message to File message flow

This sample shows how to process a sequence of input WebSphere MQ messages each of which contains two "$" separated variable length fields:

- City
- Temperature in Celsius

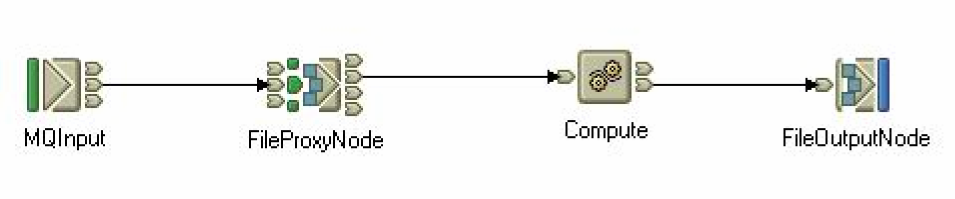
Each message is transformed in a file record having this format having the following fields:

- Temperature in Fahrenheit
- City
- Timestamp

The records are aggregated into output files, a new file is created when *any* of the conditions below is true:

- No new messages are read from for at least 5 seconds (as specified in the *File Close Timeout* attribute of the FileProxy node)
- At least 50 records have been already appended to the currently open file (as specified in the *File Record Threshold* attribute of the FileProxy node)
- The City field read from the current input WebSphere MQ message has the special value "Finisterre"

Below is a picture of the very simple resulting message flow



An interesting feature demonstrated by the Compute node ESQL code is the dynamic file closure, driven by the detection of a special value in the current input message.

The relevant snippet of code is:

```
IF UPPER(CITY) = 'FINISTERRE' THEN
     SET OutputLocalEnvironment.Variables.MBFEProperties.action
         = 'skipAndClose';
END IF;
```

In this case the message containing the Finisterre temperature reading is discarded just before the file closure. Please notice that an alternative value for

MBFEProperties.action tag is *close*, which would instead have caused the Finisterre record to be appended to the file just before the file closure.

To actually execute the sample message flow, run <mbfeopt_root>\bin\mbfesm2f.sh (on unix) or <mbfeopt_root>/bin/mbfesm2f.bat (on Windows), then follow the interactive instructions and wait until the output file is displayed.

Below is an example of a typical session transcript:

```
C:\Program Files\IBM\MBFE\bin>mbfesm2f.bat MQSI
-------------------------------------------------------------------------

start data acquisition
-------------------------------------------------------------------------

type the name of the city (press ctrl-d or enter to finish)
Genoa
type the temperature of the city (press enter to finish)
24

Sample AMQSPUT0 start
target queue is COM.IBM.BROKER.MBFE.SAMPLE.INPUT.QUEUE
Sample AMQSPUT0 end

type the name of the city (press enter to finish)
New York
type the temperature of the city (press enter to finish)
21

Sample AMQSPUT0 start
target queue is COM.IBM.BROKER.MBFE.SAMPLE.INPUT.QUEUE
Sample AMQSPUT0 end

type the name of the city (press enter to finish)
Hong Kong
type the temperature of the city (press enter to finish)
23

Sample AMQSPUT0 start
target queue is COM.IBM.BROKER.MBFE.SAMPLE.INPUT.QUEUE
Sample AMQSPUT0 end

type the name of the city (press ctrl-d or enter to finish)

-------------------------------------------------------------------------
end data acquisition
-------------------------------------------------------------------------

data successfully inserted

-------------------------------------------------------------------------
please wait, processing in progress
-------------------------------------------------------------------------

output file mbfesm2f.Output.Fahrenheit.txt successfully created

C:\Documents and Settings\All Users\Application
Data\IBM\MBFE\filedata\outbox\mb
fesm2f.Output.Fahrenheit.txt


75.2         Genoa          GMTTIME '14:23:38.310' DATE '2005-05-06'
```

```
69.8         New York       GMTTIME '14:23:39.100' DATE '2005-05-06'
73.4         Hong Kong      GMTTIME '14:23:40.030' DATE '2005-05-06'

-------------------------------------------------------------------------
test successfully completed
-------------------------------------------------------------------------
```

The input messages are read with a destructive get, so the queue COM.IBM.BROKER.MBFE.SAMPLE.INPUT.QUEUE will be empty after successfully running the sample, while the output files are kept in the <mbfevar_root>/filedata/outbox directory until the next run of this sample.

Chapter 5. Diagnosis

Installation problems

In the case of installation problems, please review the installation log file `mbfeinstlog.txt` created in the system TEMP directory on Windows and `/tmp`. on UNIX. At the end of the installation this file is copied to <mbfeopt_root>\mdv directory for future reference.

If the problem persists, or if you suspect that the problem lies with the installation program itself, you can try to rerun the installation program with the additional command line parameter `–is:log ishield.log` and then inspect the `ishield.log` file produced.

For other advanced installation command line switches and help with the installer error message, check the InstallShield™ Information Center at this website:
http://helpnet.installshield.com

WebSphere Message Broker File Extender installation kits have been prepared using InstallShield 10.5 Universal Edition (with a version 5 style project).

First use problems

If you are experiencing first use problems, please review carefully all the information provided in the "Configuring for first use" section.

In particular, for Unix platforms, ensure that the environment profile information provided in the <mbfeopt_root>/profiles profile file is available in the broker runtime environment.

Generic runtime problems

To diagnose any other generic runtime problem you experience with WebSphere Message Broker File Extender, you can start the service trace by editing the relevant entries in the *mbfe.properties* configuration file.

Please remember that in this version the change to *mbfe.properties* configuration parameters, including trace settings, will be picked up by the broker only upon restart.

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before you submit your problem to IBM Software Support, ensure that your company has an active IBM software maintenance contract, and that you are authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus, and Rational products, as well as DB2 and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage in one of the following ways:
  - **Online**: Go to the Passport Advantage Web site at http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home, and click **How to Enroll**.
  - **By phone**: For the phone number to call in your country, go to the "Contacts" page of the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/contacts.html, and click the name of your geographic region.
- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at http://www.techsupport.services.ibm.com/ssr/login.
- For customers with IBMLink, CATIA, Linux, S/390, iSeries, pSeries, zSeries, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the "Contacts" page of the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

- Determine the business impact of your problem.
- Describe your problem and gather background information.
- Submit your problem to IBM Software Support.

Instruction to authors: If your product provides the IBM Support Assistant, adjust these steps accordingly (users should submit problems through the Support Assistant whenever possible).

Determine the business impact of your problem

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1	The problem has a *critical* business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
Severity 2	This problem has a *significant* business impact: The program is usable, but it is severely limited.
Severity 3	The problem has *some* business impact: The program is usable, but less significant features (not critical to operations) are unavailable.
Severity 4	The problem has *minimal* business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describe your problem and gather background information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

General information

- Hardware platform
- Operating system name and level
- WebSphere Message broker version and fixpack level (**mqsiservice –v** command)
- WebSphere Message Broker File Extender version identification file located in **<mbfeopt_root>/bin/mbfe_version.dat**.
- Contents of **<mbfeopt_root>/mdv** directory (when present)
- Product installation log (**/tmp/mbfeinstlog.txt** on Unix and **%temp%/mbfeinstlog.txt** on Windows – before installation completion, and **<mbfeopt_root>/mdv/mbfeinstlog.txt** after installation completion.)
- Do you have logs, and messages that are related to the problem symptoms (such as error log entries in Unix syslog or Windows Event log)? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps do you perform to re-create the problem?

- WebSphere Message Broker File Extender service trace files created while executing the use cases that fail
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, or other system components?
- Are you currently using a workaround for the problem? If so, please be prepared to describe the workaround when you report the problem.

Information for first use related problems:

1. Results of the **mbfecfgqmgr.bat/mbfecfgqmgr.sh** command execution
2. Results of the **mbfeivp.bat/mbfeivp.sh** command execution (this is the installation verification procedure; please check the product manual for instructions on how to prepare the environment for it)
3. Results of running samples (please check the product manual for instructions on how to prepare the environment for them)

Information for problems related to the use of WebSphere Message Broker File Extender within message flows:

1. Short description of the intended function of the message flow
2. Complete export of the relevant message flow artifacts needed in order to reproduce the problem (for complex scenarios it is recommended to create a minimal message flow containing just what is relevant for WebSphere Message Broker File Extender troubleshooting, e.g. leaving out all database related subflows and custom plug-in nodes if any). If the export is not feasible, provide at least screenshots of file node configuration tabs.
3. Description of input and output data (file and/or messages), providing non-confidential data samples if available
4. Step by step instructions on how to reproduce the problem using the artifacts and information provided above.

Submit your problem

You can submit your problem to IBM Software Support in one of two ways:

- Online: Go to the Submit and track problems tab on the IBM Software Support site at http://www.ibm.com/software/support/probsub.html. Type your information into the appropriate problem submission tool.
- By phone: For the phone number to call in your country, go to the "Contacts" page of the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the

Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Chapter 6. Error Messages and error codes

Informational, Warning and Error messages are written by WebSphere Message Broker File Extender using the standard platform-specific system log (for example, EventLog on Windows).

Below is a table of all the diagnostic messages.

Message Code	Message text	User Action
MBFE_00001	Error during File Extender initialization.	Potential initial configuration problem. Ensure that the mbfecfgqgr command has been successfully run against the problem broker. For unix platforms, also ensure that the relevant environment profile file (see <mbfeopt_root>/profiles) is available in the broker runtime environment.
MBFE_00002	Error clearing MbMessage.	Product internal error. Contact IBM support.
MBFE_00003	Error accessing file transfer layer. Server is not available.	Potential WebSphere MQ problem. Ensure that the broker queue manager is running without problems.
MBFE_00004	Exception caught during execution "{0}"	Unexpected condition detected during processing. Review additional information in {0} to determine the cause of the problem.
MBFE_00005	Internal error occurred during execution "{0}"	Product internal error. Contact IBM support.
MBFE_00006	Error communicating with File Extender file	Potential initial configuration problem.

	management layer "{0}".	Ensure that the mbfecfgqgr command has been successfully run against the problem broker. For unix platforms, also ensure that the relevant environment profile file (see <mbfeopt_root>/profiles) is available in the broker runtime environment.
MBFE_00007	Found a restarted session with ID "{0}" connecting with File Extender file management layer. Node key "{1}".	This is an information message. A file management session in restartable status has been detected and correctly restarted.
MBFE_00008	No work pending for current restarted session. Starting a new one.	This is an information message. A file management session in restartable status has been detected and correctly closed.
MBFE_00009	Missing original input data for restarted session.	Product internal error. Contact IBM support.
MBFE_00010	Recovering (REDO) operation for current file.	This is an information message. An in doubt session has been automatically resolved in REDO mode as specified in the nodes configuration file.
MBFE_00011	Recovering (SKIP) operation for current file.	This is an information message. An in doubt session has been automatically resolved in SKIP mode as specified in the nodes configuration file.
MBFE_00012	Recovering (SKIP) operation for current file.	This is an information message. An in doubt session is about to be resolved in SKIP mode as specified in the nodes

		configuration file.
MBFE_00013	Error managing troubleshooting "{0}"	Check that the node *Service String* parameter has not been accidentally filled.
MBFE_00014	Internal error occurred during execution "{0}"	Product internal error. Contact IBM support.
MBFE_00015	JMSException detected in evaluate method "{0}".	Unexpected condition detected during processing. Review additional information in {0} to determine the cause of the problem.
MBFE_00016	Unexpected exception detected in evaluate method "{0}".	Unexpected condition detected during processing. Review additional information in {0} to determine the cause of the problem.
MBFE_00017	Node "{0}" - Unexpected exception detected in evaluate method: "{1}".	Unexpected condition detected during processing. Review additional information in {0} and {1} to determine the cause of the problem.
MBFE_00018	Error deleting the node "{0}": "{1}"	Unexpected error during clean up phase; this problem can be transient. Review additional information in {0} and {1} to determine the cause of the problem.
MBFE_00019	Cannot create terminal "{0}"	Product internal error. Contact IBM support.
MBFE_00020	Exception caught during execution "{0}"	Unexpected condition detected during processing. Review additional information in {0} to determine the cause of the problem.
MBFE_00021	Error trying to initialize ProxyNode: "{0}"	Unexpected condition detected during processing.

		Review additional information in {0} and the FileProxy node configuration/placement within the message flow to determine the cause of the problem.
MBFE_00022	Unknown input terminal "{0}"	Product internal error. Contact IBM support.
MBFE_00023	Error performing checkpoint "{0}"	Unexpected condition detected during processing. Review additional information in {0} to determine the cause of the problem.
MBFE_00024	The terminal "{0}" has to be attached if the transaction is required.	A FileProxy node was configured as transactional but the synch terminal was not attached to an MQReply node. Review your message flow design accordingly.
MBFE_00025	The corresponding terminal of "{0}" must be attached.	A FileProxy node shows an incorrect placement in a message flow. Review your message flow design accordingly.
MBFE_00026	Error trying to add session properties to incoming message "{0}"	Product internal error. Contact IBM support.
MBFE_00027	Error propagating to the 'out' terminal "{0}".	Unexpected condition detected during processing. Review additional information in {0} and FileProxy node configuration/placement within the message flow to determine the cause of the problem.
MBFE_00028	Error propagating to the 'catch' terminal "{0}".	Unexpected condition detected during processing.

		Review additional information in {0} and FileProxy node configuration/placement within the message flow to determine the cause of the problem.
MBFE_00029	Error propagating to the 'fail' terminal ''{0}''.	Unexpected condition detected during processing. Review additional information in {0} and FileProxy node configuration/placement within the message flow to determine the cause of the problem.
MBFE_00030	The previous session is in doubt and will be deleted because the property inDoutbPolicy is set to 'CLEAR'. Node ''{0}''.	This is an information message. An in doubt session is about t be automatically resolved in CLEAR mode as specified in the nodes configuration file.
MBFE_00031	The previous session is in doubt but will be continued because the property inDoutbPolicy is set to 'CONTINUE'. Node ''{0}''.	This is an information message. An in doubt session is about t be automatically resolved in CONTINUE mode as specified in the nodes configuration file.
MBFE_00032	Invalid data received on input terminal 'from catch''. Check FileProxy placement within the message flow and FIleProxy terminal connections.	Unexpected condition detected during processing. Review additional error messages and FileProxy node configuration/placement within the message flow to determine the cause of the problem.
MBFE_00033	Internal error in node execution ''{0}''.	Product internal error. Contact IBM support.
MBFE_00034	Error preparing to commit ''{0}''.	Unexpected condition detected during processing. Review additional

		information in {0} to determine the cause of the problem.
MBFE_00035	Error preparing to checkpoint "{0}".	Unexpected condition detected during processing. Review additional information in {0} to determine the cause of the problem.
MBFE_00036	Error performing commit "{0}".	Unexpected condition detected during processing. Review additional information in {0} to determine the cause of the problem.
MBFE_00038	File Extender PATH configuration file not found. The file "{0}" is contained in mbfe.path.jar file under the broker jplugin directory.	Potential installation error, review the contents of the broker *jplugin* directory as indicated in the error text.
MBFE_00039	The property "{0}" has not been found in configuration file "{1}".	Problem potentially caused by an accidental deletion of some lines in {1} configuration file. Restore a backup of the last known working {1} or manually retype just the relevant {0} entry.
MBFE_00040	Internal exception trying to restart session "{0}" with node key "{1}": "{2}".	Unexpected condition detected during processing. Review additional information in {0}, {1} and {2} to determine the cause of the problem.
MBFE_00041	Session "{0}" reached max retry count. Node key is "{1}".	This is a warning message. A transient error in a session has not been recovered within the configured retry interval.
MBFE_00042	Invalid file received reading from path, "{0}". File ID is "{1}". Node key is "{2}".	Potential configuration problem or bad directory contents detected while reading files from path {0}

		Review FileInput configuration and input path content.
MBFE_00043	Blocking error condition detected on session "{0}", the session will be frozen. Use transaction utilities to recover data. Node key is "{1}".	The session {0} contains a poison file. Investigate the cause of the problem and use the *mbfesolve* command to perform the most appropriate recovery action.
MBFE_01001	Invalid input path "{0}" for file input node. Not present on file system.	Configuration problem for FileInput, an invalid input path {0} has been specified. Check if {0} exists, is a directory and is accessible to the broker user.
MBFE_01002	Invalid input path "{0}" for file input node. It is not a directory.	Configuration problem for FileInput, an invalid input path {0} has been specified. Check if {0} exists, is a directory and is accessible to the broker user.
MBFE_01003	Invalid input path "{0}" for file input node. Wrong access rights.	Configuration problem for FileInput, an invalid input path {0} has been specified. Check if {0} exists, is a directory and is accessible to the broker user.
MBFE_01004	Managing in doubt session with in doubt policy set to FAIL. Node Key: "{0}", session ID "{1}", - File ID "{2}", file name "{3}", directory "{4}".;	This is an information message. An in doubt session is about to be automatically resolved in FAIL mode as specified in the nodes configuration file.
MBFE_01005	Empty destination list	This is a warning message. A FileOutput node was configured to use a destination list but the dynamic destination list detected at runtime was empty.

MBFE_01006	The node "{0}" read an invalid destination from LocalEnvironment - "{1}"	A FileOutput node was configured to use a destination list but the dynamic destination list detected at runtime contained an invalid destination directory. Review your message flow design and/or ESQL code to correct the problem.
MBFE_02001	Message generated by mbfe feeder reading from a directory	The product component for reading input directories has detected an unexpected condition. Review chained error message to investigate the cause of the problem.

Chapter 7. Related information

You can find documentation on WebSphere Business Integration Message Broker at the following website:

http://www.ibm.com/software/integration/wbimessagebroker/library/

You can find documentation on WebSphere MQ at the following website:

http://www.ibm.com/software/integration/wmq/library/

You can search the IBM Support technotes database with keywords “file extender” at the following website:

http://www.ibm.com/support/us/

You can download WebSphere Message Broker File Extender interim fixes and fixpacks from the following FTP site:

ftp://ftp.software.ibm.com/software/mqseries/fixes/wbimbfev50/

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. 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. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information 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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM’s suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
AIX 5L
WebSphere
WebSphere MQ
xSeries

SPAZIO is a trademark of Primeur Group in Italy, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

InstallShield is a trademark of InstallShield Software Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Program number: 5724-M01
Form number: GC34-6666-01