



**VSAM Nodes  
SupportPac IA13**





**VSAM Nodes  
SupportPac IA13**

**Note**

Before using this information and the product it supports, read the information in the Notices appendix.

**Second Edition (December 2005)**

This edition applies to WebSphere Message Broker Version 6 and to all subsequent releases and modifications, unless otherwise indicated in new editions. This supersedes the first edition of SupportPac IA13 which applied only to WebSphere Business Integration Message Broker Version 5.

© Copyright International Business Machines Corporation 2004, 2005. All rights reserved.

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

---

## Contents

Introduction . . . . .	1
VSAM basics . . . . .	3
VSAMInput node . . . . .	7
VSAMRead node . . . . .	19
VSAMWrite node . . . . .	29
VSAMUpdate node . . . . .	37
VSAMDelete node . . . . .	47
Using the VSAM nodes in a message flow . . . . .	55
Sample 1 - Using a VSAMInput node . . . . .	57
Sample 2 - Using a VSAMWrite node . . . . .	61
Sample 3 - Using a VSAMRead node . . . . .	63
Sample 4 - Using a VSAMUpdate node . . . . .	67
Sample 5 - Using a VSAMDelete node . . . . .	71
Error messages and error codes . . . . .	75
Dependencies. . . . .	83
Related information . . . . .	85
Appendix A. Notices . . . . .	87
Appendix B. Trademarks . . . . .	89



---

## Introduction

SupportPac™ IA13 provides five built-in message-processing nodes that you can use in a message flow to read, write, delete, and update records in VSAM (Virtual Storage Access Method) data sets.

**Note:** The VSAM nodes are only available for message flows that run on z/OS® platforms.

The names of the five nodes are **VSAMInput**, **VSAMRead**, **VSAMWrite**, **VSAMUpdate**, and **VSAMDelete**.

Use the **VSAMInput** node to read a specified number of records from a VSAM data set and propagate each record to subsequent nodes in your message flow. You can configure the node to read either every record in the data set, or a specified number of records from a part of the data set.

Use the **VSAMRead** node to read a record from a VSAM data set and propagate the record to subsequent nodes in your message flow.

Use the **VSAMWrite** node to write a record to a VSAM data set using data that is provided in an input request message.

Use the **VSAMUpdate** node to update a record in a VSAM data set using data that is provided in an input request message.

Use the **VSAMDelete** node to delete a record from a VSAM data set.

**Restriction:** This SupportPac can be used only with Version 5.0, Fix Pack 3, and later versions of WebSphere® Business Integration Message Broker, or WebSphere Message Broker, Version 6.





---

## VSAM basics

Virtual Storage Access Method (VSAM) is an access method that is used to process data on large-scale IBM operating systems such as z/OS.

### Accessing VSAM data sets

You can access VSAM data sets either by direct access or sequentially:

By direct access	Search arguments are used to determine which records in the data set are processed. Successive search arguments do not need to be related to each other.
Sequentially	Records are processed in sequence, starting from any record in the data set. The entire data set or part of the data set can be processed.

### Types of VSAM data sets

There are three types of VSAM data sets:

- A **key-sequenced data set (KSDS)**. Records in this type of data set are arranged in ascending order of the key value. A record is accessed using a key that is located within each record. Each record in a KSDS must have a unique key value. Records can be read, written, deleted, or updated.
- An **entry-sequenced data set (ESDS)**. Records in this type of data set are arranged in the order that they are created in the data set. Access to the records is by direct access using a relative byte address (RBA). Records can be read, written, or updated, but cannot be deleted. Updated records must keep the same record length.
- A **relative record data set (RRDS)**. This type of data set consists of preformatted fixed-length slots, some of which might be empty. Access to the records is by direct access using a relative record number (RRN). Records can be read, written, deleted, or updated.

For KSDS and ESDS, in addition to the primary VSAM access described above, records can be accessed using one or more additional key fields that are located within each record. These additional keys are called an alternate index. The values of these additional keys do not have to be unique. Data sets that contain records that contain an alternate index are called **KSDS PATH**, and **ESDS PATH**, respectively.

After a record has been accessed by direct access, in any of these data set types, access to other records in the data set can be sequentially.

### Accessing a record within a VSAM data set

All records within a VSAM data set can be accessed using one of the following:

- |            |   |
|------------|---|
| <b>Key</b> | This is a field within each record of the data set that uniquely identifies the record.   |
| <b>RBA</b> | The RBA of a record is the offset of its first byte from the start of the data set. The RBA of the first record in a data set is 0. |

**RRN** The RRN of a record is the number of the record within the data set. The RRN of the first record in a data set is 1.

## Key operations

The Position Mode property of a VSAM node is used to determine which record or records are accessed, and can have one of the following values:

### KEY\_FIRST

Access is to the first record in the data set.

### KEY\_LAST

Access is to the last record in the data set.

### KEY\_EQ

Access is to the first record with a key or RRN that is equal to the value specified in the KEY or RRN property.

### KEY\_GE

Access is to the first record with a key or RRN that is equal to, or greater than, the value specified in the KEY or RRN property.

### KEY\_EQ\_BWD

Access is to the first record with a key or RRN that is equal to the value specified in the KEY or RRN property when the VSAM data set is in 'read backwards' mode.

**Note:** A full key must be provided when this mode is used. Partial keys are not supported with KEY\_EQ\_BWD.

### RBA\_EQ

Access is to the first record with an RBA that is equal to the value specified in the RBA property.

### RBA\_EQ\_BWD

Access is to the first record with an RBA that is equal to the value specified in the RBA property when the VSAM data set is in 'read backwards' mode.

### CURRENT

Access is to a record whose position is determined by the position of the record that has been processed by a VSAM node that is upstream from this node in the message flow, and that has the Hold Position property set to Downstream.

If the Position Mode property is set to CURRENT in a **VSAMRead** node, the next sequential record in the data set is accessed. This record is then held for subsequent nodes.

If the Position Mode property is set to CURRENT in a **VSAMUpdate** node or a **VSAMDelete** node, the held record is updated or deleted, respectively.

The following table shows the effect that each value of the Position Mode property has on each type of VSAM data set.

Position Mode	KSDS	ESDS	RRDS	KSDS_PATH ESDS_PATH
KEY_FIRST	Supported by all data set types			
KEY_LAST				

Position Mode	KSDS	ESDS	RRDS	KSDS_PATH ESDS_PATH
KEY_EQ KEY_GE KEY_EQ_BWD	By Key	Not applicable	By RRN	By Key
RBA_EQ RBA_EQ_BWD	By RBA		Not supported	
CURRENT	Supported by all data set types			

## Transactionality

The VSAM operations that are performed by these nodes do not participate in the transactions that exist within the message flow. Successful VSAM operations are not automatically backed out if a failure occurs in the message flow.

When you consider the effect of using VSAM SHAREOPTIONS on a VSAM data set, note that an execution group rather than a message flow should be regarded as the equivalent of a user task or application program.

**Note:** The services that are used by the VSAM nodes to access VSAM data sets do not use SHAREOPTIONS to enforce which intents the data set can be opened with. Therefore, any VSAM data set can be opened for read or update intent by the VSAM nodes, regardless of the setting of SHAREOPTIONS.



---

## VSAMInput node

Use a VSAMInput node to read batches of records from a VSAM data set, and propagate each of these records to the rest of a message flow.

To do this, send a message to the *Control* terminal of a VSAMInput node. Include a VSAM Request in this message if you want to override the settings of properties that are configured in the VSAMInput node.

When a batch has been processed, a VSAM Status message that summarizes the outcome of the request is propagated from the Status terminal.

A VSAM Report message can also be sent, as part of each output message, to the next node in the message flow.

The VSAM read operations that are performed by this node are non-transactional. However, if the message flow interacts with a database or other external resource (for example, a WebSphere MQ queue), these interactions are performed as part of a transaction. The processing of batches within a VSAMInput node is always non-persistent.

The VSAMInput node provides commit or rollback, depending on how the message flow ends, and how it is configured for error-handling.

A VSAMInput node is represented in the workbench by the following icon:



---

## Configuring a VSAMInput node

When you have put an instance of the VSAMInput node into a message flow, you can configure it.

**Note:** Values that you specify for properties in the node are superseded by any corresponding fields that are included in a VSAM Request that is part of the input message. All supplied values are validated even when they might not be used.

To configure the node, do the following:

Right-click the node in the editor view of the message flow and click **Properties**. The **Basic** properties of the node are displayed in the properties dialog.

Configure the **Basic** properties of a VSAMInput node by doing the following:

- Specify the *Default File Name*. This should be the fully resolved name of the VSAM data set that you want to perform operations on. The format of the file name must be `//'dataset name'`.

Configure the **Advanced** properties of a VSAMInput node by selecting **Advanced** in the properties dialog navigator, and then doing the following:

- If you want the VSAM data set to remain open after an operation on the data set, select *Until Idle* in the *Retain File* property. The data set remains open until the idle time period expires, unless the data set is explicitly closed before this. If you select *No* for this property, the data set is closed when the current operation completes.
- Select a value for the *Hold Position* property to determine whether the data set that is opened by this node should remain positioned at the record that is read by this node for subsequent nodes in the message flow. Valid values are:
  - *Downstream*. The data set remains positioned at the record that is read by this node for nodes downstream from the Out terminal of the node.  
To make use of this positioning, select *CURRENT* for the *Position Mode* property of subsequent nodes in the message flow.

**Note:** Any record that is read by a *VSAMRead* node that is downstream from this *VSAMInput* node in the message flow, and that has *CURRENT* specified for its *Position Mode* property, is not read by the *VSAMInput* node.

  - *No*. Positioning is not held on the record that is read by this node.
- If you want data sets that are opened by this node to be updated by subsequent VSAM nodes, select *Update* in the *Additional Open Intent* property. If you select *None* for this property, any data sets opened by this node are opened for read intent only.
- If you want less than whole records to be read, specify in the *Maximum Record Length* property the maximum number of bytes that can be read for each record. If you do not specify a value, whole records are read.
- Specify in the *Commit Count* property the number of records to be read within each unit of work.
- If you want a failure to cause the processing of a batch of records to stop immediately, select *End Request* for the *Failure Action* property. If you select *Backout Full Batch* for this property, the remaining records in the batch are processed before the whole batch is backed out. However, the remaining records for any request are then processed.

Configure the **Default** properties of a *VSAMInput* node by selecting **Default** in the properties dialog navigator, and then doing the following:

- For the *Domain* property, enter the name of the parser that you are using to parse the record. Valid options include:
  - *MRM*
  - *XML*
  - *XMLNS*
  - *XMLNSC*
  - *JMSMap*
  - *JMSStream*
  - *BLOB*
- If you are using the *MRM* parser, enter the message set in the *Message Set* property.  
Leave the *Message Set* field blank for *XML*, *XMLNS*, *XMLNSC*, *JMS*, and *BLOB* parsers.
- If you are using the *MRM* parser, enter the message type in the *Message Type* property.  
Leave the *Message Type* property blank for *XML*, *XMLNS*, *XMLNSC*, *JMS*, and *BLOB* parsers.

- Enter the format of the message in the *Message Format* property.  
Leave the *Message Format* property blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- Enter in the *CCSID* property the code page to use when you parse the message.
- Enter in the *Encoding* property the encoding to use when you parse the message.

Configure the **Request** properties of a VSAMInput node by selecting **Request** in the properties dialog navigator, and then doing the following:

- Specify in the *Request Location* property the ESQL location of the request message. If you do not specify a value, the default location `InputBody.VSAM.Request` is used.
- Select from the options for the *Position Mode* property to determine how to position at the first record that is to be processed. The default, `KEY_FIRST`, positions at the first record in the VSAM data set.
- Specify in the *Key* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the first record within the batch of records that is to be propagated. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- Specify in the *End Key* property a value for the highest key; records with a key that is higher than this value are not propagated. This property is ignored if it is blank.
- The *Key Type* property describes how the key is interpreted. Select Parser String in the *Key Type* property if you want the key to be interpreted using the code page that is specified in the *CCSID* property. Otherwise, with the default Hexadecimal set, the key is interpreted as a hexadecimal string.
- Specify in the *RBA* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the first record within the batch of records that is to be propagated. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- Specify in the *End RBA* property a value for the highest RBA; records with an RBA that is higher than this value are not propagated. This property is ignored if it is blank.
- Specify in the *RRN* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the first record within the batch of records that is to be propagated. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- Specify in the *End RRN* property a value for the highest RRN; records with an RRN that is higher than this value are not propagated. This property is ignored if it is blank.
- If more than one of the properties *End Key*, *End RBA*, and *End RRN*, are specified, a record is not propagated whose *Key*, *RBA*, or *RRN* value, respectively, is higher. That is, a record is propagated only if *Key* is not higher than *End Key*, *RBA* is not higher than *End RBA*, and *RRN* is not higher than *End RRN*.
- Specify in the *Maximum Number of Records* property the maximum number of records that are to be read.

Configure the **Result** properties of a VSAMInput node by selecting **Result** in the properties dialog navigator, and then doing the following:

- Specify in the *Output Data Location* property the ESQL path expression for the location of where the VSAM record data is written.

Configure the **Status** properties of a VSAMInput node by selecting **Status** in the properties dialog navigator, and then doing the following:

- If you want a report to be included in the output message assembly, specify **Yes** in the *Include Record Report* property and an ESQL path expression in the *Report Location* property to describe where in the output message assembly the report is located.
- Specify in the *Status Location* property the ESQL path expression for the location of the status of the request. If you do not specify a value, the status message is placed at `OutputLocalEnvironment.VSAM.Status`.

Select **Description** in the properties dialog navigator if you want to give a short description, a long description, or both.

Click **Apply** to make the changes to the VSAMInput node without closing the properties dialog.

Click **OK** to apply the changes and close the properties dialog.

Click **Cancel** to close the properties dialog and discard all the changes that you have made to the properties of the node.

---

## Terminals of a VSAMInput node

The terminals of a VSAMInput node are described in the following table:

Terminal	Description
Control	The input terminal that receives the VSAM request message.
Out	An output terminal through which a message is routed for each VSAM record read.
Catch	An output terminal through which the original VSAM request message and an exception list are routed if an exception is thrown, and not handled, during propagation from the <i>Out</i> terminal.
Failure	An output terminal through which a failure message is routed if there is a failure within the node.
Status	An output terminal through which status messages are routed.

## Connecting the output terminals

If you want to do further processing within the message flow, connect one or more of the output terminals to other nodes in the message flow. If you do not connect any of the output terminals to another node, no further processing is performed.

If an error occurs within the VSAMInput node, what happens depends on when the error occurred and whether the *Failure* terminal was connected to another node.

An error can occur at one of the following times:

- While processing the VSAM request message; the VSAMInput node has received a message through the *Control* terminal but the message is not a valid request message (for example, it might contain a non-numeric value for a numeric field).



- While performing the VSAM request; the VSAMInput node has received a VSAM request message through the *Control* terminal and is in the process of reading the VSAM data and propagating it to its output terminals. A typical error might be a failure to open the VSAM data set.

The resultant actions are described in the following table

<b>Time of error</b>	<i>Failure terminal connected</i>	<i>Failure terminal not connected (or subsequent error occurs downstream from the Failure terminal)</i>
Processing the control message	The entire input message and an exception list that describes the error are propagated down the <i>Failure</i> terminal.	An exception is thrown upstream of the <i>Control</i> terminal.
Servicing the request.	An exception list that describes the error is propagated down the <i>Failure</i> terminal.	Details of the error are written to syslog.

If you do not connect the *Out* terminal of a node, any information about the successful completion of the processing of the node is discarded.

---

## Properties of a VSAMInput node

The following tables describe the properties of the VSAMInput 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.

The **Basic** properties of the VSAMInput node are described in the following table:

<b>Property</b>	<b>M</b>	<b>C</b>	<b>Default</b>	<b>Description</b>
Default File Name	No	Yes	Blank	The fully qualified name of the VSAM data set of the form <i>// ' data set name'</i> .

The **Advanced** properties of the VSAMInput node are described in the following table:

<b>Property</b>	<b>M</b>	<b>C</b>	<b>Default</b>	<b>Description</b>
Retain File	No	No	'No'	Indicates whether the data set remains open after the initial opening of the data set.  Valid values are: 'No' 'Until Idle'

Property	M	C	Default	Description
Hold Position	No	No	'No Hold'	Specifies whether the data set can be used by another VSAM node later in the message flow that has specified the Position Mode option 'Current'.  Valid values are: 'No Hold' 'Downstream'
Additional Open Intent	No	No	'None'	Allows update intent to be added to the file options that are requested by this node.  Valid values are: 'None' 'Update'
Maximum Record Length	No	No	0	The maximum number of bytes that is read for each record.  If this property is 0 (the default), the whole record is read.
Commit Count	Yes	No	1	Specifies the number of records to be read between commits.
Failure Action	No	No	'Backout Full Batch'	Describes the failure action for any exception that is not handled in the message flow'.  Valid values are: 'End request' 'Backout Full Batch'

The **Default** properties of the VSAMInput node are described in the following table:

Property	M	C	Default	Description
Message Domain	No	No	Blank	The message domain to use when you parse the message.
Message Set	No	No	Blank	The message set to use when you parse the message.
Message Type	No	No	Blank	The message type to use when you parse the message.
Message Format	No	No	Blank	The message format to use when you parse the message.
CCSID	No	No	0	The code page to use when you parse the message.
Encoding	No	No	0	The encoding to use when you parse the message.

The **Request** properties of the VSAMInput node are described in the following table:

Property	M	C	Default	Description
Request Location	No	No	See note below.	Specifies the location of the request message. This property takes the form of an ESQL field reference.
Position Mode	No	No	'KEY_FIRST'	Method of positioning to the record that is read.  Valid values are: 'KEY_FIRST' 'KEY_LAST' 'KEY_EQ' 'KEY_GE' 'KEY_EQ_BWD' 'RBA_EQ' 'RBA_EQ_BWD'
Key	No	No	Blank	Key to be used with key-based position modes. This property is ignored if its value is blank.
End Key	No	No	Blank	The highest key; records with a key higher than this are not propagated. This property is ignored if its value is blank.
Key Type	No	No	'Hexadecimal'	Describes the format of the key.  Valid values are: 'Hexadecimal' 'Parser String'
RBA	No	No	0	RBA to be used with RBA-based position modes. This property is ignored if its value is 0.
End RBA	No	No	0	The highest RBA; records with an RBA higher than this are not propagated. This property is ignored if its value is 0.
RRN	No	No	0	RRN to be used with RRN-based position modes. This property is ignored if its value is 0.
End RRN	No	No	0	The highest RRN; records with an RRN higher than this are not propagated. This property is ignored if its value is 0.
Maximum Number of Records			0	The maximum number of records that can be read from the initial location. If the value is 0, records are read up to the end of file.

**Note:** The default value for the *Request Location* property is `InputBody.VSAM.Request`.

The **Result** property of the VSAMInput node is described in the following table:

Property	M	C	Default	Description
Output Data Location	No	No	Blank	Location in the message assembly where the VSAM record data is written. This property takes the form of an ESQL field reference.

The **Status** properties of the VSAMInput node are described in the following table:

Property	M	C	Default	Description
Include Record Report	No	No	'No'	Whether to include in the output record a report that describes the records read.  Valid values are: 'Yes' 'No'
Report Location	No	No	Blank	Where the report is placed. This property takes the form of an ESQL field reference.
Status Location	No	No	Blank	Where the status for each request is placed. This property takes the form of an ESQL field reference.

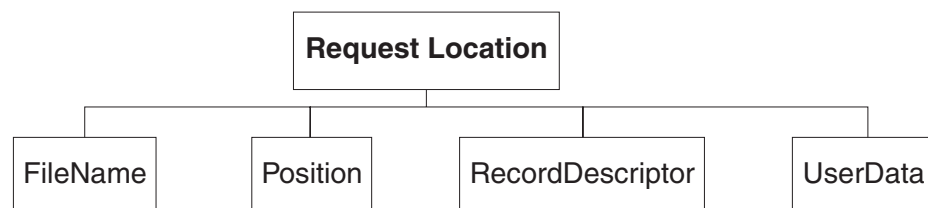
---

## VSAM Request message

The VSAM Request message contains information about the VSAM operation. The location of the VSAM Request message is specified by the Request Location property of the VSAMInput node. All elements of the VSAM Request message are optional.

All values that are specified in the VSAM request message override those that are defined in the corresponding properties of the VSAMInput node. If the tree contains no request at the Request Location, all values are taken from the node properties.

The structure of the VSAM Request message that is used by a VSAMInput node is shown in the following figure:



In the tree structure shown, VSAM Request has the following children:

- **FileName:** The name of the VSAM data set to be used for this request; this overrides the Default File Name property. This element must be supplied with a data type of CHARACTER.
- **Position**  
The fields and data type for each element of the Position subtree are shown in the following table. The last column of the table shows the node property that the field overrides.

Field name	Data type	Node property overridden
Mode	CHARACTER	Position Mode
Key	CHARACTER	Key
KeyType	CHARACTER	Key Type
EndKey	CHARACTER	End Key
RBA	INTEGER or CHARACTER	RBA
EndRBA	INTEGER or CHARACTER	End RBA
RRN	INTEGER or CHARACTER	RRN
EndRRN	INTEGER or CHARACTER	End RRN
MaxNumber	INTEGER or CHARACTER	Maximum Number of Records

- **RecordDescriptor**  
The fields and data type for each element of the RecordDescriptor subtree are shown in the following table. The last column of the table shows the corresponding node property.

Field name	Data type	Node property overridden
Domain	CHARACTER	Message Domain
MessageSet	CHARACTER	Message Set
MessageType	CHARACTER	Message Type
MessageFormat	CHARACTER	Message Format
CCSID	INTEGER or CHARACTER	CCSID
Encoding	INTEGER or CHARACTER	Encoding

- **UserData:** A single element of any data type that is copied to the VSAM Report and VSAM Status messages.

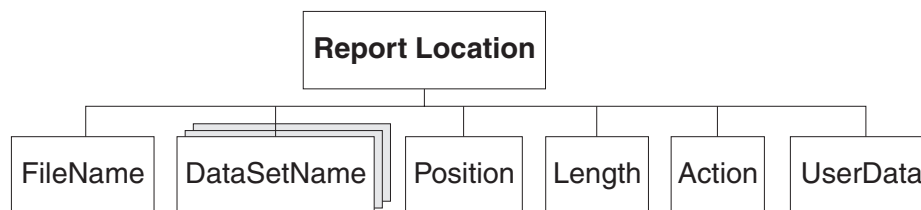
---

## VSAM Report message

A Report message can be part of the output message that is propagated through the Out terminal of the VSAMInput node to another node in the message flow.

If a report message is included as part of the output message of a VSAMInput node, its location is specified in the Report Location property of the VSAMInput node.

The structure of a report message for the VSAMInput node is shown in the following figure:



In the tree structure shown, Report Location has the following children:

- **FileName:** The name of the VSAM data set that is specified for this request. This element is of type CHARACTER.
- **DataSetName:** A repeating element of type CHARACTER that contains the data set qualifiers for the file used by this node.
- **Position**  
The fields and data type for each element of the Position subtree are shown in the following table.

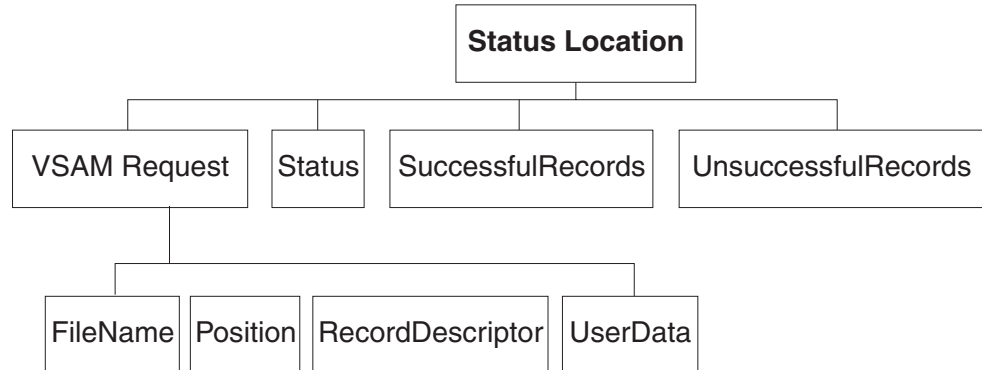
Field name	Data type	Description
Key	CHARACTER	Key of the record that has been read. This element is included following the successful read of a record from a KSDS, KSDS PATH, or ESDS PATH data set.
KeyType	CHARACTER	The value is always 'Hexadecimal'.
RBA	INTEGER	RBA of the record that has been read. This element is included following the successful read of a record from a KSDS or ESDS data set.
RRN	INTEGER	RRN of the record that has been read. This element is included following the successful read of a record from an RRDS data set.

- **Length:** The length of the record that was read by this node. This element is of data type INTEGER.
- **Action:** The VSAM operation performed by this node. This element is of type CHARACTER and, for the VSAMInput node, its value is 'Read'.
- **UserData:** A copy of the UserData element that was specified in the VSAM Request message. The element is only included if a UserData element is present in the VSAM request.

## VSAM Status message

A VSAM status message forms the output message that is propagated through the Status terminal when a VSAM request has been completed.

The structure of a VSAMInput status message is shown below:



In the tree structure shown, Status Location has the following children:

- **FileName:** The name of the VSAM data set that is specified for this request. This element is of type CHARACTER.
- **Position**  
The fields and data type for each element of the Position subtree are shown in the following table.

Field name	Data type	Description
Key	CHARACTER	The value of the Key that was used by this request.
KeyType	CHARACTER	Describes the type of the Key and EndKey elements.
EndKey	CHARACTER	The value of the EndKey that was used by this request.
RBA	INTEGER	The value of the RBA that was used by this request.
EndRBA	INTEGER	The value of the EndRBA that was used by this request.
RRN	INTEGER	The value of the RRN that was used by this request.
EndRRN	INTEGER	The value of the EndRRN that was used by this request.
MaxNumber	INTEGER	The value of MaxNumber that was used by this request.

- **RecordDescriptor**

The fields and data type for each element of the RecordDescriptor subtree are shown in the following table. The last column of the table shows the corresponding node property.

Field name	Data type	Description
Domain	CHARACTER	The Domain that is used for this request.
MessageSet	CHARACTER	The Message Set that is used for this request.
MessageFormat	CHARACTER	The Message Format that is used for this request.
MessageType	CHARACTER	The Message Type that is used for this request.
CCSID	INTEGER	The CCSID that is used for this request.
Encoding	INTEGER	The Encoding that is used for this request.

- **Status:** Details the overall status of the request. The data type is CHARACTER and the element has one of the following values:
  - 'success': All records were processed successfully.
  - 'warning': The request completed, but not all records were processed successfully.
  - 'error': An error occurred and the request was ended; End Request was specified for the Failure Action.
  - 'severe': An error occurred and the node was unable to continue with the request.
- **SuccessfulRecords:** A count of the records that were successfully processed by the request. The data type is INTEGER.
- **UnsuccessfulRecords:** A count of the exceptions that were not handled by the request. The data type is INTEGER.
- **UserData:** A copy of the UserData element that was specified in the VSAM Request message. This element is only present in the status message if a UserData element was present in the VSAM Request message.



---

## VSAMRead node

Use a VSAMRead node to read a record from a VSAM data set, and propagate the record to the rest of a message flow.

To do this, send a message to the *In* terminal of a VSAMRead node. Include a VSAM Request in this message if you want to override the settings of properties that are configured in the VSAMRead node.

A VSAM Report message can also be sent, as part of the output message, to the next node in the message flow.

A VSAMRead node is represented in the workbench by the following icon:



---

## Configuring a VSAMRead node

When you have put an instance of the VSAMRead node into a message flow, you can configure it.

**Note:** Values that you specify for properties in the node are superseded by any corresponding fields that are included in a VSAM Request that is part of the input message. All supplied values are validated even when they might not be used.

To configure the node do the following:

Right-click the node in the editor view of the message flow and click **Properties**. The **Basic** properties of the node are displayed in the properties dialog.

The VSAMRead node has only one Basic property, the *Default File Name* property.

Configure the **Basic** property by doing the following:

Specify in the *Default File Name* property the fully resolved name of the VSAM data set from which you want to read a record. The format of the file name must be `//'datasetname'`.

Configure the **Advanced** properties of a VSAMRead node by selecting **Advanced** in the properties dialog navigator, and then doing the following:

- Specify a value for the *Generate Mode* property to determine the content of the output message assembly.

If you specify *Message*, the output message assembly contains the input *LocalEnvironment* tree and a new message.

If you specify *LocalEnvironment*, the output message assembly contains the new *LocalEnvironment* tree and the input message tree.

If you specify *Message* and *LocalEnvironment*, the output message assembly contains the new *LocalEnvironment* tree and a new message.

If you specify *None*, the output message assembly contains both the input *LocalEnvironment* tree and the input message.

Because the input message cannot be changed, you should specify *OutputLocalEnvironment* or *Environment* for the output data location and report location.

- Select an option for the *Copy Message* property to determine what is included in a new generated message.

This property has an effect when *Message* or *Message and LocalEnvironment* is selected for the *Generate Mode* property. The *Copy Message* property has no effect if *None* or *LocalEnvironment* is selected for the *Generate Mode* property.

If you want the generated output message to contain a copy of the input message headers, but not Input body (that is, all children except the last child of *Root*), specify *Copy Headers* in the *Copy Message* property.

If you do not want the generated output message to contain anything from the input message, specify *None* in the *Copy Message* property.

If you want the generated output message to contain a full copy of the input message, specify *Copy Entire Message* in the *Copy Message* property.

- Select an option for the *Copy Local Environment* property to determine what is included in a new generated message.

This property has an effect when *LocalEnvironment* or *Message and LocalEnvironment* is selected for the *Generate Mode* property. The *Copy Local Environment* property has no effect if *None* or *Message* is selected for the *Generate Mode* property.

If you want the new generated output local environment to contain a copy of the input local environment, specify *Copy Entire LocalEnvironment* in the *Copy Local Environment* property.

If you specify *None* in the *Copy Local Environment* property, the new generated local environment will not contain a copy of the input *LocalEnvironment*.

- If you want the data set to remain open after an operation on the data set, specify *Until Idle* in the *Retain File* property. The data set is then closed only when the idle time period expires. If you do not specify a value, the data set is closed when the current operation completes.
- If you want to keep the position in the VSAM data set for subsequent VSAM nodes in the message flow, specify *Downstream* in the *Hold Position* property. If you do not specify a value, subsequent VSAM nodes might need to reposition in the VSAM data set.
- If you want to allow records to be updated by subsequent VSAM nodes, specify *Update* in the *Additional Open Intent* property.
- If you want to read less than a complete record, specify in the *Maximum Record Length* property a value for the maximum number of bytes that can be read for each record. If you do not specify a value, a complete record is read.

Configure the **Default** properties of a VSAMRead node by selecting **Default** in the properties dialog navigator, and then doing the following:

- For the *Domain* property, enter the name of the parser that you are using to parse the record. Valid options include:
  - MRM
  - XML
  - XMLNS
  - XMLNSC
  - JMSTMap
  - JMSSStream

- BLOB
- If you are using the MRM parser, enter the message set in the *Message Set* property.  
Leave the *Message Set* field blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- If you are using the MRM parser, enter the message type in the *Message Type* property.  
Leave the *Message Type* property blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- Enter the format of the message in the *Message Format* property.  
Leave the *Message Format* property blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- Enter in the *CCSID* property the code page to use when you parse the message.
- Enter in the *Encoding* property the encoding to use when you parse the message.

Configure the **Request** properties of a VSAMRead node by selecting **Request** in the properties dialog navigator, and then doing the following:

- Specify in the *Request Location* property the ESQL location of the request message. If you do not specify a value, the default location `InputBody.VSAM.Request` is used.
- Select from the options for the *Position Mode* property to determine how to position at the record that is to be read. The default, `KEY_FIRST`, positions at the first record in the VSAM data set.
- Specify in the *Key* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be read. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- The *Key Type* property describes how the key is interpreted. Select `Parser String` in the *Key Type* property if you want the key to be interpreted using the code page that is specified in the *CCSID* property. Otherwise, with the default `Hexadecimal` set, the key is interpreted as a hexadecimal string.
- Specify in the *RBA* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be read. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- Specify in the *RRN* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be read. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.

Configure the **Result** properties of a VSAMRead node by selecting **Result** in the properties dialog navigator, and then doing the following:

- Specify in the *Output Data Location* property the ESQL path expression for the location of where the VSAM record data is written.

Configure the **Status** properties of a VSAMRead node by selecting **Status** in the properties dialog navigator, and then doing the following:

- If you want a report to be included in the output message assembly, specify `Yes` in the *Include Record Report* property and an ESQL path expression in the *Report Location* to describe where in the output message assembly the report is located.

If you leave the *Report Location* property blank, and the *Generate Mode* property is *None* or *Message*, the report location is `InputLocalEnvironment.VSAM.Report`; otherwise the report location is `OutputLocalEnvironment.VSAM.Report`.

Select **Description** in the properties dialog navigator if you want to give a short description, a long description, or both.

Click **Apply** to make the changes to the VSAMRead node without closing the properties dialog.

Click **OK** to apply the changes and close the properties dialog.

Click **Cancel** to close the properties dialog and discard all the changes that you have made to the properties of the node.

---

## Terminals of a VSAMRead node

The terminals of a VSAMRead node are described in the following table:

Terminal	Description
In	The input terminal that receives the input message.
Out	An output terminal through which a message is routed following the successful read of the VSAM record.
Failure	An output terminal through which a failure message is routed if there is a failure within the node.

### Connecting the output terminals

If you want to do further processing within the message flow, connect one or more of the output terminals to other nodes. If you do not connect any of the output terminals to another node, no further processing is performed.

If you connect the *Failure* terminal to another node, and an error occurs within the VSAMRead node, the message and information about the error are propagated to the node that is connected to the Failure terminal.

If you do not connect the *Failure* terminal to another node, and an error occurs within the VSAMRead node, information about the error is passed back to the previous node in the message flow.

If you do not connect the *Out* terminal of a node, any information about the successful completion of the processing of the node is discarded.

---

## Properties of a VSAMRead node

The following tables describe the properties of the VSAMInput 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.

The **Basic** property of the VSAMRead node is described in the following table:

Property	M	C	Default	Description
Default File Name	No	Yes	blank	The fully qualified name of the VSAM data set of the form <code>/' data set name'</code> .

The **Advanced** properties of the VSAMRead node are described in the following table:

Property	M	C	Default	Description
Generate Mode	Yes	No	'Message'	Specifies what is generated and subsequently propagated in the output message assembly.  Valid values are: 'Message' 'LocalEnvironment' 'Message And LocalEnvironment' 'None'
Copy message	No	No	'Copy Headers'	Specifies how much of the input message is copied into the output message.  Valid values are: 'None' 'Copy Entire Message' 'Copy Headers'
Copy Local Environment	Yes	No	'None'	Specifies whether the input local environment is copied into the output local environment.  Valid values are: 'None' 'Copy Entire LocalEnvironment'
Retain File	Yes	No	'No'	Indicates whether the file remains open.  Valid values are: 'No' 'Until Idle'
Hold Position	Yes	No	'No Hold'	Specifies whether the file can be used by another VSAM node later in the message flow that has specified the location option 'Current'.  Valid values are: 'No Hold' 'Downstream'

Property	M	C	Default	Description
Additional Open Intent	Yes	No	'None'	Allows update intent to be added to the file options that are requested by this node.  Valid values are: 'None' 'Update'
Maximum Record Length	No	No	0	The maximum number of bytes that is read for each record.  If this property is 0 (the default), the whole record is read.

The **Default** properties of the VSAMRead node are described in the following table:

Property	M	C	Default	Description
Message Domain	No	No	Blank	The message domain to use when you parse the message.
Message Set	No	No	Blank	The message set to use when you parse the message.
Message Type	No	No	Blank	The message type to use when you parse the message.
Message Format	No	No	Blank	The message format to use when you parse the message.
CCSID	No	No	0	The code page to use when you parse the message.
Encoding	No	No	0	The encoding to use when you parse the message.

The **Request** properties of the VSAMRead node are described in the following table:

Property	M	C	Default	Description
Request Location	No	No	See note below.	Specifies the location of the request message. This property takes the form of an ESQL field reference.
Position Mode	No	No	'KEY_FIRST'	Method of positioning to the record that is read.  Valid values are: 'KEY_FIRST' 'KEY_LAST' 'KEY_EQ' 'KEY_GE' 'KEY_EQ_BWD' 'RBA_EQ' 'RBA_EQ_BWD' 'CURRENT'

Property	M	C	Default	Description
Key	No	No	Blank	Key to be used with key-based position modes. This property is ignored if its value is blank.
Key Type	No	No	'Hexadecimal'	Describes the format of the key.  Valid values are: 'Hexadecimal' 'Parser String'
RBA	No	No	0	RBA to be used with RBA-based position modes. This property is ignored if its value is 0.
RRN	No	No	0	RRN to be used with RRN-based position modes. This property is ignored if its value is 0.

**Note:** The default value for the *Request Location* property is `InputLocalEnvironment.VSAM.Request`.

The **Result** property of the VSAMRead node is described in the following table:

Property	M	C	Default	Description
Output Data Location	No	No	Blank	Location in the message assembly where the VSAM record data is written. This property takes the form of an ESQL field reference.

The **Status** properties of the VSAMRead node are described in the following table:

Property	M	C	Default	Description
Include Record Report	No	No	'No'	Whether to include in the output record a report that describes the records read.  Valid values are: 'Yes' 'No'
Report Location	No	No	Blank	Where the report is placed. This property takes the form of an ESQL field reference.

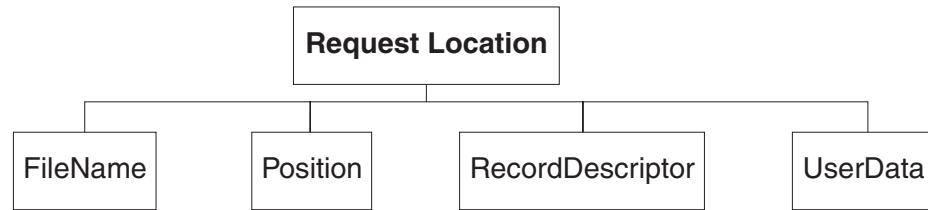
---

## VSAM Request message

The VSAM Request message contains information about the VSAM operation. The location of the VSAM Request message is specified by the Request Location property of the VSAMRead node. All elements of the VSAM Request message are optional.

All values that are specified in the VSAM Request message override the values of corresponding properties in the VSAMRead node. If the message tree contains no request at the Request Location, all values are taken from the node properties.

The structure of the VSAM Request message that is used by a VSAMRead node is shown in the following figure:



In the tree structure shown, Request has the following children:

- **FileName:** The name of the VSAM data set to be used for this request; this overrides the Default File Name property. This element must be supplied with a data type of CHARACTER.
- **Position**  
The fields and data type for each element of the Position subtree are shown in the following table. The last column of the table shows the node property that the field overrides.

Field name	Data type	Node property overridden
Mode	CHARACTER	Position Mode (see note below)
Key	CHARACTER	Key
KeyType	CHARACTER	Key Type
RBA	INTEGER or CHARACTER	RBA
RRN	INTEGER or CHARACTER	RRN

**Note:** The valid values for the Mode field are the same as for the *Position Mode* property, except that 'CURRENT' is not valid.

- **RecordDescriptor**  
The fields and data type for each element of the RecordDescriptor subtree are shown in the following table. The last column of the table shows the corresponding node property.

Field name	Data type	Node property overridden
Domain	CHARACTER	Message Domain
MessageSet	CHARACTER	Message Set
MessageType	CHARACTER	Message Type
MessageFormat	CHARACTER	Message Format
CCSID	INTEGER or CHARACTER	CCSID
Encoding	INTEGER or CHARACTER	Encoding

- **UserData:** A single element of any data type that is copied to the VSAM Report message.



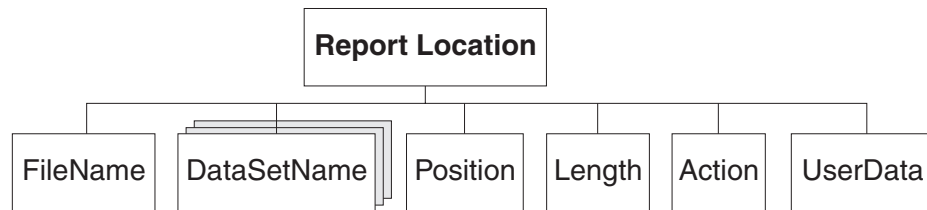
---

## VSAM Report message

A Report message can be part of the output message that is propagated through the *Out* terminal of the VSAMRead node to another node in the message flow.

If a report message is included as part of the output message of a VSAMRead node, its location is specified in the Report Location property of the VSAMRead node.

The structure of a report message for the VSAMRead node is shown in the following figure:



In the tree structure shown, Report has the following children:

- **FileName:** The name of the VSAM data set that is specified for this request. This element is of type CHARACTER.
- **DataSetName:** A repeating element of type CHARACTER that contains the data set qualifiers for the file used by this node.
- **Position**

**Note:** If no record is found, this child is not included in the Report message.

The fields and data type for each element of the Position subtree are shown in the following table.

Field name	Data type	Description
Key	CHARACTER	Key of the record that has been read. This element is included following the successful read of a record from a KSDS, KSDS PATH, or ESDS PATH data set.
KeyType	CHARACTER	Describes the type of the previous Key element..
RBA	INTEGER	RBA of the record that has been read. This element is included following the successful read of a record from a KSDS or ESDS data set.
RRN	INTEGER	RRN of the record that has been read. This element is included following the successful read of a record from an RRDS data set.

- Length: The length of the record that was read by this node. This element is of data type INTEGER. If no record was found, this has the value 0.
- Action: The VSAM operation performed by this node. This element is of type CHARACTER and, for the VSAMRead node, its value is 'Read'.
- UserData: A copy of the UserData element that was specified in the VSAM Request message. The element is only included if a UserData element is present in the VSAM request.

---

## VSAMWrite node

Use a VSAMWrite node to write a record to a VSAM data set.

To do this, send a message to the *In* terminal of a VSAMWrite node. Include a VSAM Request in this message if you want to override the settings of properties that are configured in the VSAMWrite node.

A VSAM Report message can also be sent, as part of the output message, to the next node in the message flow.

A VSAMWrite node is represented in the workbench by the following icon:



---

## Configuring a VSAMWrite node

When you have put an instance of the VSAMWrite node into a message flow, you can configure it.

**Note:** Values that you specify for properties in the node are superseded by any corresponding fields that are included in a VSAM Request that is part of the input message. All supplied values are validated even when they might not be used.

To configure the node do the following:

Right-click the node in the editor view of the message flow and click **Properties**. The **Basic** properties of the node are displayed in the properties dialog.

Configure the **Basic** properties of a VSAMWrite node by doing the following:

- Specify the *Default file name*. This should be the fully resolved name of the VSAM data set that you want to perform operations on.

Configure the **Advanced** properties of a VSAMWrite node by selecting **Advanced** in the properties dialog navigator, and then doing the following:

- Specify a value for the *Generate Mode* property to determine the content of the output message assembly.

If you specify *LocalEnvironment*, the output message assembly contains the new *LocalEnvironment* tree and the input message tree.

If you specify *None*, the output message assembly contains both the input *LocalEnvironment* tree and the input message.

Because the input message cannot be changed, you should specify *InputLocalEnvironment* or *Environment* for the report location.

- If you want the output local environment to contain a copy of the input local environment, specify *Copy Entire Local Environment* in the *Copy Local Environment* property.

If you do not specify a value for the *Copy Local Environment* property, or if you specify the value *None*, the output local environment is either empty, or contains only a copy of the VSAM data written and, optionally, report information.

- If you want the VSAM data set to remain open after an operation on the data set, select *Until Idle* in the *Retain File* property. The data set remains open until the idle time period expires, unless the data set is explicitly closed before this. If you select *No* for this property, the data set is closed when the current operation completes.

- Specify in the *Padding* property a string that is used if the data being written is shorter than the minimum length of records in the data set.

The minimum length depends on the type of the VSAM data set:

- KSDS, ESDS, KSDS\_PATH, or ESDS\_PATH: no minimum; padding is ignored.
- RRDS: minimum = fixed length of records for that data set.

If this property is left blank, no padding occurs and an exception is thrown if the data is shorter than the minimum length.

**Note:** Padding only occurs using the whole padding string that is specified. For example, if there are 8 bytes that can be padded, but the padding string that is specified is only 5 bytes long, only 5 bytes are padded. Similarly, if there are 11 bytes that can be padded, only 10 (2 times 5) bytes are padded.

- Specify in the *Padding Type* property a value that describes how the string that you described in the *Padding* property is interpreted.

Valid values for this property are:

'Parser String'

'Hexadecimal'

The default value for this property is *Parser String*.

- If you want the data that is written to be truncated to the maximum length of a record in the data set, select the *Truncate* check box; otherwise, make sure that the *Truncate* check box is cleared.

If the *Truncate* check box is cleared, an exception is thrown if the data is longer than the maximum length.

Configure the **Default** properties of a VSAMWrite node by selecting **Default** in the properties dialog navigator, and then doing the following:

- If you are using the MRM parser, enter the message set in the *Message Set* property.  
Leave the *Message Set* field blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- If you are using the MRM parser, enter the message type in the *Message Type* property.  
Leave the *Message Type* property blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- Enter the format of the message in the *Message Format* property.  
Leave the *Message Format* property blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- Enter in the *CCSID* property the code page to use when you parse the message.
- Enter in the *Encoding* property the encoding to use when you parse the message.

Configure the **Request** properties of a VSAMWrite node by selecting **Request** in the properties dialog navigator, and then doing the following:

- Specify in *Input Data Location* the ESQL location of the data that is to be written as a record to the VSAM data set.

**Note:** If you do not specify a value for this property, *InputBody* is assumed.

- Specify in the *Request Location* property the ESQL location of the request message. If you do not specify a value, the default location `InputBody.VSAM.Request` is used.
- Specify in the *RRN* property the RRN of the record that is to be written.

Configure the **Status** properties of a VSAMWrite node by selecting **Status** in the properties dialog navigator, and then doing the following:

- If you want a report to be included in the output message assembly, specify **Yes** in the *Include Record Report* property and an ESQL path expression in the *Report Location* to describe where in the output message assembly the report is located. If you leave the *Report Location* property blank, and the *Generate Mode* property is **None**, the report location is `InputLocalEnvironment.VSAM.Report`; otherwise the report location is `OutputLocalEnvironment.VSAM.Report`.

Select **Description** in the properties dialog navigator if you want to give a short description, a long description, or both.

Click **Apply** to make the changes to the VSAMWrite node without closing the properties dialog.

Click **OK** to apply the changes and close the properties dialog.

Click **Cancel** to close the properties dialog and discard all the changes that you have made to the properties of the node.

---

## Terminals of a VSAMWrite node

The terminals of a VSAMWrite node are described in the following table:

Terminal	Description
In	The input terminal that receives the VSAM request message.
Out	An output terminal through which a message is routed following the successful write of a VSAM record.
Failure	An output terminal through which a failure message is routed if there is a failure within the node.

### Connecting the output terminals

If you want to do further processing within the message flow, connect one or more of the output terminals to other nodes in the message flow. If you do not connect any of the output terminals to another node, no further processing is performed.

If you connect the *Failure* terminal to another node, and an error occurs within the VSAMWrite node, the message and the exception are propagated to the node that is connected to the Failure terminal.

If you do not connect the *Failure* terminal to another node, and an error occurs within the VSAMWrite node, the exception is passed back to the previous node in the message flow.

If you do not connect the *Out* terminal of a node, any information about the successful completion of the processing of the node is discarded.

---

## Properties of a VSAMWrite 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.

The **Basic** properties of the VSAMWrite node are described in the following table:

Property	M	C	Default	Description
Default File Name	No	Yes	Blank	The fully qualified name of the VSAM data set of the form // ' data set name'.

The **Advanced** properties of the VSAMWrite node are described in the following table:

Property	M	C	Default	Description
Generate Mode	Yes	No	'None'	Valid values are: 'LocalEnvironment' 'None'
Copy Local Environment	Yes	No	'None'	Valid values are: 'None' 'Copy Entire LocalEnvironment'
Retain File	Yes	No	'No'	Indicates whether the file remains open after the initial opening of the file.  Valid values are: 'No' 'Until Idle'
Padding	No	No	Blank	Specifies whether to pad when the data being written is less than the record length of the VSAM file being written to, and what to pad with.
Padding Type	No	No	'Parser String'	Specifies how the padding is to be done.  Valid values are: Hexadecimal Parser String

Property	M	C	Default	Description
Truncate	No	No	Cleared	The <i>Truncate</i> check box should be selected if, when the data being written is greater than the maximum record length of the VSAM data set that is being written to, you want the data to be truncated.  If the check box is cleared, an exception is thrown if the data to be written is larger than the maximum record size of the VSAM data set that is being written to.

The **Default** properties of the VSAMWrite node are described in the following table.

Property	M	C	Default	Description
Message Set	No	No	Blank	The message set to use when you write the message.
Message Type	No	No	Blank	The message type to use when write the message.
Message Format	No	No	Blank	The message format to use when you write the message.
CCSID	No	No	0	The code page to use when you write the message.
Encoding	No	No	0	The encoding to use when you write the message.

The **Request** properties of the VSAMWrite node are described in the following table.

Property	M	C	Default	Description
Input Data Location	No	No	InputBody	ESQL path expression that indicates the location within the input message assembly of the data that will be written as a record to the VSAM file.  If nothing is specified, InputBody is assumed to be the location of the data.
Request Location	No	No	See note below.	Specifies (in ESQL) the location of the request message.
RRN	No	No	0	RRN for the record that is to be written.

**Note:** The default value for the *Request Location* property is `InputLocalEnvironment.VSAM.Request`.

The **Status** properties of the VSAMWrite node are described in the following table.

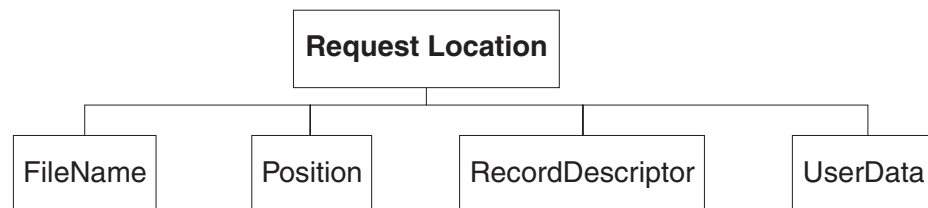
Property	M	C	Default	Description
Include Record Report	No	No	'No'	Whether to include in the output record a report that describes the records written.  Valid values are: 'Yes' 'No'
Report Location	No	No	Blank	Where the report is placed.

## VSAM Request message

The VSAM Request message contains information about the VSAM operation. The location of the VSAM Request message is specified by the Request Location property of the VSAMWrite node. All elements of the VSAM Request message are optional.

All values that are specified in the VSAM Request message override the values of corresponding properties in the VSAMWrite node. If the message tree contains no request at the Request Location, all values are taken from the node properties.

The structure of the VSAM Request message that is used by a VSAMWrite node is shown below.



In the tree structure shown, Request has the following children:

- **FileName:** The name of the VSAM data set to be used for this request; this overrides the Default File Name property. This element must be supplied with a data type of CHARACTER.

- **Position**

The fields and data type for each element of the Position subtree are shown in the following table. The last column of the table shows the node property that the field overrides.

Field name	Data type	Node property overridden
RRN	INTEGER or CHARACTER	RRN

- **RecordDescriptor**

The fields and data type for each element of the RecordDescriptor subtree are shown in the following table. The last column of the table shows the corresponding node property.



Field name	Data type	Node property overridden
MessageSet	CHARACTER	Message Set
MessageType	CHARACTER	Message Type
MessageFormat	CHARACTER	Message Format
CCSID	INTEGER or CHARACTER	CCSID
Encoding	INTEGER or CHARACTER	Encoding

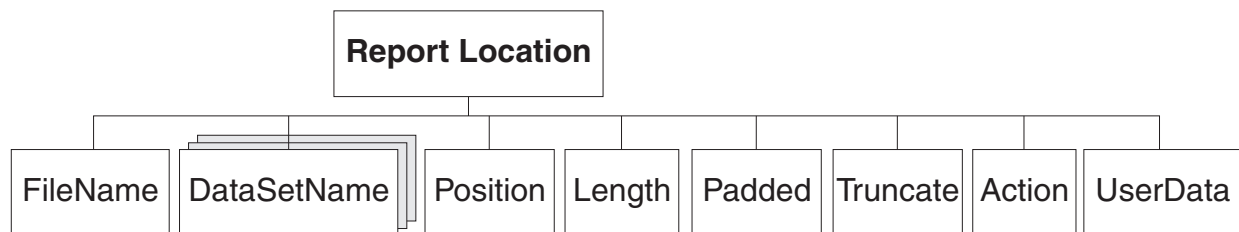
- UserData: A single element of any data type that is copied to the VSAM Report message.

## VSAM Report message

A Report message can be part of the output message that is propagated through the Out terminal of the VSAMWrite node to another node in the message flow.

If a report message is included as part of the output message of a VSAMWrite node, its location is specified in the Report Location property of the VSAMWrite node.

The structure of a report message for the VSAMWrite node is shown in the following figure:



In the tree structure shown, Report has the following children:

- FileName: The name of the VSAM data set that is specified for this request. This element is of type CHARACTER.
- DataSetName: A repeating element of type CHARACTER that contains the data set qualifiers for the file used by this node.
- Position

**Note:** If no record is found, this child is not included in the Report message.

The fields and data type for each element of the Position subtree are shown in the following table.

Field name	Data type	Description
Key	CHARACTER	Key of the record that has been read. This element is included following the successful read of a record from a KSDS, KSDS PATH, or ESDS PATH data set.

Field name	Data type	Description
KeyType	CHARACTER	Describes the type of the previous Key element..
RBA	INTEGER	RBA of the record that has been read. This element is included following the successful read of a record from a KSDS or ESDS data set.
RRN	INTEGER	RRN of the record that has been read. This element is included following the successful read of a record from an RRDS data set.

- Length: The length of the record that was read by this node. This element is of data type INTEGER.
- Action: The VSAM operation that was performed by this node. This element is of type CHARACTER and, for the VSAMWrite node, its value is 'Write'.
- Padded: Whether padding was applied to any records that were written by this node. Valid values of this field are 'True' and 'False'.
- Truncate: Whether records that were written by this node were truncated. Valid values of this field are 'True' and 'False'.
- UserData: A copy of the UserData element that was specified in the VSAM Request message. The element is only included if a UserData element is present in the VSAM request.

---

## VSAMUpdate node

Use a VSAMUpdate node to update a record in a VSAM data set, and propagate the record to the rest of the message flow.

To do this, send a message to the *In* terminal of a VSAMUpdate node. Include a VSAM Request in this message if you want to override the settings of properties that are configured in the VSAMUpdate node.

A VSAM Report message can also be sent, as part of the output message, to the next node in the message flow.

A VSAMUpdate node is represented in the workbench by the following icon:



---

## Configuring a VSAMUpdate node

When you have put an instance of the VSAMUpdate node into a message flow, you can configure it.

**Note:** Values that you specify for properties in the node are superseded by any corresponding fields that are included in a VSAM Request that is part of the input message. All supplied values are validated even when they might not be used.

To configure the node, do the following:

Right-click the node in the editor view of the message flow and click **Properties**. The **Basic** properties of the node are displayed in the properties dialog.

Configure the **Basic** properties of a VSAMUpdate node by doing the following:

- Specify the *Default file name*. This should be the fully resolved name of the VSAM data set that you want to perform operations on.

Configure the **Advanced** properties of a VSAMUpdate node by selecting **Advanced** in the properties dialog navigator, and then doing the following:

- Specify a value for the *Generate Mode* property to determine the content of the output message assembly.

If you specify *LocalEnvironment*, the output message assembly contains the new *LocalEnvironment* tree and the input message tree.

If you specify *None*, the output message assembly contains both the input *LocalEnvironment* tree and the input message.

Because the input message cannot be changed, you should specify *InputLocalEnvironment* or *Environment* for the report location.

- If you want the output local environment to contain a copy of the input local environment, specify *Copy Entire Local Environment* in the *Copy Local Environment* property.

If you do not specify a value for the *Copy Local Environment* property, or specify the value *None*, the output local environment is either empty, or contains only a copy of the VSAM data updated (and, optionally, report information).

- If you want the VSAM data set to remain open after an operation on the data set, select *Until Idle* in the *Retain File* property. The data set remains open until the idle time period expires, unless the data set is explicitly closed before this. If you select *No* for this property, the data set is closed when the current operation completes.

- Specify in the *Padding* property a string that is used if the data being written is shorter than the minimum length of the record being updated.

The minimum length depends on the type of the VSAM data set:

- KSDS or KSDS\_PATH: no minimum; padding is ignored.
- ESDS, ESDS\_PATH, or RRDS: minimum = length of the existing record.

If this property is left blank, no padding occurs and a partial update takes place if the data is shorter than the minimum length.

**Note:** Padding only occurs using the whole padding string that is specified. For example, if there are 8 bytes that can be padded, but the padding string that is specified is only 5 bytes long, only 5 bytes are padded. Similarly, if there are 11 bytes that can be padded, only 10 (2 times 5) bytes are padded.

- Specify in the *Padding Type* property a value that describes how the string that you described in the *Padding* property is interpreted.

Valid values for this property are:

- 'Parser String'
- 'Hexadecimal'

The default value for this property is *Parser String*.

- If you want the data that is written to be truncated to the maximum length of the VSAM record that is being updated, select the *Truncate* check box; otherwise, make sure that the *Truncate* check box is cleared.

If the *Truncate* check box is cleared, an exception is thrown if the data is longer than the maximum length.

The maximum length depends on the type of the VSAM data set:

- KSDS or KSDS\_PATH: maximum = maximum record length for the data set.
- ESDS, ESDS\_PATH, or RRDS: maximum = length of the existing record.

Configure the **Default** properties of a VSAMUpdate node by selecting **Default** in the properties dialog navigator, and then doing the following:

- If you are using the MRM parser, enter the message set in the *Message Set* property.  
Leave the *Message Set* field blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- If you are using the MRM parser, enter the message type in the *Message Type* property.  
Leave the *Message Type* property blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- Enter the format of the message in the *Message Format* property.  
Leave the *Message Format* property blank for XML, XMLNS, XMLNSC, JMS, and BLOB parsers.
- Enter in the *CCSID* property the code page to use when you parse the message.

- Enter in the *Encoding* property the encoding to use when you parse the message.

Configure the **Request** properties of a VSAMUpdate node by selecting **Request** in the properties dialog navigator, and then doing the following:

- Specify in the *Input Data Location* property an ESQL expression for the location in the input message of the data that is to be written as a record to the VSAM file.

**Note:** If you do not specify a value for this property, *InputBody* is assumed.

- Specify in the *Request Location* property the ESQL location of the request message. If you do not specify a value, the default location `InputBody.VSAM.Request` is used.
- Select from the options for the *Position Mode* property to determine how to position at the record that is to be updated. The default, `KEY_FIRST`, positions at the first record in the VSAM data set.
- Specify in the *Key* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be updated. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- The *Key Type* property describes how the key is interpreted. Select `Parser String` in the *Key Type* property if you want the key to be interpreted using the code page that is specified in the *CCSID* property. Otherwise, with the default `Hexadecimal` set, the key is interpreted as a hexadecimal string.
- Specify in the *RBA* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be updated. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- Specify in the *RRN* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be updated. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.

Configure the **Status** properties of a VSAMUpdate node by selecting **Status** in the properties dialog navigator, and then doing the following:

- If you want a report to be included in the output message assembly, specify `Yes` in the *Include Record Report* property and an ESQL path expression in the *Report Location* to describe where in the output message assembly the report is located. If you leave the *Report Location* property blank, and the *Generate Mode* property is `None`, the report location is `InputLocalEnvironment.VSAM.Report`; otherwise the report location is `OutputLocalEnvironment.VSAM.Report`.

Select **Description** in the properties dialog navigator if you want to give a short description, a long description, or both.

Click **Apply** to make the changes to the VSAMUpdate node without closing the properties dialog.

Click **OK** to apply the changes and close the properties dialog.

Click **Cancel** to close the properties dialog and discard all the changes that you have made to the properties of the node.

---

## Terminals of a VSAMUpdate nodes

The terminals of a VSAMUpdate node are described in the following table:

Terminal	Description
In	The input terminal that receives the VSAM request message.
Out	An output terminal through which a message is routed following the successful update of the VSAM record.
Failure	An output terminal through which a failure message is routed if there is a failure within the node. The failure message contains a copy of the VSAM request and an exception list that describes the failure.

### Connecting the output terminals

If you want to do further processing within the message flow, connect one or more of the output terminals to other nodes in the message flow. If you do not connect any of the output terminals to another node, no further processing is performed.

If you connect the *Failure* terminal to another node, and an error occurs within the VSAMUpdate node, the message and information about the error are propagated to the node that is connected to the Failure terminal.

If you do not connect the *Failure* terminal to another node, and an error occurs within the VSAMUpdate node, information about the error is passed back to the previous node in the message flow.

If you do not connect the *Out* terminal of a node, any information about the successful completion of the processing of the node is discarded.

---

## Properties of a VSAMUpdate 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.

The **Basic** properties of the VSAMUpdate node are described in the following table:

Property	M	C	Default	Description
Default File Name	No	Yes	Blank	The fully qualified name of the VSAM data set of the form // ' data set name'.

The **Advanced** properties of the VSAMUpdate node are described in the following table:

Property	M	C	Default	Description
Generate Mode	Yes	No	'None'	Describes what the output message assembly contains.  Valid values are: 'LocalEnvironment' 'None'
Copy Local Environment	Yes	No	'None'	Valid values are: 'None' 'Copy Entire LocalEnvironment'
Retain File	Yes	No	'No'	Indicates whether the file remains open after the initial opening of the file.  Valid values are: 'No' 'Until Idle'
Padding	No	No	Blank	Specifies whether to pad when the data being written is less than the record length of the VSAM file being written to, and what to pad with.
Padding Type	No	No	'ParserString'	Specifies how the padding is to be done.  Valid values are: ParserString Hexadecimal
Truncate	No	No	Cleared	The <i>Truncate</i> check box should be selected if, when the data being written is greater than the maximum record length of the VSAM data set that is being written to, you want the data to be truncated.  If the check box is left cleared, an exception is thrown if the data to be written is larger than the maximum record size of the VSAM data set that is being written to.

The **Default** properties of the VSAMUpdate node are described in the following table.

Property	M	C	Default	Description
Message Set	No	No	Blank	The message set to use when you write the message.
Message Type	No	No	Blank	The message type to use when you write the message.
Message Format	No	No	Blank	The message format to use when you write the message.
CCSID	No	No	0	The code page to use when you write the message.

Property	M	C	Default	Description
Encoding	No	No	0	The encoding to use when you write the message.

The **Request** properties of the VSAMUpdate node are described in the following table.

Property	M	C	Default	Description
Input Data Location	No	No	InputBody	ESQL path expression that indicates the location within the input message assembly of the data that will be written as a record to the VSAM file.
Request Location	No	No	See note below.	Specifies, in ESQL, the location of the request message.
Position Mode	No	No	'KEY_FIRST'	Method of positioning to the record that is read.  Valid values are: 'KEY_FIRST' 'KEY_LAST' 'KEY_EQ' 'KEY_GE' 'KEY_EQ_BWD' 'RBA_EQ' 'RBA_EQ_BWD' 'CURRENT'
Key	No	No	Blank	Key to be used with key-based position modes.
Key Type	No	No	'Hexadecimal'	Describes the format of the key.  Valid values are: 'Hexadecimal' 'Parser String'
RBA	No	No	0	RBA to be used with RBA-based position modes.
RRN	No	No	0	RRN to be used with RRN-based position modes.

**Note:** The default value for the *Request Location* property is `InputLocalEnvironment.VSAM.Request`.

The **Status** properties of the VSAMUpdate node are described in the following table.



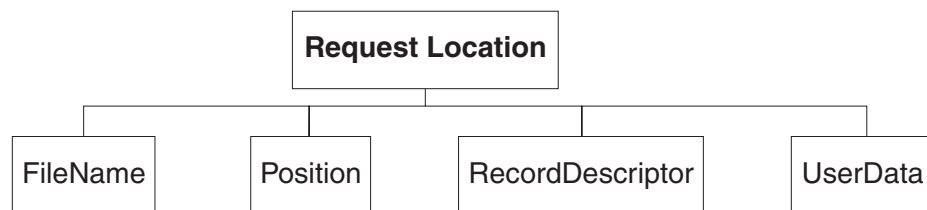
Property	M	C	Default	Description
Include Record Report	No	No	'No'	Whether to include in the output record a report that describes the records written.  Valid values are: 'Yes' 'No'
Report Location	No	No	Blank	Where the report is placed.

## VSAM Request message

The VSAM Request message contains information about the VSAM operation. The location of the VSAM request message is specified by the Request Location property of the VSAMUpdate node. All elements of the VSAM Request message are optional.

All values that are specified in the VSAM request message override those that are defined in the corresponding properties of the VSAMUpdate node. If the tree contains no request at the Request Location, all values are taken from the node properties.

The structure of the VSAM Request message that is used by a VSAMUpdate node is shown in the following figure:



In the tree structure shown, Request has the following children:

- **FileName:** The name of the VSAM data set to be used for this request; this overrides the Default File Name property. This element must be supplied with a data type of CHARACTER.
- **Position**

The fields and data type for each element of the Position subtree are shown in the following table. The last column of the table shows the node property that the field overrides.

Field name	Data type	Node property overridden
Mode	CHARACTER	Position Mode (see note below)
Key	CHARACTER	Key
Key Type	CHARACTER	Key Type
RBA	INTEGER or CHARACTER	RBA

Field name	Data type	Node property overridden
RRN	INTEGER or CHARACTER	RRN

**Note:** The valid values for the Mode field are the same as for the *Position Mode* property, except that 'CURRENT' is not valid.

- RecordDescriptor

The fields and data type for each element of the RecordDescriptor subtree are shown in the following table. The last column of the table shows the corresponding node property.

Field name	Data type	Node property overridden
MessageSet	CHARACTER	Message Set
MessageType	CHARACTER	Message Type
MessageFormat	CHARACTER	Message Format
CCSID	INTEGER or CHARACTER	CCSID
Encoding	INTEGER or CHARACTER	Encoding

- UserData: A single element of any data type that is copied to the VSAM Report and VSAM Status messages.

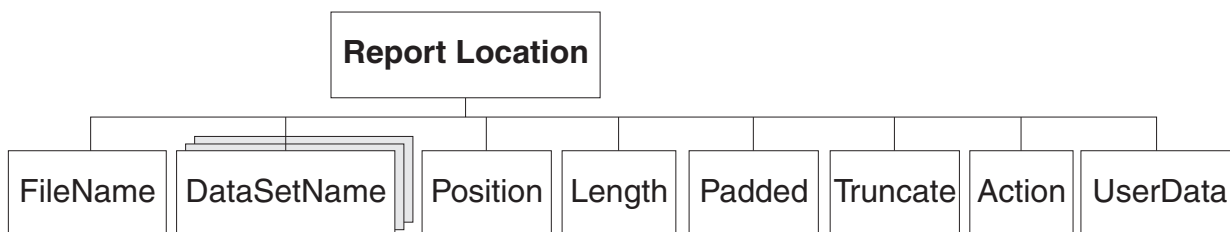
---

## VSAM Report message

A Report message can be part of the output message that is propagated through the Out terminal of the VSAMUpdate node to another node in the message flow.

If a report message is included as part of the output message of a VSAMUpdate node, its location is specified in the Report Location property of the VSAMUpdate node.

The structure of a report message for the VSAMUpdate node is shown in the following figure:



In the tree structure shown, Report has the following children:

- FileName: The name of the VSAM data set that is specified for this request. This element is of type CHARACTER.
- DataSetName: A repeating element of type CHARACTER that contains the data set qualifiers for the file used by this node.
- Position

**Note:** If no record is found, this child is not included in the Report message. The fields and data type for each element of the Position subtree are shown in the following table.

Field name	Data type	Description
Key	CHARACTER	Key of the record that has been read. This element is included following the successful read of a record from a KSDS, KSDS PATH, or ESDS PATH data set.
KeyType	CHARACTER	Describes the type of the previous Key element..
RBA	INTEGER	RBA of the record that has been read. This element is included following the successful read of a record from a KSDS or ESDS data set.
RRN	INTEGER	RRN of the record that has been read. This element is included following the successful read of a record from an RRDS data set.

- **Length:** The length of the record that was read by this node. This element is of data type INTEGER.
- **Action:** The VSAM operation that was performed by this node. This element is of type CHARACTER and, for the VSAMUpdate node, its value is 'Update'.
- **Padded:** Whether padding was applied to any records that were updated by this node. Valid values of this field are 'True' and 'False'.
- **Truncate:** Whether records that were written by this node were truncated. Valid values of this field are 'True' and 'False'.
- **UserData:** A copy of the UserData element that was specified in the VSAM Request message. The element is only included if a UserData element is present in the VSAM request.



---

## VSAMDelete node

Use a VSAMDelete node to delete a record in a VSAM data set, and propagate the record to the rest of the message flow.

To do this, send a message to the *In* terminal of a VSAMDelete node. Include a VSAM Request in this message if you want to override the settings of properties that are configured in the VSAMDelete node.

A VSAM Report message can also be sent, as part of the output message, to the next node in the message flow.

A VSAMDelete node is represented in the workbench by the following icon:



---

## Configuring a VSAMDelete node

When you have put an instance of the VSAMDelete node into a message flow, you can configure it.

**Note:** Values that you specify for properties in the node are superseded by any corresponding fields that are included in a VSAM Request that is part of the input message. All supplied values are validated even when they might not be used.

To configure the node, do the following:

Right-click the node in the editor view of the message flow and click **Properties**. The **Basic** properties of the node are displayed in the properties dialog.

Configure the **Basic** properties of a VSAMDelete node by doing the following:

- Specify the *Default file name*. This should be the fully resolved name of the VSAM file that you want to perform operations on.

Configure the **Advanced** properties of a VSAMDelete node by selecting **Advanced** in the properties dialog navigator, and then doing the following:

- Specify a value for the *Generate Mode* property to determine the content of the output message assembly.

If you specify *LocalEnvironment*, the output message assembly contains the new *LocalEnvironment* tree and the input message tree.

If you specify *None*, the output message assembly contains both the input *LocalEnvironment* tree and the input message.

- If you want the output local environment to contain a copy of the input local environment, specify *Copy Entire Local Environment* in the *Copy Local Environment* property.

If you do not specify a value for the *Copy Local Environment* property, or specify the value *None*, the output local environment is either empty, or contains only the report information.

Configure the **Default** properties of a VSAMDelete node by selecting **Default** in the properties dialog navigator, and then doing the following:

- Enter in the *CCSID* property the code page to use to interpret the *Key* when the *Key Type* property is Parser String.

Configure the **Request** properties of a VSAMDelete node by selecting **Request** in the properties dialog navigator, and then doing the following:

- Specify in the *Request Location* property the ESQL location of the request message. If you do not specify a value, the default location `InputBody.VSAM.Request` is used.
- Select from the options for the *Position Mode* property to determine how to position at the record that is to be deleted. The default, `KEY_FIRST`, positions at the first record in the VSAM data set.
- Specify in the *Key* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be deleted. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- The *Key Type* property describes how the key is interpreted. Select Parser String in the *Key Type* property if you want the key to be interpreted using the code page that is specified in the *CCSID* property. Otherwise, with the default Hexadecimal set, the key is interpreted as a hexadecimal string.
- Specify in the *RBA* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be deleted. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.
- Specify in the *RRN* property a value that, in conjunction with the *Position Mode* property and the data set type, determines the position of the record that is to be deleted. See “Accessing a record within a VSAM data set” on page 3 for more details. This property is ignored if it is blank.

Configure the **Status** properties of a VSAMDelete node by selecting **Status** in the properties dialog navigator, and then doing the following:

- If you want a report to be included in the output message assembly, specify **Yes** in the *Include Record Report* property and an ESQL path expression in the *Report Location* property to describe where in the output message assembly the report is located.

If you leave the *Report Location* property blank, and the *Generate Mode* property is **None**, the report location is `InputLocalEnvironment.VSAM.Report`; otherwise the report location is `OutputLocalEnvironment.VSAM.Report`.

Select **Description** in the properties dialog navigator if you want to give a short description, a long description, or both.

Click **Apply** to make the changes to the VSAMDelete node without closing the properties dialog.

Click **OK** to apply the changes and close the properties dialog.

Click **Cancel** to close the properties dialog and discard all the changes that you have made to the properties of the node.

---

## Terminals of a VSAMDelete node

The terminals of a VSAMDelete node are described in the following table:

Terminal	Description
In	The input terminal that receives the VSAM request message.
Out	An output terminal through which a message is routed following the successful deletion of the VSAM record.
Failure	An output terminal through which a failure message is routed if there is a failure within the node. The failure message contains a copy of the VSAM request and an exception list that describes the failure.

### Connecting the output terminals

Connect one or more of the output terminals to other nodes, if you want to do further processing within the message flow. If you do not connect any output terminals to another node, no further processing is performed.

If you connect the *Failure* terminal to another node, and an error occurs within the VSAMDelete node, the message and information about the error are propagated to the node that is connected to the Failure terminal.

If you do not connect the *Failure* terminal to another node, and an error occurs within the VSAMDelete node, information about the error is passed back to the previous node in the message flow.

If you do not connect the *Out* terminal of a node, any information about the successful completion of the processing of the node is discarded.

---

## Properties of a VSAMDelete 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.

The **Basic** properties of the VSAMDelete node are described in the following table:

Property	M	C	Default	Description
Default File Name	No	Yes	Blank	The fully qualified name of the VSAM data set of the form <code>/'data set name'</code> .

The **Advanced** properties of the VSAMDelete node are described in the following table:

Property	M	C	Default	Description
Generate Mode	Yes	No	'None'	Valid values are: 'LocalEnvironment' 'None'

Property	M	C	Default	Description
Copy Local Environment	Yes	No	'None'	Valid values are: 'None' 'Copy Entire LocalEnvironment'
Retain File	Yes	No	'No'	Indicates whether the file remains open after the initial opening of the file.  Valid values are: 'No' 'Until Idle'

The **Default** properties of the VSAMDelete node are described in the following table.

Property	M	C	Default	Description
CCSID	No	No	0	The code page to use to interpret the <i>Key</i> when the <i>Key Type</i> property is Parser String

The **Request** properties of the VSAMDelete node are described in the following table.

Property	M	C	Default	Description
Request Location	No	No	See note below.	Specifies (in ESQL) the location of the request message.
Position Mode	No	No	'KEY_FIRST'	Method of positioning to the record that is deleted.  Valid values are: 'KEY_FIRST' 'KEY_LAST' 'KEY_EQ' 'KEY_GE' 'KEY_EQ_BWD' 'RBA_EQ' 'RBA_EQ_BWD' 'CURRENT'
Key	No	No	Blank	Key to be used with key-based position modes.
Key Type	No	No	'Hexadecimal'	Describes the format of the key.  Valid values are: 'Hexadecimal' 'Parser String'
RBA	No	No	0	RBA to be used with RBA-based position modes.
RRN	No	No	Blank	RRN to be used with RRN-based position modes.



**Note:** The default value for the *Request Location* property is `InputLocalEnvironment.VSAM.Request`.

The **Status** properties of the VSAMDelete node are described in the following table.

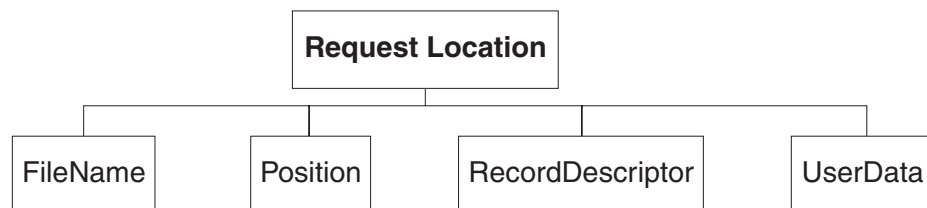
Property	M	C	Default	Description
Include Record Report	No	No	'No'	Whether to include in the output record a report that describes the records written.  Valid values are: 'Yes' 'No'
Report Location	No	No	Blank	Where the report is placed.

## VSAM Request message

The VSAM Request message contains information about the VSAM operation. The location of the VSAM Request message is specified by the Request Location property of the VSAMDelete node. All elements of the VSAM Request message are optional.

All values that are specified in the VSAM Request message override the values of corresponding properties in the VSAMDelete node. If the message tree contains no request at the Request Location, all values are taken from the node properties.

The structure of the VSAM Request message that is used by a VSAMDelete node is shown in the following figure:



In the tree structure shown, Request has the following children:

- **FileName:** The name of the VSAM data set to be used for this request; this overrides the Default File Name property. This element must be supplied with a data type of CHARACTER.
- **Position**

The fields and data type for each element of the Position subtree are shown in the following table. The last column of the table shows the node property that the field overrides.

Field name	Data type	Node property overridden
Mode	CHARACTER	Position Mode (see note below)

Field name	Data type	Node property overridden
Key	CHARACTER	Key
KeyType	CHARACTER	Key Type
RBA	INTEGER or CHARACTER	RBA
RRN	INTEGER or CHARACTER	RRN

**Note:** The valid values for the Mode field are the same as for the *Position Mode* property, except that 'CURRENT' is not valid.

- RecordDescriptor

The fields and data type for each element of the RecordDescriptor subtree are shown in the following table. The last column of the table shows the corresponding node property.

Field name	Data type	Node property overridden
CCSID	INTEGER or CHARACTER	CCSID

- UserData: A single element of any data type that is copied to the VSAM Report message.

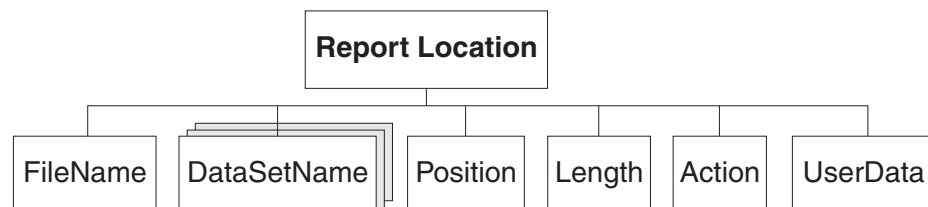
---

## VSAM Report message

A Report message can be part of the output message that is propagated through the Out terminal of the VSAMDelete node to another node in the message flow.

If a report message is included as part of the output message of a VSAMDelete node, its location is specified in the Report Location property of the VSAMDelete node.

The structure of a report message for the VSAMDelete node is shown in the following figure:



In the tree structure shown, Report has the following children:

- FileName: The name of the VSAM data set that is specified for this request. This element is of type CHARACTER.
- DataSetName: A repeating element of type CHARACTER that contains the data set qualifiers for the file used by this node.
- Position

**Note:** If no record is found, this child is not included in the Report message.

The fields and data type for each element of the Position subtree are shown in the following table.

Field name	Data type	Description
Key	CHARACTER	Key of the record that has been read. This element is included following the successful read of a record from a KSDS, KSDS PATH, or ESDS PATH data set.
KeyType	CHARACTER	Describes the type of the previous Key element..
RBA	INTEGER	RBA of the record that has been read. This element is included following the successful read of a record from a KSDS or ESDS data set.
RRN	INTEGER	RRN of the record that has been read. This element is included following the successful read of a record from an RRDS data set.

- Length: The length of the record that was read by this node. This element is of data type INTEGER.
- Action: The VSAM operation performed by this node. This element is of type CHARACTER and, for the VSAMDelete node, its value is 'Read'.
- UserData: A copy of the UserData element that was specified in the VSAM Request message. The element is only included if a UserData element is present in the VSAM request.



---

## Using the VSAM nodes in a message flow

Some samples that show the use of VSAM nodes in message flows are included in this SupportPac.

These samples consist of the following items:

- The WebSphere Business Integration Message Broker Toolkit resources that are required to run the samples:
  - Message flows
  - ESQL
  - MRM Message Set
- Documentation that describes the message flows and ESQL. This documentation is in the following sections of this PDF:
  - “Sample 1 - Using a VSAMInput node” on page 57.
  - “Sample 2 - Using a VSAMWrite node” on page 61.
  - “Sample 3 - Using a VSAMRead node” on page 63.
  - “Sample 4 - Using a VSAMUpdate node” on page 67.
  - “Sample 5 - Using a VSAMDelete node” on page 71.
- Sample JCL that you can use to create and populate your VSAM data set. Refer to the Readme file in the Datasets folder for more information.e B

Before running the samples, you must have the message flows, ESQL, and Message Set in your WebSphere Business Integration Message Broker Toolkit workspace.

For the message flows and ESQL you can either:

- Follow the documentation for each sample to create these resources.
- Import the supplied versions of the samples into an existing message flow project:
  1. Extract the supplied zip file into a temporary location in your file system.
  2. Highlight the message flow project that you want to use and select **File->Import...**
  3. Select FileSystem and click **Next**.
  4. On the next dialogue box, click **Browse**.
  5. Navigate to the location of the samples in your file system. Select the *Samples* folder and click **OK**.
  6. In the left-hand pane, expand (by clicking the '+' symbol) the *Samples* folder and check folders *Sample1*, *Sample2*, *Sample3*, *Sample4*, and *Sample5*.
  7. Select **Create selected folders only**.
  8. Click **Finish**.

For the Message Set, either use an existing Message Set project or create a new Message Set project:

1. In the *Broker Application Development* perspective, select **File->New->Message Set Project**.
2. Give your project a suitable name (for example, VSAMSamp1esMRM) and click **Finish**.

Import the sample message set:

1. Extract the supplied zip file into a temporary location in your file system.
2. Highlight the Message Set project that you want to use and select **File->Import...**
3. Select FileSystem and click **Next**.
4. On the next dialogue box, click **Browse**.
5. Navigate to the location of the samples in your file system. Select the *Samples* folder and click **OK**.
6. In the left-hand pane, expand (by clicking the '+' symbol) the *Samples* folder and check the folder *VSAMSET1*.
7. Click **Finish**.

Before running the samples, you must deploy the message flows and the message set. Do this in the usual way by adding them to a *.bar* file and then dragging the *.bar* file onto the relevant execution group.

**Note:** The message set and message flows must be deployed to the same execution group.

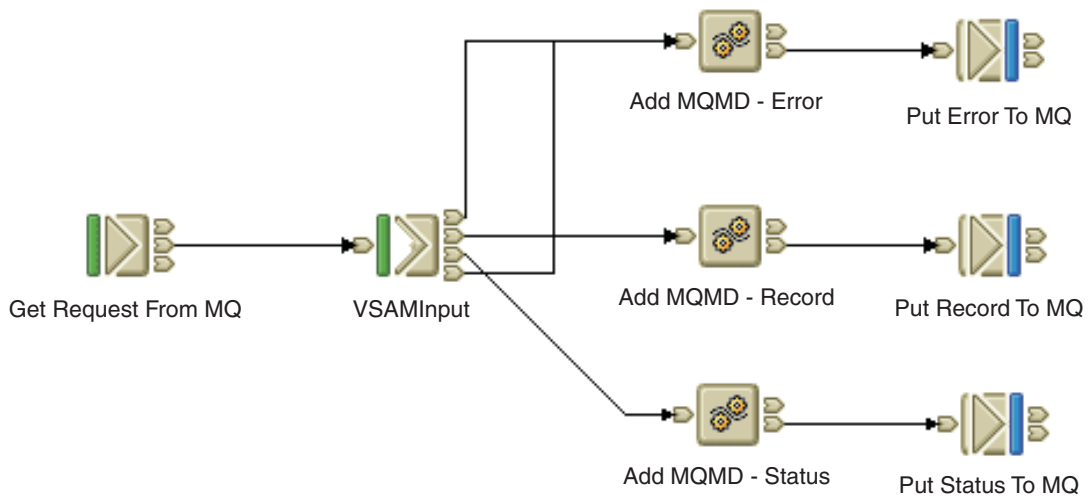
## Sample 1 - Using a VSAMInput node

This sample shows how you can use the VSAMInput node to read all the records in a VSAM data set.

The message flow for this sample does the following tasks:

1. Reads a message from an MQ queue.
2. Reads all the records from a VSAM data set.
3. Sends each record to an MQ queue.
4. Sends a status message to an MQ queue

The message flow that does this is shown in the following diagram:



Failure logic is included in the message flow to handle failures that occur within the VSAMInput node (these are propagated through the Failure terminal) or downstream from the Out terminal (these are propagated through the Catch terminal).

### Developing the flow

Configure the nodes in the message flow so that they have the following properties:

Node name	Node type	Property	Value
Get Request from MQ	MQInput	Queue Name	VSAM.1.IN
Get Request from MQ	MQInput	Order Mode	By Queue Order
VSAMInput	VSAMInput	Default File Name	///HLQ.VSAM.KSDS'
VSAMInput	VSAMInput	Retain File	'Until Idle'

Node name	Node type	Property	Value
Put Error to MQ	MQOutput	Queue Name	VSAM.1.ERROR
Put Error to MQ	MQOutput	Message Context	'Set All'
Put Record to MQ	MQOutput	Queue Name	VSAM.1.OUT
Put Record to MQ	MQOutput	Message Context	'Set All'
Put Status to MQ	MQOutput	Queue Name	VSAM.1.STATUS
Put Status to MQ	MQOutput	Message Context	'Set All'

Create the MQ queues VSAM.1.IN, VSAM.1.ERROR, VSAM.1.OUT, and VSAM.1.STATUS on the broker's queue manager.

Set the *Default File Name* property of the VSAMInput node to `//'HLQ.VSAM.KSDS'`; this determines which VSAM data set is used.

Set the *Retain File* property of the VSAMInput node to `'Until Idle'`; this avoids the overhead of opening and closing the VSAM data set for each read operation. But it might restrict the use of the data set by other applications.

Default values are used for all other properties of the VSAMInput node, but these values are replaced by any values that are specified in the request message `InputBody.VSAM.Request`. The VSAMInput node reads all the records in the VSAM data set that is named, starting with the first record in the data set.

Set the *Order Mode* property of the MQInput node to `Queue Order`. This is important if you deploy more than one instance of the message flow. This prevents a build up of messages coming from the MQInput node before the previous message has been handled by the VSAMInput node.

Because the messages propagated from the VSAMInput node do not come from MQ, an MQMD must be added to each message, and the *Message Context* property of each MQOutput node must be set to `Set All`.

For each of the Compute nodes, make sure that the ESQL matches that given below:

- Add MQMD - Record

```
CREATE COMPUTE MODULE MQMD_RecordCompute
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN

      -- Add the minimum MQMD
      -- Note: this must be the first header of the root tree
      SET OutputRoot.MQMD.StrucID=MQMD_STRUC_ID;
      SET OutputRoot.MQMD.Version=MQMD_CURRENT_VERSION;

      -- Copy the message to the OutputRoot
      -- Note: this copy occurs following the creation of the MQMD
      SET OutputRoot.BLOB=InputRoot.BLOB;

      RETURN TRUE;

  END;
END MODULE;
```

- Add MQMD - Error

```
CREATE COMPUTE MODULE MQMD_ErrorCompute
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN
```



```

-- Add the minimum MQMD
-- Note: this must be the first header of the root tree
SET OutputRoot.MQMD.StrucID=MQMD_STRUC_ID;
SET OutputRoot.MQMD.Version=MQMD_CURRENT_VERSION;

-- Copy the Exception List
SET OutputRoot.XML.VSAM.Exception=InputExceptionList;

RETURN TRUE;

END;
END MODULE;

```

- Add MQMD - Status

```

CREATE COMPUTE MODULE MQMD_StatusCompute
CREATE FUNCTION Main() RETURNS BOOLEAN
BEGIN

-- Add the minimum MQMD
-- Note: this must be the first header of the root tree
SET OutputRoot.MQMD.StrucID=MQMD_STRUC_ID;
SET OutputRoot.MQMD.Version=MQMD_CURRENT_VERSION;

-- Copy the VSAM Status Message
SET OutputRoot.XML.VSAM.Status=InputLocalEnvironment.VSAM.Status;

RETURN TRUE;

END;
END MODULE;

```

Connect the nodes in the message flow by making the following connections:

Node name	Terminal	Connected to
Get Request from MQ	Out	VSAMInput
VSAMInput	Failure	Add MQMD - Error
VSAMInput	Out	Add MQMD - Record
VSAMInput	Status	Add MQMD - Status
VSAMInput	Catch	Add MQMD - Error
Add MQMD - Error	Out	Put Error to MQ
Add MQMD - Record	Out	Put Record to MQ
Add MQMD - Status	Out	Put Status to MQ



---

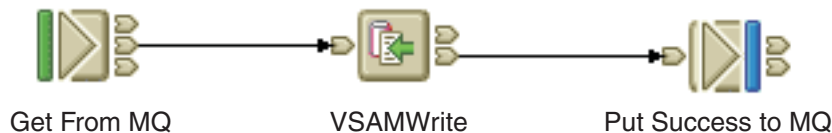
## Sample 2 - Using a VSAMWrite node

This sample shows how you can use the VSAMWrite node to write a record to a VSAM data set.

The message flow for this sample does the following tasks:

1. Reads a message from an MQ queue.
2. Writes part of the message to a VSAM data set.
3. Sends the original message, unchanged to an MQ queue.

The message flow that does this is shown in the following diagram:



### Developing the flow

Configure the nodes in the message flow so that they have the following properties:

Node name	Node type	Property	Value
Get From MQ	MQInput	Queue Name	VSAM.2.IN
VSAMWrite	VSAMWrite	Default File Name	///HLQ.VSAM.KSDS'
VSAMWrite	VSAMWrite	Retain File	'Until Idle'
VSAMWrite	VSAMWrite	Truncate	Selected
Put Success to MQ	MQOutput	Queue Name	VSAM.2.OUT

Create the MQ queues VSAM.2.IN and VSAM.2.OUT on the broker's queue manager.

Define the VSAM data set by setting the *Default File Name* property of the VSAMWrite node to `///HLQ.VSAM.KSDS'`.

Because this message flow does not use the local environment, in the Advanced Group of node properties, set *Generate Mode* to None and *Copy Local Environment* to None.

Because no other applications or Execution Groups will be using this VSAM data set, set the *Retain File* property from the Advanced Group of node properties to *Until Idle*. This means that the VSAM data set is not closed after each record is written to the data set, but remains open.

The VSAM data set is a key-sequenced data set (KSDS); the record length is variable and depends on the amount of data written for each record. Therefore, padding is unnecessary and the *Padding Type* property is ignored. However, select the *Truncate* check box so that only the amount of data determined by the maximum record length for the data set is written.

Because the MRM message set is being used, set the following values for properties in the Default Group:

Property	Value
Message Domain	MRM
Message Set	VSAMSET1
Message Type	msg_SampleStruct
Message Format	CWF1

Use the default values for properties in both the Request Group and the Status Group of node properties; only the message body is written to the VSAM data set and no report message is produced.

Connect the nodes in the message flow by making the following connections:

Node name	Terminal	Connected to
Get From MQ	Out	VSAMWrite
VSAMWrite	Out	Add MQMD - Record
VSAMInput	Status	Put Success to MQ

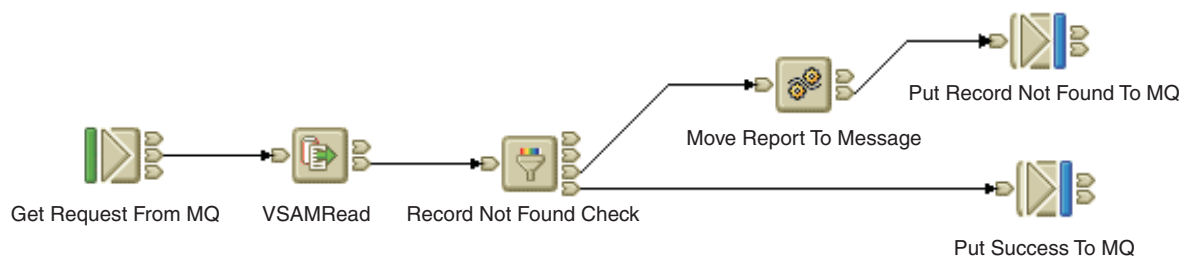
## Sample 3 - Using a VSAMRead node

This sample shows how you can use the VSAMRead node to read a record from a VSAM data set.

The message flow for this sample does the following tasks:

1. Reads a message from an MQ queue; this message contains the VSAM Request that is used by the VSAMRead node.
2. Reads a record from a VSAM data set, and adds the data read to the original message.
3. Checks whether the requested record has been read from the VSAM data set.
4. Sends the updated message to an MQ queue

The message flow that does this is shown in the following diagram:



### Developing the flow

Configure the nodes in the message flow so that they have the following properties:

Node name	Node type	Property	Value
Get Request from MQ	MQInput	Queue Name	VSAM.3.IN
VSAMRead	VSAMRead	Default File Name	///HLQ.VSAM.KSDS'
VSAMRead	VSAMRead	Retain File	'Until Idle'
VSAMRead	VSAMRead	Request Location	InputRoot.XML.VSAM.Request
VSAMRead	VSAMRead	Position Mode	'KEY_EQ'
VSAMRead	VSAMRead	Key Type	'Parser String'
VSAMRead	VSAMRead	Include Record Report	'Yes'
Put Record Not Found to MQ	MQOutput	Queue Name	VSAM.3.RNF
Put Success to MQ	MQOutput	Queue Name	VSAM.3.OUT

Create the MQ queues VSAM.3.IN, VSAM.3.RNF, and VSAM.3.OUT on the broker's queue manager.

Define the VSAM data set by setting the *Default File Name* property of the VSAMRead node to `//'HLQ.VSAM.KSDS'`.

Because no other applications or Execution Groups will be using this VSAM data set, set the *Retain File* property from the Advanced Group of node properties to `Until Idle`. This means that the VSAM data set is not closed after each record is read, but remains open.

Leave all other properties in the Advanced Group to take their default values.

In the Default Group of node properties, set *Message Domain* to `BLOB` and use the default values for all other properties.

In the Request Group of node properties, set *Request Location* to `InputRoot.XML.VSAM.Request`, the location of the VSAM Request message. Because we want to read a specific record, identified by a key, set *Position Mode* to `KEY_EQ` and, because the key is a string, set *Key Type* to `Parser String`. Use the default values for all other properties in the Request Group of node properties.

In the Status Group of node properties, set *Include Record Report* to `Yes`. This causes a VSAM Report message to be created at the default location which is `InputLocalEnvironment.VSAM.Report`. The length of the VSAM Report message is used to identify whether a record has been successfully read; if the required record has not been read, the length of the VSAM Report message is 0.

The Filter Node checks the length of the VSAM Report message. If the length is 0, the message is propagated through the False terminal of the Filter node to a Compute node which copies the VSAM Report message from `LocalEnvironment` to the message body and passes the output message to the MQ queue `VSAM.3.RNF`.

For the Filter node - Record Not Found Check - make sure that the ESQL matches that given below:

```
CREATE FILTER MODULE Record_Not_Found_Check
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN
    -- If the VSAMRead node attempts to read a record, but finds none,
    -- it does not throw an error.
    -- Instead, it reports a Length of 0 in the Report Message.
    -- The user can therefore check for the 'Record Not Found' condition
    -- by checking the value of this field.

    -- To generate a Report Message from the preceding VSAMRead node,
    -- set 'Include Record Report' to 'Yes' in the VSAMRead node Status tab,
    -- and set a value for 'Report Location' that matches the value shown in
    -- the ESQL below.

    IF LocalEnvironment.VSAM.Report.Length = 0 THEN
      RETURN FALSE;
    ELSE
      RETURN TRUE;
    END IF;
  END;
END MODULE;
```

For the Compute node - Move Report To Message - make sure that the ESQL matches that given below:

```

CREATE COMPUTE MODULE MoveReport
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN
    -- Copy the original MQMD
    SET OutputRoot.MQMD = InputRoot.MQMD;

    -- Copy the VSAM REport to the OutputBody
    SET OutputRoot.XML.VSAM.Report = InputLocalEnvironment.VSAM.Report;

    RETURN TRUE;
  END;
END MODULE;

```

Connect the nodes in the message flow by making the following connections:

Node name	Terminal	Connected to
Get Request from MQ	Out	VSAMRead
VSAMRead	Out	Record Not Found Check
Record Not Found Check	True	Put Success To MQ
Record Not Found Check	False	Move Report To Message
Move Report To Message	Out	Put Record Not Found To MQ





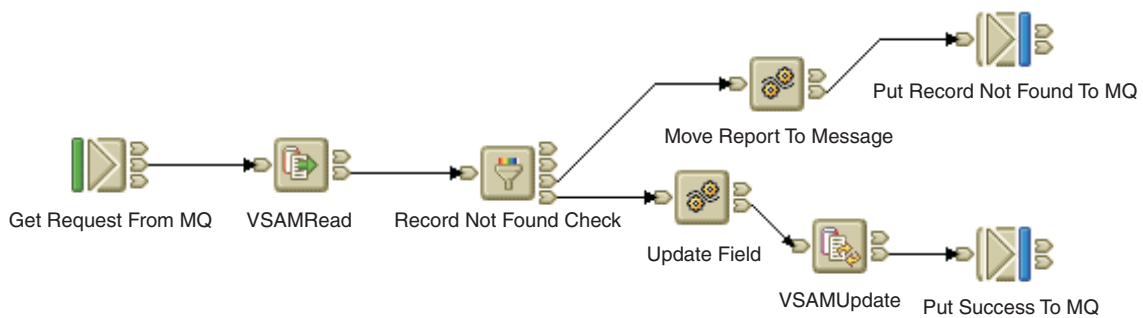
## Sample 4 - Using a VSAMUpdate node

This sample shows how you can use the VSAMRead and VSAMUpdate nodes to update a record in a VSAM data set.

The message flow for this sample does the following tasks:

1. Reads a message from an MQ queue; this message contains a VSAM Request that is used by the VSAMRead node to read a record from a VSAM data set.
2. Confirms that the required record has been read; a Filter node is used to confirm this.
3. Uses a VSAMUpdate node to update the record that has been read and write it back to the VSAM data set.

The message flow that does this is shown in the following diagram:



### Developing the flow

Configure the nodes in the message flow so that they have the following properties:

Node name	Node type	Property	Value
Get Request from MQ	MQInput	Queue Name	VSAM.4.IN
VSAMRead	VSAMRead	Default File Name	///HLQ.VSAM.KSDS'
VSAMRead	VSAMRead	Retain File	'Until Idle'
VSAMRead	VSAMRead	Hold Position	'Downstream'
VSAMRead	VSAMRead	Additional Open Intent	'Update'
VSAMRead	VSAMRead	Request Location	InputRoot.XML.VSAM.Request
VSAMRead	VSAMRead	Position Mode	'KEY_EQ'
VSAMRead	VSAMRead	Key Type	'Parser String'
VSAMRead	VSAMRead	Include Record Report	'Yes'
VSAMUpdate	VSAMUpdate	Default File Name	///HLQ.VSAM.KSDS'
VSAMUpdate	VSAMUpdate	Retain File	'Until Idle'

Node name	Node type	Property	Value
VSAMUpdate	VSAMUpdate	Input Data Location	'InputRoot.MRM'
VSAMUpdate	VSAMUpdate	Position Mode	'CURRENT'
Put Record Not Found to MQ	MQOutput	Queue Name	VSAM.4.RNF
Put Success to MQ	MQOutput	Queue Name	VSAM.4.OUT

Create the MQ queues VSAM.4.IN, VSAM.4.RNF, and VSAM.4.OUT on the broker's queue manager.

Configure the VSAMRead node by setting values for the following node properties:

- Define the VSAM data set by setting the *Default File Name* property of the VSAMRead node to `//'HLQ.VSAM.KSDS'`.
- In the Advanced Group of node properties, set the *Retain File* property to `Until Idle`. This means that the VSAM data set is not closed after each record is read, but remains open.
- Also in the Advanced Group of node properties, set the *Hold Position* property to `Downstream`, and the *Additional Open Intent* property to `Update`. This keeps available, for nodes that are downstream from this VSAMRead node, the position in the VSAM data set of the record that is read, and allows the record to be updated by one of these nodes (the VSAMUpdate node in this message flow) that has a *Position Mode* value set to `CURRENT`.

**Note:** It is important that you set correct values for the *Hold Position* and *Additional Open Intent* properties of the VSAMRead node and the *Position Mode* of the downstream VSAMUpdate node. For example, an error will occur when you try to update a record if you set the *Hold Position* property of the VSAMRead node to `Downstream`, and the *Position Mode* property of the VSAMUpdate node to `CURRENT`, but leave the *Additional Open Intent* property of the VSAMRead node set to `None`.

- Leave all the other Advanced Group properties to take their default values.
- Because the MRM message set is being used, set the following values for properties in the Default Group:

Property	Value
Message Domain	MRM
Message Set	VSAMSET1
Message Type	msg_SampleStruct
Message Format	CWF1

- In the Request Group of node properties, set *Request Location* to `InputRoot.XML.VSAM.Request`, the location of the VSAM Request message. Because we want to read a specific record, identified by a key, set *Position Mode* to `KEY_EQ` and, because the key is a string, set *Key Type* to `Parser String`. Use the default values for all other properties in the Request Group of node properties.
- Leave all the Result Group properties to take their default values. Because MRM is being used in this sample, the output data will be located at `OutputRoot.MRM` in the output message tree.
- In the Status Group of node properties, set *Include Record Report* to `Yes`. This causes a VSAM Report message to be created at the default location which is

InputLocalEnvironment.VSAM.Report. The length of the VSAM Report message is used to identify whether a record has been successfully read; if the required record has not been read, the length of the VSAM Report message is 0.

The Filter Node checks the length of the VSAM Report message. If the length is 0, the message is propagated through the False terminal of the Filter node to a Compute node which copies the VSAM Report message from LocalEnvironment to the message body and passes the output message to the MQ queue VSAM.4.RNF.

For the Filter node - Record Not Found Check - make sure that the ESQL matches that given below:

```
CREATE FILTER MODULE Record_Not_Found_Check
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN
    -- If the VSAMRead node attempts to read a record, but finds none,
    -- it does not throw an error.
    -- Instead, it reports a Length of 0 in the Report Message.
    -- The user can therefore check for the 'Record Not Found' condition
    -- by checking the value of this field.

    -- To generate a Report Message from the preceding VSAMRead node,
    -- set 'Include Record Report' to 'Yes' in the VSAMRead node Status tab,
    -- and set a value for 'Report Location' that matches the value shown in
    -- the ESQL below.

    IF LocalEnvironment.VSAM.Report.Length = 0 THEN
      RETURN FALSE;
    ELSE
      RETURN TRUE;
    END IF;
  END;
END MODULE;
```

For the Compute node - Move Report To Message - make sure that the ESQL matches that given below:

```
CREATE COMPUTE MODULE MoveReport
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN
    -- Copy the original MQMD
    SET OutputRoot.MQMD = InputRoot.MQMD;

    -- Copy the VSAM REport to the OutputBody
    SET OutputRoot.XML.VSAM.Report = InputLocalEnvironment.VSAM.Report;

    RETURN TRUE;
  END;
END MODULE;
```

For the Compute node - Update Field - make sure that the ESQL matches that given below:

```
CREATE COMPUTE MODULE UpdateField
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN
    SET OutputRoot = InputRoot;

    -- The record format in our MRM message is:
    -- intField1 (4-byte integer)
    -- intField2 (4-byte integer)
    -- charField1 (8-byte string)
    -- charField2 (8-byte string)

    -- Update a value in the MRM message
```

```
-- Our changed message tree is then passed into the following
-- VSAMUpdate node where we attempt to update our record.
```

```
SET OutputRoot.MRM.charField2 = 'UPDATED FIELD...';

RETURN TRUE;

END;
END MODULE;
```

Configure the VSAMUpdate node by setting values for the following node properties:

- Define the VSAM data set by setting the *Default File Name* property of the VSAMUpdate node to `//'HLQ.VSAM.KSDS'`. This is the same data set that is used by the VSAMRead node in the message flow.
- In the Advanced Group of node properties, set the *Retain File* property to `Until Idle`. This means that the VSAM data set is not closed after a record is updated, but remains open.
- Leave all the other Advanced Group properties to take their default values.
- Because the MRM message set is being used, set the following values for properties in the Default Group:

Property	Value
Message Domain	MRM
Message Set	VSAMSET1
Message Type	msg_SampleStruct
Message Format	CWF1

- In the Request Group of node properties:
  - Set the *Position Mode* property to `CURRENT`.
  - Set *Input Data Location* to `InputRoot.MRM`; this is the location in the message tree of the data that we want to write to the VSAM data set.
  - Accept default values for all other Request Group properties.
- Accept default values for all Status Group properties.

Connect the nodes in the message flow by making the following connections:

Node name	Terminal	Connected to
Get Request from MQ	Out	VSAMRead
VSAMRead	Out	Record Not Found Check
Record Not Found Check	True	Update Field
Update Field	Out	VSAMUpdate
VSAMUpdate	Out	Put Success to MQ
Record Not Found Check	False	Move Report To Message
Move Report To Message	Out	Put Record Not Found To MQ

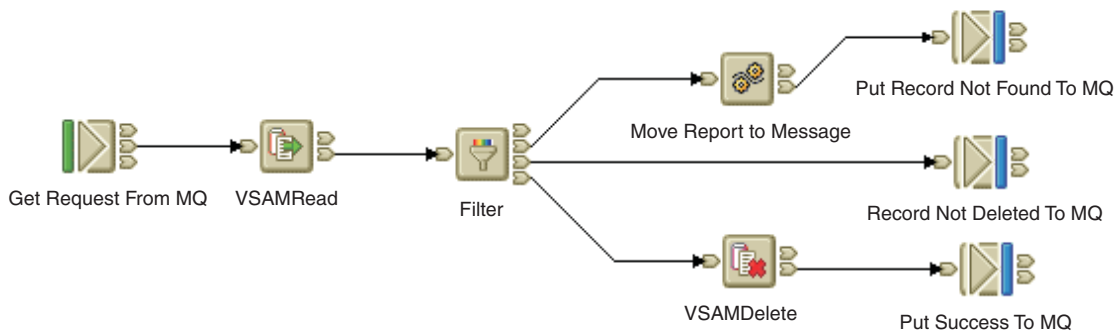
## Sample 5 - Using a VSAMDelete node

This sample shows how you can use the VSAMDelete to delete a record from a VSAM data set.

The message flow for this sample does the following tasks:

1. Reads a message from an MQ queue; this message contains a VSAM Request that is used by the VSAMRead node to read a record from a VSAM data set.
2. Confirms that the required record has been read.
3. Checks the value of a field in the record that was read.
4. Uses a VSAMDelete node to delete the record from the VSAM data set if the value of the field was negative.
5. Puts a message onto an MQ queue.

The message flow that does this is shown in the following diagram:



### Developing the flow

Configure the nodes in the message flow so that they have the following properties:

Node name	Node type	Property	Value
Get Request from MQ	MQInput	Queue Name	VSAM.5.IN
VSAMRead	VSAMRead	Default File Name	///HLQ.VSAM.KSDS'
VSAMRead	VSAMRead	Retain File	'Until Idle'
VSAMRead	VSAMRead	Hold Position	'Downstream'
VSAMRead	VSAMRead	Additional Open Intent	'Update'
VSAMRead	VSAMRead	Request Location	InputRoot.XML.VSAM.Request
VSAMRead	VSAMRead	Position Mode	'KEY_EQ'
VSAMRead	VSAMRead	Include Record Report	'Yes'
VSAMDelete	VSAMDelete	Default File Name	///HLQ.VSAM.KSDS'
VSAMDelete	VSAMDelete	Retain File	'Until Idle'
VSAMDelete	VSAMDelete	Position Mode	'CURRENT'

Node name	Node type	Property	Value
Put Record Not Found to MQ	MQOutput	Queue Name	VSAM.5.RNF
Record Not Deleted to MQ	MQOutput	Queue Name	VSAM.5.NOTDLTD
Put Success to MQ	MQOutput	Queue Name	VSAM.5.OUT

Create the MQ queues VSAM.5.IN, VSAM.5.RNF, VSAM.5.NOTDLTD, and VSAM.5.OUT on the broker's queue manager.

Configure the VSAMRead node by setting values for the following node properties:

- Define the VSAM data set by setting the *Default File Name* property of the VSAMRead node to `//'HLQ.VSAM.KSDS'`.
- In the Advanced Group of node properties, set the *Retain File* property to `Until Idle`. This means that the VSAM data set is not closed after each record is read, but remains open.
- Also in the Advanced Group of node properties, set the *Hold Position* property to `Downstream`, and the *Additional Open Intent* property to `Update`. This keeps available, for nodes that are downstream from this VSAMRead node, the position in the VSAM data set of the record that is read, and allows the VSAM data set to be updated by one of these nodes (the VSAMDelete node in this message flow) that has a *Position Mode* value set to `CURRENT`.

**Note:** It is important that you set correct values for the *Hold Position* and *Additional Open Intent* properties of the VSAMRead node and the *Position Mode* of the downstream VSAMDelete node. For example, an error will occur when you try to delete a record if you set the *Hold Position* property of the VSAMRead node to `Downstream`, and the *Position Mode* property of the VSAMDelete node to `CURRENT`, but leave the *Additional Open Intent* property of the VSAMRead node set to `None`.

- Leave all the other Advanced Group properties to take their default values.
- Because the MRM message set is being used, set the following values for properties in the Default Group:

Property	Value
Message Domain	MRM
Message Set	VSAMSET1
Message Type	msg_SampleStruct
Message Format	CWF1

- In the Request Group of node properties, set *Request Location* to `InputRoot.XML.VSAM.Request`, the location of the VSAM Request message. Because we want to read a specific record, identified by a key, set *Position Mode* to `KEY_EQ` and, because the key is a string, set *Key Type* to `Parser String`. Use the default values for all other properties in the Request Group of node properties.
- In the Status Group of node properties, set *Include Record Report* to `Yes`. This causes a VSAM Report message to be created at the default location which is `InputLocalEnvironment.VSAM.Report`. The length of the VSAM Report message is used to identify whether a record has been successfully read; if the required record has not been read, the length of the VSAM Report message is 0.

The Filter Node checks the length of the VSAM Report message. If the length is 0, the message is propagated through the False terminal of the Filter node to a Compute node which copies the VSAM Report message from LocalEnvironment to the message body and passes the output message to the MQ queue VSAM.4.RNF.

The Filter Node also checks the value of a field (Root.MRM.intField2) in the record read by the VSAMRead node; if the value is negative a message is passed to the VSAMDelete node; otherwise the record is added to the VSAM.5.NOTDLTD MQ queue.

For the Filter node - Filter - make sure that the ESQL matches that given below:

```
CREATE FILTER MODULE ReportFilter
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN
    -- Check for record not found
    IF LocalEnvironment.VSAM.Report.Length = 0 THEN
      RETURN UNKNOWN;
    ELSEIF Root.MRM.intField2 < 0 THEN
      RETURN TRUE;
    ELSE
      RETURN FALSE;
    END IF;
  END;
END MODULE;
```

For the Compute node - Move Report To Message - make sure that the ESQL matches that given below:

```
CREATE COMPUTE MODULE MoveReport
  CREATE FUNCTION Main() RETURNS BOOLEAN
  BEGIN
    -- Copy the VSAM Report to OutputBody
    SET OutputRoot.XML.VSAM.Report = InputLocalEnvironment.VSAM.Report;
    RETURN TRUE;
  END;
END MODULE;
```

Configure the VSAMDelete node by setting values for the following node properties:

- Define the VSAM data set by setting the *Default File Name* property of the VSAMDelete node to `//'HLQ.VSAM.KSDS'`. This is the same data set that is used by the VSAMRead node in the message flow.
- In the Advanced Group of node properties, set the *Retain File* property to `Until Idle`. This means that the VSAM data set is not closed after each record is deleted, but remains open.
- Leave all the other node properties to take their default values.

Connect the nodes in the message flow by making the following connections:

Node name	Terminal	Connected to
Get Request from MQ	Out	VSAMRead
VSAMRead	Out	Filter
Filter	True	Update Field
Update Field	Out	VSAMDelete
VSAMDelete	Out	Put Success to MQ

<b>Node name</b>	<b>Terminal</b>	<b>Connected to</b>
Filter	False	Record Not Deleted To MQ
Filter	Unknown	Move Report To Message
Move Report To Message	Out	Put Record Not Found To MQ



## Error messages and error codes

Error messages that might be returned are shown in the following table:

Message number	Text	Explanation	User Action
BIP4541E	An unknown value was found in the VSAM request message.	Value "{0}" is invalid for VSAMRequest element "{1}". Valid values are "{2}".	Examine the source of the VSAMRequest message to determine why this value was specified.
BIP4542I	Request message found.	A request message was found by node "{0}" at "{1}". Any control settings in this request message will override those set as node attributes.	No user action required.
BIP4543I	No Request message found.	Node "{0}" received a message assembly but no VSAM Request message was found at the ESQL PATH expression "{1}". The operation will be completely defined by node attribute settings.	No user action required.
BIP4544I	Request for file operation.	Node "{0}" has been requested to perform an operation on file "{1}". The parameters for this operation are: Action="{2}". Key="{3}". KeyType="{4}". RBA="{5}". RRN="{6}". Position Mode="{7}". CCSID="{8}".	No user action required.
BIP4545I	Record not found.	Node "{0}" has been requested to perform an operation on file "{1}". The record was not found within this file using the specified parameters.	No user action required. Failure to find a record is not treated as an error condition. If a VSAM node successfully opens the file but fails to locate a record using the KEY, RBA, or RRN that was specified by the user, a message is still propagated to the Out terminal. For the VSAMRead node, this message has an empty tree under the path specified in <i>Output Data Location</i> . Also, the Report message does not contain a <i>Position</i> child and <i>Length</i> has value 0.
BIP4546I	Node (class={0}, name={1}) propagating to {2} terminal.	The specified node is about to propagate a message to the specified terminal.	No user action required. This is an information message.

Message number	Text	Explanation	User Action
BIP4547E	Invalid configuration message containing attribute value "{1}" which is not valid for target attribute "{0}", object "{3}"; valid values are "{2}".	The message broker received a configuration message containing the attribute value "{1}" which is not valid for the target attribute "{0}", on object "{3}". Valid values are "{2}". This can be caused by a mismatch in levels between the Message Brokers Toolkit, the Configuration Manager and the Broker, or as a result of a user or third party written plug-in node where the implementation library installed at the broker does not match the node definition held at the Configuration manager and Message Brokers Toolkit.	Ensure that the levels of code installed at the Message Brokers Toolkit, Configuration Manager and Broker are all consistent. If they are, identify the supplier of the target object and report the problem to them. If this is IBM, contact your IBM support center.
BIP4548E	Error casting character string "{0}" to an integer.	An attempt was made to cast the character string "{0}" to an integer, but the string was of the wrong format.	Other messages will give an indication of context of this error.
BIP4549E	Message broker internal error: diagnostic information "{0}".	An internal software error has occurred in the message broker. Further messages will indicate the effect of this error on the broker's transactions. The diagnostic information associated with this message is: "{0}".	Shutdown and restart the message broker. If the problem continues to occur, then restart the system. If the problem still continues to occur contact your IBM support center.
BIP4550I	Parsing record read from VSAM.	Node "{0}" has read a VSAM record and is attempting to parse using: Domain="{1}", MessageSet="{2}", MessageType="{3}", MessageFormat="{4}", CCSID="{5}", Encoding="{6}".	No user action required.
BIP4551E	Unable to interpret key.	Node "{0}" was unable to interpret key value "{1}" using keyType "{2}".	No user action required.
BIP4552E	Unable to determine queue manager's CCSID.	A problem occurred while trying to obtain the queue manager's CCSID.	No user action required.
BIP4553W	MQ call failed.	An unexpected return code was received when issuing "{0}", on queue manager "{1}". Return code "{2}", reason code "{3}".	No user action required.
BIP4554E	Position Mode of CURRENT was specified without previous node holding the position.	Node "{0}" attempted to read a record from file "{1}" using Position Mode CURRENT but current position has not been held.	Ensure that an upstream node has the Hold Position property set to Downstream or change the Position Mode from CURRENT.

Message number	Text	Explanation	User Action
BIP4555E	Unknown code page specified.	An attempt was made to convert the string "{0}" from unicode using CCSID "{1}" but this does not correspond to any known code page.	Review the CCSID used and ensure it corresponds to a known code page.
BIP4556E	Failed to navigate to ESQL path "{0}" in node "{2}".	An attempt was made to navigate to ESQL path, "{0}" , given a Generate Mode of "{1}" but it was not possible to navigate to this location.	Review the node attribute settings and ensure that the ESQL is navigable.
BIP4557E	Unable to interpret key.	Node "{0}" was unable to interpret padding value "{1}" using paddingType "{2}".	No user action required.
BIP4558E	No value was found for an element within the VSAMRequest message.	VSAM Request element "{0}" does not contain a value.	Examine the source of the VSAMRequest message to determine why this element is missing a value.
BIP4559E	An invalid value was found in the VSAM request message.	The value type found "{0}" is invalid for VSAMRequest element "{1}" valid types are "{2}".	Examine the source of the VSAMRequest message to determine why the value is of the type specified.
BIP4560I	Exception detected.	A component of the VSAM nodes has detected an exception with message number "{0}" and inserts "{1}". File:{2}, Line:{3}, Function:{4}, TraceText:{5},	The usual exception handling mechanisms will determine if this exception was dealt with internally or if another exception was thrown in its place.
BIP4561I	Exception detected.	A component of the VSAM nodes has detected an exception. The following message will contain details of the exception. File:{2}, Line:{3}, Function:{4}, TraceText:{5},	The usual exception handling mechanisms will determine if this exception was dealt with internally or if another exception was thrown in its place.
BIP4562E	Failed to navigate to ESQL path "{0}" in node "{1}".	An attempt was made to navigate to ESQL path, "{0}" , but it was not possible to navigate to this location.	Review the node attribute settings and ensure that the ESQL is navigable.
BIP4563I	Request for file operation.	Node "{0}" has been requested to perform an operation on file "{1}". The parameters for this operation have been gleaned from the Request message, if found, and node attributes. Action="{2}". RRN="{3}". CCSID="{4}".	No user action required.
BIP4564E	Message broker internal error: diagnostic information "{0}".	An internal software error has occurred in the VSAMNodes. A core dump has been written to the broker's started task user's home directory. The execution group will now be restarted. Information associated with this message is: "{0}".	If the problem continues to occur, contact your IBM support center.

Message number	Text	Explanation	User Action
BIP4570E	Message broker internal file error: file name "{0}".	An internal software error has occurred in the message broker whilst processing the following file "{0}". Further messages will indicate the effect of this error on the broker's transactions.	Shutdown and restart the message broker. If the problem continues contact your IBM support center.
BIP4571E	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.	Unable to open file "{0}" for the reason given by the system error message "{1}". The following information may help to diagnose the problem: __last_op: "{2}". __rc: "{3}". __fdbk: "{4}". Further messages will indicate the effect of this error on the broker's transactions.	Failed to open file: file name "{0}"; system error message "{1}".
BIP4572E	Failed to close file: file name "{0}"; system error message "{1}".	Unable to close file "{0}" for the reason given by the system error message "{1}". The following information may help to diagnose the problem: __last_op: "{2}". __rc: "{3}". __fdbk: "{4}". Further messages will indicate the effect of this error on the broker's transactions.	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4573E	Failed to retrieve file information: file name "{0}"; system error message "{1}".	Unable to retrieve file information for "{0}" for the reason given by the system error message "{1}". The following information may help to diagnose the problem: __last_op: "{2}". __rc: "{3}". __fdbk: "{4}". Further messages will indicate the effect of this error on the broker's transactions.	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4574E	Failed to locate record: file name "{0}"; system error message "{1}".	A call to locate a record has failed while processing file "{0}", for the reason given by the system error message "{1}". Further messages will indicate the effect of this error on the broker's transactions. This failure does not include the simple case where no record has been found.	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.

Message number	Text	Explanation	User Action
BIP4575E	Failed to read a record: file name "{0}"; system error message "{1}".	A call to read a record has failed while processing file "{0}", for the reason given by the system error message "{1}". The following information may help to diagnose the problem: __last_op: "{2}". __rc: "{3}". __fdbk: "{4}". Further messages will indicate the effect of this error on the broker's transactions.	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4576E	Attempted to update a record in file "{0}" with data larger than the maximum record length.	An attempt was made to update a record in file "{0}" with the truncate option set to false. The maximum record length for this file is "{1}" but the length of the data specified was "{2}".	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4577E	Attempted to update a record in file "{0}" with data larger than the existing record length.	An attempt was made to update a record in file "{0}" with the truncate option set to false. This file does not allow changing the length of the record. The existing record length is "{1}" but the length of the data specified was "{2}".	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4578E	Failed to update a record: file name "{0}"; system error message "{1}".	A call to update a record has failed while processing file "{0}", for the reason given by the system error message "{1}". The following information may help to diagnose the problem: __last_op: "{2}". __rc: "{3}". __fdbk: "{4}". Further messages will indicate the effect of this error on the broker's transactions.	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4579E	Attempted to write a record in file "{0}" with data larger than the maximum record length.	An attempt was made to write a record in file "{0}" with the truncate option set to false. The maximum record length for this file is "{1}" but the length of the data specified was "{2}".	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4580E	Unsupported file type: file name "{0}".	An attempt was made to process an unsupported file type whilst processing the following file "{0}". Further messages will indicate the effect of this error on the broker's transactions.	Supply a valid file name to this node via the Default File name property or overriding request message.

Message number	Text	Explanation	User Action
BIP4581E	Invalid position mode for file "{0}".	An attempt was made to locate a record within file "{0}" which is of type "{1}". A position mode of "{2}" was provided which is not valid for this file type.	Supply a position mode which is valid for this file type.
BIP4582E	Failed to write a record: file name "{0}"; system error message "{1}".	A call to write a record has failed while processing file "{0}", for the reason given by the system error message "{1}". The following information may help to diagnose the problem: __last_op: "{2}" __rc: "{3}" __fdbk: "{4}". Further messages will indicate the effect of this error on the broker's transactions.	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4583E	Unsupported file type for delete: file name "{0}".	An attempt was made to delete a record on an unsupported (ESDS or ESDS_PATH) file type whilst processing the following file "{0}". Further messages will indicate the effect of this error on the broker's transactions.	Supply a valid file to this node via the Default File name property or overriding request message.
BIP4584E	Failed to delete a record: file name "{0}"; system error message "{1}".	A call to delete a record has failed while processing file "{0}", for the reason given by the system error message "{1}". The following information may help to diagnose the problem: __last_op: "{2}" __rc: "{3}" __fdbk: "{4}". Further messages will indicate the effect of this error on the broker's transactions.	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4585E	Attempted to write a new record in file "{0}" with data shorter than the fixed record length.	An attempt was made to write a new record in file "{0}" without any padding specified. The fixed record length for this file is "{1}" but the length of the data specified was "{2}".	Use the file name and system error message to determine the cause of this error. The system log may contain additional messages related to this error.
BIP4586E	Key supplied too long for file "{0}". Supplied key in hexadecimal "{1}".	An attempt was made to position to a record using a key that was too long for file "{0}". The file operation for which this position was attempted will not take place.	Use the file name and key to determine the cause of this error. The system log may contain additional messages related to this error.

Message number	Text	Explanation	User Action
BIP4999E	Message broker internal error: diagnostic information "{0}".	An internal software error has occurred in the message broker. Further messages will indicate the effect of this error on the broker's transactions. The diagnostic information associated with this message is: "{0}".	Shutdown and restart the message broker. If the problem continues, restart your system. If the problem continues to occur, contact your IBM support center.





---

## Dependencies

The minimum levels of software that are required to support the VSAM nodes are:

- IBM® z/OS Version 1.4.
- WebSphere Business Integration Message Broker Version for z/OS V5.0, Fix Pack 3.



---

## Related information

For an introduction to VSAM, you might want to read the first chapter of the RedBook *VSAM Demystified*.

You can obtain *VSAM Demystified*, as HTML or as a PDF, from the following websites:

**<http://www.redbooks.ibm.com/redbooks/SG246105.html>**

**<http://www.redbooks.ibm.com/redbooks/pdfs/sg246105.pdf>**

Details of the commands that you need to create and write to VSAM files are in the IBM manual *z/OS Version 1 Release 3, DFSMS Access Method Services for Catalogs, SC26-7394-01*, which is available from the following website:

**<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>**

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

**<http://www.ibm.com/software/integration/wbimessagebroker/library>**



---

## 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, 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 United Kingdom Laboratories,
- Mail Point 151,
- Hursley Park,
- Winchester,
- Hampshire,
- England
- SO21 2JN

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.

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.

This information includes 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. 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 IBM's application programming interfaces.

---

## Appendix B. Trademarks

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

AIX	CICS	DB2
DB2 Universal Database	developerWorks	
Everyplace	FFST	First Failure Support Technology
IBM	IMS	IMS/ESA
iSeries	Language Environment	MQSeries
MVS	NetView	OS/400
OS/390	pSeries	RACF
RETAIN	RS/6000	SupportPac
Tivoli	VisualAge	WebSphere
xSeries	z/OS	zSeries

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. 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.

Pentium is a registered trademark of Intel.

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

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









Printed in USA