



Active Messaging 用户指南

Adaptive Server® Enterprise

15.7

文档 ID: DC01278-01-1570-01

最后修订日期: 2011 年 9 月

版权所有 © 2011 by Sybase, Inc. 保留所有权利。

本出版物适用于 Sybase 软件及所有后续版本, 除非在新版本或技术说明中另有说明。此文档中的信息如有更改, 恕不另行通知。此处说明的软件按许可协议提供, 其使用和复制必须符合该协议的条款。

若要订购附加文档, 美国和加拿大的客户请拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

持有美国许可协议的其它国家 / 地区的客户可通过上述传真号码与客户服务部门联系。所有其他国际客户请与 Sybase 子公司或当地分销商联系。仅在定期安排的软件发布日期提供升级。未经 Sybase, Inc. 的事先书面许可, 本书的任何部分不得以任何形式、任何手段 (电子的、机械的、手动、光学的或其它手段) 进行复制、传播或翻译。

可从 Sybase 商标页中查看 Sybase 商标, 网址为 <http://www.sybase.com/detail?id=1011207>。Sybase 和文中列出的标记均是 Sybase, Inc. 的商标。® 表示已在美国注册。

SAP 和此处提及的其它 SAP 产品与服务及其各自的徽标是 SAP AG 在德国和世界各地其它几个国家 / 地区的商标或注册商标。

Java 和所有基于 Java 的标记都是 Sun Microsystems, Inc. 在美国和其它国家 / 地区的商标或注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 的注册商标。

IBM 和 Tivoli 是 International Business Machines Corporation 在美国和 / 或其它国家 / 地区的注册商标。

提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568。

目录

第 1 章	简介	1
	Active Messaging 的一些概念	1
	实时地自动做出决策	2
	消息传送模型	2
	JMS	3
	WebSphere MQ 消息传送模型	3
	消息格式	4
	JMS 消息属性	4
	MQ 消息主题	4
	消息选择器	4
第 2 章	了解 Active Messaging	5
	向队列发送消息和从队列中接收消息	5
	向 JMS 主题发布消息以及使用其中的消息	6
	使用消息属性	6
	预览消息传送接口	7
	MQ 概述	9
	使用 SSL 保护通道安全	12
	MQ 发布 / 预订	13
	主题的语法	15
	发布者和预订者的标识	17
	MQ 发布 / 预订示例	17
	使用 MQ 集群队列对象	24
	使用远程队列对象	25
	使用文本消息传送	26
	文本消息和 JMS	26
	文本消息和 MQ	26
	Adaptive Server Cluster Edition 支持	27
	登录重定向	28
	扩展的高可用性	29
	对线程化内核的 Active Messaging 支持	29
	国际化支持	30
	事务型消息行为	31
	事务型消息传送的 set 选项	32

	MQ 安全性	33
	连接到 MQ 队列管理器	33
	在 Adaptive Server 主机上安装 MQ 客户端	33
	MQ 授权	34
	查询 MQ 信息	35
第 3 章	SQL 参考	37
	与消息相关的全局变量	38
	<msgheader> 和 <msgproperties> 文档	44
	特定于 Adaptive Server 的消息属性	46
	关键字	47
	存储过程	47
	内置函数	48
	语法段	48
	sp_configure 'enable real time messaging'	49
	sp_engine	53
	sp_msgadmin	57
	msgconsume	68
	msgpropcount	71
	msgproplist	72
	msgpropname	73
	msgproptype	74
	msgpropvalue	76
	msgpublish	77
	msgrecv	80
	msgsend	93
	msgsubscribe	122
	msgunsubscribe	123
	endpoint	125
	option_string	128
	sizespec	129
	timespec	130
第 4 章	示例	133
	Sybase 目录	133
	使用 SQL 代码示例	134
	使用 Java/JDBC 代码示例	134
	词汇表	135
	索引	139

简介

尽管本手册假定您已基本了解用于数据库管理的消息传送系统，但本章还是要介绍一些基本的消息概念和模型，并对在本文档中贯穿使用的部分术语进行解释。

大部分讨论涉及的都是特定于 Adaptive Server[®] 的消息传送方面。在本文档中将此功能称作实时消息传送服务。

主题	页码
Active Messaging 的一些概念	1
实时地自动做出决策	2
消息传送模型	2
消息格式	4
消息选择器	4

Active Messaging 的一些概念

消息传送是指两个或更多软件应用程序之间进行的信息交换。消息是独立的信息包。

很多 Adaptive Server 客户都在自己的应用程序环境中使用消息传送和排队系统或发布和预订系统。这些应用程序称作面向消息的中间件。通常，同一应用程序会将数据库操作与消息传送操作结合起来。

Active Messaging 通过将 Adaptive Server 与 Tibco Enterprise Message Service (EMS)、EAServer Java Messaging System (JMS)、IBM WebSphere MQ (MQ) 和 Sonic Software 开发的 SonicMQ 结合使用，简化了此类应用程序的开发。

注释 EMS 是 Tibco 实现的 Java 消息传送服务 (JMS)。除非明确说明，否则在本文档中 JMS 泛指 Tibco EMS、EAServer JMS 和 SonicMQ JMS。

消息传送系统允许发送方与接收方分离。由于应用程序发送消息时不要求接收应用程序处于运行状态，因此消息传送系统可以是异步的。

JMS 和 MQ 是定义客户端如何与消息提供程序通信的 API。消息发送方和消息接收方都担当消息提供程序的客户端。

消息传送系统由消息提供程序提供。消息提供程序可以实现集中存储消息的体系结构或分散存储消息的体系结构，也可以实现结合运用这两种存储方式的体系结构。

Active Messaging 使用内置函数在 SQL 语句内执行消息传送操作。

通过它还可以捕获 Adaptive Server 数据库中的事务（数据更改），并使用以下任一项将这些事务作为事件传递给外部应用程序：

- 由 Tibco、EAServer 和 Sonic Software 提供的 JMS 消息总线，或
- 由 WebSphere MQ 提供的消息队列接口 (MQI)

实时地自动做出决策

在管理数据库的过程中，有时必须允许实时地自动做出决策来响应特定事件。所谓实时，是指数据库可以在事件发生时做出有关事件的决策，而不是仅仅将事件排入队列。必须将事件（例如记录中发生的更改）与其它更改一起评估，并选择最高效的响应。也就是说，有效的决策支持系统需要：

- 具备低延迟的特点，从而实现实时企业
- 描述事件及其相关数据的自动化系统
- 可降低应用程序成本并实现低延迟的技术

Active Messaging 使用 Tibco 或 EAServer JMS 消息总线或者 IBM WebSphere MQ 来满足这些业务需求。

消息传送模型

本节介绍 JMS 和 MQ 的消息传送模型。

JMS

JMS 定义了以下两个消息传送模型：

- 发布和预订（主题）
- 点对点（队列）

发布和预订（主题）

发布和预订（发布 / 预订）模型属于一对多模型。发送消息的应用程序称作“消息生产者”，接收消息的应用程序称作“消息使用者”。消息生产者通过建立预订进行注册，以表示其对发送到某一主题的消息感兴趣。主题是此消息模型的目标。

在发布 / 预订模型中可以建立以下两种类型的预订：

- **永久性** — 即使消息使用者应用程序未处于连接状态，也会为消息使用者保留消息。消息由消息提供程序保留，而非由 Adaptive Server 保留。
- **非永久性** — 仅在使用者应用程序与消息提供程序保持连接时才保留消息。

点对点（队列）

点对点模型属于一对一模型，原因是一个应用程序（称作“消息发送方”）发送的任何消息都只能由一个接收应用程序（称作“消息接收方”）读取。点对点消息的目标是队列。一个队列可能包含多个活动消息接收方，但消息传送提供程序会确保消息仅传递到一个消息接收方。

WebSphere MQ 消息传送模型

所有 MQ 消息传送模型均为点对点模型，也就是说，消息的发送目标或接收来源始终都是由队列管理器管理的队列。

MQ 发布 / 预订是建立在 MQ 队列基础之上的发布和预订模型；相应的消息并非不同类型的对象。与 MQ 发布 / 预订的交互使用 MQ 队列进行。

所有消息都发送至 MQ 发布 / 预订代理的代理命令队列。这包括发布者或预订者的注册消息，以及一些控制消息（例如删除某一消息或请求获得某一消息的更新）。

发布者将发布发送到流队列。MQ 发布 / 预订代理将消息分发到对该消息感兴趣的所有预订者。发布者使用主题来描述消息，主题即描述消息内容的主题。

预订者通过指定一个或多个感兴趣的主题进行注册，以表示其对发送到指定流队列的消息感兴趣。此类消息发送到流队列后，MQ 发布 / 预订代理会将这些消息复制到预订者在注册时指定的本地队列。

消息格式

MQ 和 JMS 的消息格式均包括：

- 消息标头 — 包含由标准规定的固定大小的信息部分和可变大小的信息部分。大部分此类信息都是由消息提供程序自动指定的。
- 消息正文 — 即客户端应用程序交换的应用程序数据。

JMS 定义结构化的消息类型（如 `stream` 和 `map`）以及非结构化的消息类型（例如 `text`、`byte` 和 `object`）。

在 MQ 中，消息正文可以同时包含文本数据和二进制数据。

JMS 消息属性

在 Tibco、EAServer 和 Sonic MQ 消息中，属性是用户定义的属性，可以纳入到消息中。有些类型的消息属性定义了特定于应用程序的信息，消息使用者以后可以使用这些信息来选择其感兴趣的`消息`。消息属性类型为 Java 本机类型 `int`、`float` 或 `String`（类）。

MQ 消息主题

在 MQ 中，发布 / 预订模型允许使用“主题”，即消息的主题。主题包含在消息中的规则和格式设置 (RF) 标头中。与 JMS 不同的是，MQ 主题并非由名称及其随附值组成的名称 — 值对，而是用来描述 MQ 发布 / 预订消息的自由形式字符串。

消息选择器

消息使用者通过 Tibco 和 EAServer 的 JMS 消息选择器可以对消息流进行过滤，选择其感兴趣的`消息`。这些过滤器会应用引用消息属性及其值的条件。消息选择器是一种 SQL-92 `where` 子句。

MQ 消息选择功能仅使用消息 ID 和消息相关 ID 作为消息选择器。消息读取者可以通过指定消息 ID 或消息相关 ID，有选择地读取特定`消息`。

了解 Active Messaging

本章对特定于 Adaptive Server® 的 Active Messaging 进行了概述，Active Messaging 允许使用 Adaptive Server 作为消息提供程序的客户端。可以使用 Transact-SQL™ 向消息传送提供程序发送消息，或从消息传送提供程序中检索消息。

主题	页码
向队列发送消息和从队列中接收消息	5
向 JMS 主题发布消息以及使用其中的消息	6
使用消息属性	6
预览消息传送接口	7
MQ 概述	9
使用 SSL 保护通道安全	12
MQ 发布 / 预订	13
使用 MQ 集群队列对象	24
使用远程队列对象	25
使用文本消息传送	26
Adaptive Server Cluster Edition 支持	27
国际化支持	30
事务型消息行为	31
MQ 安全性	33
在 Adaptive Server 主机上安装 MQ 客户端	33
MQ 授权	34
查询 MQ 信息	35

向队列发送消息和从队列中接收消息

利用内置函数 `msgsend` 和 `msgrecv`，Transact-SQL 应用程序可以向队列发送消息，也可以从 JMS 和 MQ 中的队列中读取消息。

可以使用应用程序逻辑来构造消息正文（即有效负载），也可以直接使用来自关系表的 `character` 或 `binary` 数据构造正文。

可以用关系数据或应用程序逻辑来构造消息属性（标头或用户属性）的值，并且可以将所构造的消息属性纳入到要发送的消息中。

从 JMS 或 MQ 队列中读取的消息可以由应用程序逻辑加以处理，也可以直接插入到关系表中。要在执行读取操作时仅过滤出感兴趣的消息，请指定消息选择器。

所读取的消息中的消息属性可由应用程序逻辑单独处理。有关消息属性的详细信息，请参见第 93 页的“msgsend”。

向 JMS 主题发布消息以及使用其中的消息

利用内置函数 `msgpublish` 和 `msgconsume`，Transact-SQL 应用程序可以向 JMS 主题发布消息或使用其中的消息。

首先，必须使用 `sp_msgadmin 'register'` 注册预订。注册预订时会创建一个可供 `msgpublish`、`msgconsume`、`msgsubscribe` 和 `msgunsubscribe` 函数引用的名称。可将预订注册为**永久性**预订或**非永久性**预订，可以指定消息选择器来控制进入的消息，从而确保仅读取感兴趣的消息。

使用 `msgsubscribe` 可告知 JMS 提供程序将消息保存到应用程序逻辑准备好处理它们时。使用 `msgunsubscribe` 可告知 JMS 提供程序，应用程序已不再对此预订中的消息感兴趣。使用 `msgunsubscribe` 可从 JMS 提供程序中删除永久性预订。

所读取的消息中的消息属性可由应用程序逻辑单独处理。

有关 `sp_msgadmin` 及函数的语法、参数和用法信息，请参见第 3 章“SQL 参考”。

使用消息属性

读取某一消息后，Transact-SQL 应用程序逻辑可以使用内置的 SQL 函数来处理该消息的标头和用户属性。这些函数返回以下内容：

- 第 *n* 个属性的名称
- 指定的属性的值
- 指定的属性的类型
- 属性数目

- 属性的列表

以下内置函数允许应用程序逻辑在运行期根据消息属性的值做出处理决定：

- msgproplist
- msgpropname
- msgpropvalue
- msgproptype
- msgpropcount

预览消息传送接口

以下示例简要地预览了 Transact-SQL 消息传送接口。

示例

示例 1 (JMS) 向队列发送消息：

```
select msgsend('hello world',
  ('eas_jms:iiop://my_eas:7222?queue=queue.sample'
  message property 'city=Detroit')
```

示例 2 (JMS) 分别以使用过滤器和不使用过滤器的方式从队列中读取消息：

```
select msgrecv('tibco_jms:tcp://my_jms_host:7222?queue=queue.sample')

select msgrecv
  ('eas_jms:iiop://my_eas:7222?queue=queue.sample'
  message selector 'city='Detroit'')
```

示例 3 (JMS) 将消息发布到主题：

```
sp_msgadmin register, subscription, sub1,
  'eas_jms:iiop://my_eas:7222?topic=topic.sample'
select msgpublish
  ('hello world', 'sub1' message property 'city=Boston')
```

示例 4 (JMS) 使用主题中的消息：

```
select msgconsume('sub1')
```

示例 5 (JMS) 说明如何使用属性:

```
select msgconsume('sub1')
declare @pcount integer
declare @curr integer
declare @pname varchar(100)
select @curr=1
select @pcount = msgpropcount()
while(@curr<=@pcount)
begin
    select @pname=msgpropname(@curr)
    select msgproptype(@pname)
    select msgpropvalue(@pname)
    select @curr=@curr+1
end
```

示例 6 (MQ) 向队列发送消息:

```
select msgsend('hello world',
    'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=DEFAULT.QUEUE'
    message header 'priority=2')
```

示例 7 (MQ) 从队列中读取消息:

```
select msgrecv(
    'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=DEFAULT.QUEUE'
    option 'timeout=30ss')
```

示例 8 (MQ) 注册发布者并发布有关鱼的消息:

```
select msgsend(NULL,
    'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=SYSTEM.BROKER.CONTROL.QUEU
Ei'
    option 'rfhCommand=registerPublisher'
    message header 'topics=fish'
    + ',streamName=ANIMALS.STREAM')
select msgsend('something about a fish',
    'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=ANIMALS.STREAM'
    message header 'topics=fish')
```

示例 9 (MQ) 注册预订者, 读取消息, 并处理消息属性:

```
select msgsend(NULL,
    'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM,queue=SYSTEM.BROKER.CONTROL.QUE
UE'
    option 'rfhCommand=registerSubscriber'
    + ',topics=fish')
```

```
+ ', streamName=ANIMALS.STREAM'  
+ ', queueName=MY_ANIMALS.QUEUE')  
  
select msgrecv(  
    'ibm_mq:channel1/tcp/host1(1234)?qmgr=QM, queue=MY_ANIMALS.QUEUE'  
    option 'timeout=30ss')  
  
select msgpropvalue('MPQScompcode', @@msgproperties)
```

MQ 概述

IBM WebSphere MQ 允许不同的应用程序通过队列在不同的操作系统、不同的处理器以及不同的应用程序系统之间以异步方式进行通信。

WebSphere MQ 包含**消息队列接口 (MQI)**，该接口是一种公共的低级别**应用程序编程接口 (API)**。应用程序使用 MQI 来从队列中读取消息和向队列中写入消息。

队列管理器是一种系统程序，它提供排队服务，且拥有并管理 WebSphere MQ 使用的一组资源。这些资源包括队列、通道、进程定义等。

队列是一种用于存储消息的数据结构。WebSphere MQ 中提供了多种类型的队列对象：

- 本地队列对象 — 标识属于应用程序连接到的队列管理器的本地队列。所有队列都是本地队列，原因是每个队列都属于一个队列管理器；对于该队列管理器而言，该队列为本地队列。
- 远程队列对象 — 标识属于应用程序连接到的队列管理器以外的其它队列管理器的队列。该队列必须定义为远程队列对象所属的队列管理器的本地队列。
- 别名队列对象 — 不是队列，而是指向本地或远程队列的对象指针。
- 模型队列对象 — 定义用作动态队列创建模板的一组队列属性。

所有类型的队列对象都可以在消息中发送，但消息则只能从本地队列对象中读取。

除了 WebSphere MQ 中提供的队列对象类型之外，还有其它一些关于队列的概念：

- 远程队列定义 — 是其它队列管理器所拥有的队列的定义，而非队列本身。
借助远程队列定义，应用程序可以将消息放入远程队列中，而无需指定远程队列或远程队列管理器的名称，也无需指定传输队列的名称。
- 预定义的队列 — 由管理员使用适当的 MQ Series 命令 (MQSC) 或 WebSphere MQ 可编程命令格式 (PCF) 命令创建。预定义的队列是永久队列，独立于使用它们的应用程序而存在，并且在 WebSphere MQ 重新启动后依然存在。
- 动态队列 — 在应用程序发出 MQOPEN 请求（应用程序在此请求中指定了模型队列的名称）时创建。所创建的队列基于模板队列定义（称作模型队列）。动态队列的属性继承自用于创建它们的模型队列。
- 集群队列对象 — 由一个集群队列管理器承载并提供给集群中的其它队列管理器使用。

通道是 WebSphere MQ 客户端与 WebSphere MQ 服务器之间或两个 WebSphere MQ 服务器之间的逻辑通信链路。WebSphere MQ 中有以下两种类别的通道：

- 消息通道 — 是通过消息通道代理连接两个队列管理器的单向链路。
- MQI 通道 — 将 WebSphere MQ 客户端连接到服务器计算机上的队列管理器，在发出 MQCONN 或 MQCONNX 调用时建立。MQ 通道是仅用于传输 MQI 调用和响应的双向链路。

按照 MQI 通道定义，有以下两种通道类型：

- 客户端连接通道 — 连接到 WebSphere MQ 客户端。
- 服务器连接通道 — 连接到运行队列管理器的服务器，该服务器会与在 WebSphere MQ 客户端环境中运行的 WebSphere MQ 应用程序通信。

MQ 通道支持行业标准安全套接字层 (SSL) 协议。有关您所用的 5.3 或 6.0 版 MQ 平台中是否提供 SSL 的信息，请参见 IBM 提供的 WebSphere MQ 文档。

进程定义定义了传入的消息造成触发器事件时执行的进程。

WebSphere MQ 消息由以下两部分组成：

- 消息标头 — 消息控制信息，其中包括一个大小固定的部分和一个大小可变的的部分。
- 消息正文 — 应用程序数据，包含任意类型（text 或 binary）的数据。

使用 `rfhCommand` 发布一项发布时，`msgrecv` 返回的消息有效负载的设置值与其所表示的含义存在如下对应关系：

- MQRHRF — RF 标头包含在消息正文中。
- MQRHRH — 不包含 RF 标头。

可以通过查询 `@@msgproperties` 来获得 RF 标头中的名称 — 值对。

如果消息正文包含字符，则代码集约定通过 MQ 本机服务提供或通过用户出口处理程序提供。消息正文的格式由消息标头中的一个字段定义。尽管在示例中提供了一些格式，但 MQ 并不会枚举所有可能的消息正文格式。应用程序可以输入格式的任意名称。例如，“MQSTR”包含字符串数据，“MQRHRF”则包含 MQ 发布 / 预订的主题。

WebSphere MQ 消息类型包括：

- 数据报 — 不应有答复。
- 请求 — 应有答复。
- 答复 — 对请求消息的答复。
- 报告 — 包含来自队列管理器或其它应用程序的状态信息。

发送消息时，可以设置各种消息标头属性，例如有效期、持久性、优先级、相关 ID 和答复队列。

通过消息分组，可以将一组消息组织成一个逻辑命名组。组中的每个逻辑消息可以进一步划分成若干段。组用名称加以标识，组中的每个逻辑消息用序列号加以标识（起始序列号为 1），逻辑消息的每个段用消息数据相对于逻辑消息的偏移量加以标识。MQ 发布 / 预订不支持经过分段的消息，尝试发送经过分段的消息时会导致出错。

在队列中，消息以它们发送到队列的物理顺序显示。这意味着不同组的消息可能是分散的，并且在组内，消息的序列号可能是无序的（如果两个应用程序发送的消息具有相同的组 ID 和划分的序列号，则可能会出现后一种情况）。

收到消息后，读取模式可以是：

- 破坏性的 — 删除相应消息；或
- 非破坏性的 — 保留相应消息。这种读取称作“浏览”，允许应用程序先细读一条或多条消息，再决定从队列中删除特定消息。

接收方可以通过指定消息标头属性（如相关 ID 或消息 ID）来选择特定消息。

以破坏性模式或非破坏性模式读取消息时，消息可以按物理顺序或逻辑顺序返回。具体的顺序由队列的定义加以定义。队列可以定义为采用优先级顺序或先进先出顺序。

使用 SSL 保护通道安全

通过 SSL 发送和接收消息的步骤如下：

- 1 为所连接的队列管理器创建一个密钥存储库，在其中包含该队列管理器的私有密钥和 Adaptive Server 的数字证书。
- 2 为 Adaptive Server 创建一个密钥存储库，在其中包含该 Adaptive Server 及所连接的队列管理器的数字证书。
- 3 在所连接的队列管理器中创建一个启用 SSL 的服务器连接通道。
- 4 使用 `sp_msgadmin 'config', 'ibmmq_keystore'` 存储过程对 Adaptive Server 的密钥存储库进行配置；有关该存储过程的说明，请参见第 3 章“SQL 参考”中的第 57 页的 `sp_msgadmin`。

示例

本应用场景说明了在 Active Messaging 中 WebSphere MQ 在使用和不使用 SSL 的情况下分别如何通信。

在队列管理器“BACH”中有两个服务器连接通道；第一个通道“CH1”为常规连接，而“CH2”则配置为需要使用 SSL。第二个通道的 SSL 密码规范为 NULL_MD5。

- 1 在不启用 SSL 的情况下向队列管理器发送消息：

```
select msgsend('a', "ibm_mq:CH1/tcp/host1(7654)?qmgr=BACH,queue=Q1')
```

- 2 接下来，使用 SSL 协议向队列管理器发送消息：

- a 分开设置队列管理器和 Adaptive Server 的密钥存储库。Adaptive Server 的密钥数据库文件为 `/var/mqm/clients/ssl/ASE.kdb`。有关如何设置密钥存储库的操作说明，请参见 IBM 提供的 WebSphere MQ 文档。

- b 使用下面的语句对 Adaptive Server 的密钥存储库进行配置：

```
sp_msgadmin 'config', 'ibmmq_keystore', '/var/mqm/clients/ssl/ASE'
```

- c 通过 SSL 发送消息：

```
select msgsend('e', 'ibm_mq:CH2(ssl:sslciph=NULL_MD5)
/tcp/host1(7654)?qmgr=BACH,queue=Q1')
```


MQ 发布 / 预订

在利用代理进程执行预订解析的 MQ 队列中，使用 WebSphere MQ 发布 / 预订。在其最简单的形式中：

- 发布者是发送消息的应用程序。
- 预订者是接收消息的应用程序。
- 涉及以下队列：
 - 控制队列 — 在此队列中，发布者和预订者向发布 / 预订代理发送指令，如预订者注册和取消。
 - 流队列 — 在此队列中，发布者直接发送其消息。发布 / 预订代理从流队列中读取消息，并将它们分发到相应预订者的队列。
 - 预订者队列 — 在此队列中，预订者直接读取其消息。

注释 可能会涉及更多的队列，具体取决于发布类型。

- 发布 / 预订代理会对发送到控制队列的 **MQRFH** 消息进行响应。这些命令消息控制发布 / 预订代理如何处理到达流队列的消息。例如，预订者可以通过注册来表示其对特定主题感兴趣。
- 发布者直接向流队列发送消息。
- 发布 / 预订代理从流队列中读取消息，并决定要将所读取的消息复制到哪个预订者队列。这取决于预订者已通过注册表示出兴趣的主题。
- 预订者直接从预订者队列中读取消息。

预订者会注册“预订”，这意味着它对一个或多个“主题”感兴趣。

示例

本示例说明了 MQ 发布 / 预订过程，示例中用到以下变量：

```

declare @BROKER      varchar(100)
declare @STREAM      varchar(100)
declare @SUBQ        varchar(100)
declare @QM          varchar(100)
select @QM           = 'ibm_mq:channel1/tcp/host1(9876)?qmgr=QM'
select @BROKER       = 'SYSTEM.BROKER.CONTROL.QUEUE'
select @STREAM        = 'ANIMALS'
select @SUBQ          = 'MY_ANIMALS'

```

- 1 发布者进行注册，表示其要将主题为鱼的发布发送到 ANIMALS:

```
select msgsend(NULL,
               @QM + ',queue=' + @BROKER
               option 'rfhCommand=registerPublisher'
               message header 'topics=fish,streamName=' + @STREAM)
```

- 2 预订者进行注册，表示其要接收发布到 ANIMALS 且主题为鱼的发布。预订者在 MY_ANIMALS 中接收这些发布:

```
select msgsend(NULL,
               @QM + ',queue=' + @BROKER
               option 'rfhCommand=registerSubscriber'
               message header 'topics=fish'
                           + ',streamName=' + @STREAM
                           + ',queueName=' + @SUBQ1)
```

- 3 发布者向 ANIMALS 发布关于鱼的发布。MQ 发布 / 预订代理自动将此发布转发到 MY_ANIMALS:

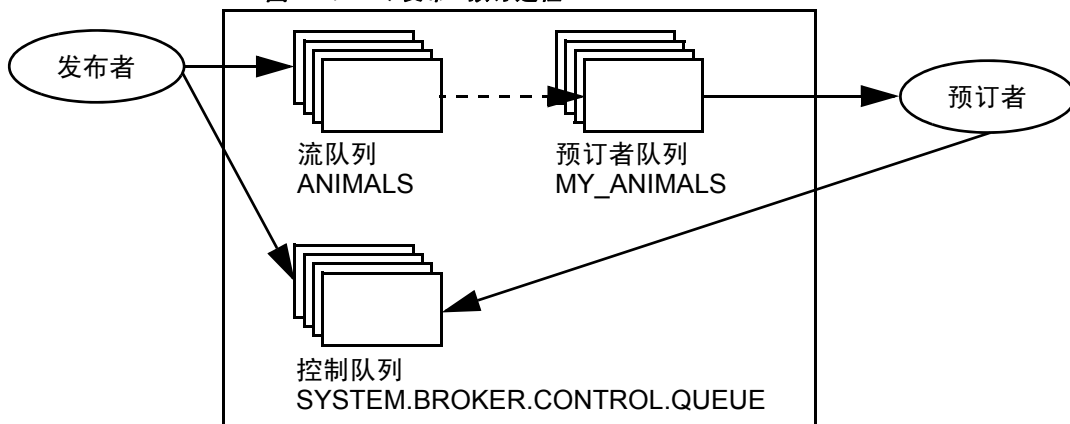
```
select msgsend('something about fish',
               @QM + ',queue=' + @STREAM
               option 'rfhCommand=publish'
               message header 'topics=fish')
```

- 4 预订者从 MY_ANIMALS 中读取转发的消息:

```
select msgrcv(@QM + ',queue=' + @SUBQ option 'timeout=30ss')
```

图 2-1 显示了此 MQ 发布 / 预订过程示例的流程。

图 2-1: MQ 发布 / 预订过程



一条消息可以具有一个或多个主题。WebSphere MQ 发布 / 预订模型建议对主题采用分层命名约定，如下面的示例中所示。预订者在指定感兴趣的主题时可以指定通配符（如 * 和 ?）。

下面是一些主题示例：

```
Sport
Sport/Soccer
Sport/Tennis
```

下面这些示例说明了预订者可以如何指定感兴趣的主题：

```
Sport/*           - Any topic about sports.
*/Soccer         - Any topics about soccer.
*/Soccer/Trades - Any topics about soccer where a 'trade' is involved.
```

保留发布是这样的一种发布：MQ 发布 / 预订代理在将消息传递到所有预订者后依然保留消息的副本。通常，将副本传递给所有预订者后，会删除相应发布。保留发布允许预订者以异步方式请求保留发布，而不是依靠 MQ 发布 / 预订代理来传递它。这些类型的消息通常包含状态信息，因而也称作状态发布。

主题的语法

- 主题通常采用“主题 / 子主题”形式，例如“sport/baseball”。
- 可以在主题中指定通配符，如“*”或“?”。
- 指定多个主题时，请用冒号将各个主题隔开。例如，“topic1:topic2:topic3:”，依此类推。
- 如果主题包含空格或逗号，请将整个主题列表用引号引起来。由于主题可以作为字符串显示在 `message header` 或 `message property` 子句中，因此，如果所传递的选项字符串是用引号引起来的标量值，则必须通过将两侧的引号输入两次来对它们进行转义。此外，如果主题内还嵌入了双引号，则必须通过将所嵌入的双引号输入四次来对它们进行转义。例如：

```
-- Topic has embedded spaces, we need to quote with escaped quotes
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics='''Sport/Football/Hometown Bulldogs''')
```

```
-- Topic has embedded spaces, we can quote with double quotes
select msgsend(NULL,
```

```

'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
option 'msgType=datagram,rfhCommand=publish'
message property 'topics="Sport/Football/Hometown Bulldogs"')

-- Topic has embedded spaces and embedded double quotes, the inner
-- double quotes need to be escaped.
set quoted_identifier off
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics="quoted ""topic"" here"')

-- Topic has embedded spaces and embedded double quotes, double the
-- quotes around the topic, and quadruple the embedded quotes.
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property "topics=""quoted """"topic"""" here""")

```

- 主题中嵌入了空格或引号时，在 MQRF 标头中会将该主题引起来。如果主题中嵌入了引号，则会先对这些引号进行转义，再将它们放入 MQRF 标头中。

在下面的示例中，MQRF 标头中放入了一个
“Sport/Football/Hometown Bulldogs” 形式的主题：

```

select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics='Sport/Football/Hometown Bulldogs''')

```

在下面的示例中，MQRF 主题中放入了一个 **“Books/” Recipes Of Spain”””** 形式的主题。

```

select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics=''Books/'Recipes Of Spain''')

```

- 可以使用 “::” 来对主题名称进行转义；任何单个的非转义尾随 “:” 都将被忽略。

在下面的示例中，有三个主题，即 **“baseball”**、**“baseball/anytown”** 和 **“baseball/scores”**。

```
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics=baseball:baseball/anytown:baseball/scores')
```

在下面的示例中，有三个主题，即 **“subject1”**、**“subject:2”** 和 **“subject3”**。使用双冒号（“::”）对嵌入的“:”进行转义。

```
select msgsend(NULL,
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=SAMPLE.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics=subject1:subject::2:subject3')
```

发布者和预订者的标识

缺省情况下，发布者或预订者的标识由以下内容组成：

- 一个队列名称。
- 一个队列管理器名称。
- 一个相关标识符（可选）。可以使用相关标识符来区分使用相同队列的不同发布者或预订者。可以为每个发布者和预订者分配一个不同的相关标识符，从而使多个应用程序能够共享一个队列或者使单个应用程序能够区分来自不同预订的发布。

MQ 发布 / 预订示例

发布者示例

Adaptive Server 会话是一个发布者。它发布关于 **“topicA”** 和 **“topicB”** 的内容；关于 **“topicB”** 的发布作为保留发布进行发布。最后删除此保留发布。

```
-- @QM has the queue manager endpoint
declare @QM          varchar(100)
-- @BROKER has the broker queue name
declare @BROKER     varchar(100)
-- @STREAM has the stream queue name
declare @STREAM     varchar(100)
-- @CORRELID has the generated correlation id
declare @CORRELID   varchar(100)

-- Put Queue manager name, broker and stream queue names into variables
select @QM          = 'ibm_mq:chan1/tcp/localhost(5678)?qmgr=QM1'
select @BROKER     = 'SYSTEM.BROKER.CONTROL.QUEUE'
select @STREAM     = 'Q1.STREAM'
```

```
-- Register the publisher, only for topicA
select msgsend(NULL, @QM + ',queue=' + @BROKER
              option 'rfhCommand=registerPublisher'
              message header 'correlationAsId=generate'
                             + ',topics=topicA'
                             + ',streamName=' + @STREAM)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014801

-- Save the generated correlation id
select @CORRELID = @@msgcorrelation

-- Send two publications on topicA
select msgsend('topicA, publication 1', @QM + ',queue=' + @STREAM
              option 'rfhCommand=publish'
              message header 'correlationAsId=yes'
                             + ',correlationId=' + @CORRELID
                             + ',topics=topicA')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014803

select msgsend('topicA, publication 2', @QM + ',queue=' + @STREAM
              option 'rfhCommand=publish'
              message header 'correlationAsId=yes'
                             + ',correlationId=' + @CORRELID
                             + ',topics=topicA')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014805

-- Add another topic for this publisher
select msgsend(NULL, @QM + ',queue=' + @BROKER
              option 'rfhCommand=registerPublisher'
              message header 'correlationAsId=yes'
                             + ',correlationId=' + @CORRELID
                             + ',topics=topicB'
                             + ',streamName=' + @STREAM)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014807

-- Publish a retained message on topicB
select msgsend('topicB, retained publication 1', @QM + ',queue=' + @STREAM
              option 'rfhCommand=publish'
              message header 'correlationAsId=yes'
                             + ',correlationId=' + @CORRELID
                             + ',topics=topicB')
```

```

        + ',retainPub=yes')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014809

-- Publish a second retained publication on topicB
-- This one will replace the current retained publication on topicB.
select msgsend('topicB, retained publication 2', @QM + ',queue=' + @STREAM
    option 'rfhCommand=publish'
    message header ',correlationAsId=Yes'
        + ',correlationId' + @CORRELID
        + ',topics=topicB'
        + ',retainPub=yes')
-----
0x414d51204652414e4349532e514d202041a3ebfb2001480b

-- Delete the retained publication on topicB
select msgsend(NULL, @QM + ',queue=' + @STREAM
    option 'rfhCommand=deletePublication'

    message header 'topics=topicB'
        + ',streamName=' + @STREAM)
-----
0x414d51204652414e4349532e514d202041a3ebfb2001480d

-- Deregister the publisher, for all topics.
select msgsend(NULL, @QM + ',queue=' + @BROKER
    option 'rfhCommand=deregisterPublisher'
    message header 'correlationAsId=yes'
        + ',correlationId=' + @CORRELID
        + ',deregAll=yes'
        + ',streamName=' + @STREAM)
-----
0x414d51204652414e4349532e514d202041a3ebfb2001480f

```

预订者示例

在本示例中，Adaptive Server 会话预订 “topicA” 和 “topicB”；关于 “topicB” 的发布作为保留发布进行发布。此预订者通过从发布 / 预订代理请求获得更新来处理保留发布。

```

-- @QM has the queue manager endpoint
declare @QM          varchar(100)
-- @BROKER has the broker queue name
declare @BROKER     varchar(100)
-- @SUBQUEUE has the subscriber queue name
declare @SUBQUEUE   varchar(100)
-- @STREAM has the stream queue name

```

```

declare @STREAM          varchar(100)
-- @CORRELID has the generated correlation id
declare @CORRELID        varchar(100)

-- Put broker and subscriber queue names into variables
select @QM                = 'ibm_mq:chan1/tcp/localhost(5678)?qmgr=QM1'
select @BROKER            = 'SYSTEM.BROKER.CONTROL.QUEUE'
select @SUBQUEUE         = 'Q1.SUBSCRIBER'
select @STREAM            = 'Q1.STREAM'

-- Register the subscriber, only for topicA
select msgsend(NULL, @QM + ',queue=' + @BROKER
                option 'rfhCommand=registerSubscriber'
                message header 'correlationAsId=generate'
                                + ',topics=topicA'
                                + ',streamName=' + @STREAM
                                + ',queueName=' + @SUBQUEUE)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014801

-- Save the generated correlation id
select @CORRELID = @@msgcorrelation

-- Add another topic for this subscriber
-- we will explicitly request update for publications on this topic.
select msgsend(NULL, @QM + ',queue=' + @BROKER
                option 'rfhCommand=registerSubscriber'
                message header 'CorrelationAsId=yes'
                                + ',correlationId=' + @CORRELID
                                + ',topics=topicB'
                                + ',streamName=' + @STREAM
                                + ',queueName=' + @SUBQUEUE
                                + ',pubOnReqOnly=yes')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014803

-- The publisher now publishes messages in the following order:
-- topicA, topicB (*), topicA, topicB (*)
-- ( '*' denotes a retained publication )

-- Get the first message on the subscriber queue, it will be on topicA.
select msgrecv(@QM + ',queue=' + @SUBQUEUE option 'timeout=30ss')
-----
publication on topicA

-- Get the second message on the subscriber queue, it will be on topicA.

```



```

select msgrecv(@QM + ',queue=' + @SUBQUEUE option 'timeout=30ss')
-----
publication on topicA

-- Request the broker to now send retained publications on topicB
select msgsend(NULL, @QM + ',queue=' + @BROKER
    option 'rfhCommand=requestUpdate'
    message header 'CorrelationAsId=yes'
                  + ',correlationId=' + @CORRELID
                  + ',topics=topicB'
                  + ',streamName=' + @STREAM
                  + ',queueName=' + @SUBQUEUE)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014805

-- Get the next message on the subscriber queue, it will be on topicB.
select msgrecv(@QM + ',queue=' + @SUBQUEUE option 'timeout=30ss')
-----
publication on topicB

-- Get the next message on the subscriber queue, it will be on topicB.
select msgrecv(@QM + ',queue=' + @SUBQUEUE option 'timeout=30ss')
-----
publication on topicB

-- Deregister the subscriber, for all topics.
select msgsend(NULL, @QM + ',queue=' + @BROKER
    option 'rfhCommand=deregisterSubscriber'
    message header 'CorrelationAsId=yes'
                  + ',correlationId=' + @CORRELID
                  + ',deregAll=yes'
                  + ',streamName=' + @STREAM
                  + ',queueName=' + @SUBQUEUE)
-----
0x414d51204652414e4349532e514d202041a3ebfb20014807

```

代理响应示例

本示例说明如何使用请求 / 响应消息传送来检查来自发布 / 预订代理的响应。user1 会注册一个预订，然后将检查发布 / 预订代理的响应。接着，user2 会以另一个预订名称再次注册此预订，从而导致发布 / 预订代理做出错误响应。

user1 执行的查询:

```
-- @QM has the queue manager endpoint
declare @QM          varchar(100)
-- @BROKER has the broker queue name
declare @BROKER      varchar(100)
-- @SUBQUEUE has the subscriber queue name
declare @SUBQUEUE    varchar(100)
-- @REPLY has the reply queue name
declare @REPLY       varchar(100)

-- Put broker, subscriber and reply queue names into variables
select @QM           = 'ibm_mq:chan1/tcp/localhost(5678)?qmgr=QM1'
select @BROKER       = 'SYSTEM.BROKER.CONTROL.QUEUE'
select @SUBQUEUE     = 'Q1.SUBSCRIBER'
select @REPLY        = 'Q1.REPLY'

-- Register the subscriber.
select msgsend(NULL, @QM + ',queue=' + @BROKER
               option 'rfhCommand=registerSubscriber, msgType=request'
               message header 'correlationAsId=generate'
                           + ',topics=topicA'
                           + ',streamName=Q1.STREAM'
                           + ',queueName=Q1.SUBSCRIBER'
                           + ',replyToQueue=Q1.REPLY')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014801

-- Read the response
select msgrecv(@QM + ',queue=' + @REPLY option 'timeout=30ss')
-----
NULL

-- Check @@msgproperties
select @@msgproperties
-----
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<msgproperties
      MQPSReasonText="&apos;MQRC_NONE&apos;"
      MQPSReason="0"
      MQPSCompCode="0">
</msgproperties>

-- Check MQPSCompCode
if (msgpropvalue('MQPSCompCode', @@msgproperties) != "0")
begin
    print "registerSubscriber failed"
end
```

user2 执行的查询:

```
-- @QM has the queue manager endpoint
declare @QM          varchar(100)
-- @BROKER has the broker queue name
declare @BROKER      varchar(100)
-- @SUBQUEUE has the subscriber queue name
declare @SUBQUEUE    varchar(100)
-- @REPLY has the reply queue name
declare @REPLY       varchar(100)

-- Put broker, subscriber and reply queue names into variables
select @QM=          'ibm_mq:chan1/tcp/localhost(5678)?qmgr=QM1'
select @BROKER=      'SYSTEM.BROKER.CONTROL.QUEUE'
select @SUBQUEUE=    'Q1.SUBSCRIBER'
select @REPLY=       'Q1.REPLY'

-- Register the subscriber
select msgsend(NULL, @QM + ',queue=' + @BROKER
    option 'rfhCommand=registerSubscriber, msgType=request'
    message header 'correlationAsId=generate'
                + ',topics=topicA'
                + ',streamName=Q1.STREAM'
                + ',queueName=Q1.SUBSCRIBER'
                + ',replyToQueue=Q1.REPLY')
-----
0x414d51204652414e4349532e514d202041a3ebfb20014801

-- Read the response
select msgrecv(@QM + ',queue=' + @REPLY option 'timeout=30ss')
-----
NULL

-- Check @@msgproperties
select @@msgproperties
-----
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<msgproperties
    MQPSUserId="&apos;user2 &apos;"
    MQPSReasonText="&apos;MQRCCF_DUPLICATE_IDENTITY&apos;"
    MQPSReason="3078"
    MQPSCompCode="2"
</msgproperties>

-- Check MQPSCompCode
if (msgpropvalue('MQPSCompCode', @@msgproperties) != "0")
begin
print "registerSubscriber failed"
end
```

使用 MQ 集群队列对象

Active Messaging 允许使用 Adaptive Server 作为客户端来与 WebSphere MQ 集群功能通信。可以使用 `msgsend` 向连接到队列管理器的任何集群中的所有集群队列发送消息。

注释 `msgrecv` 函数不支持远程队列连接。

集群中可以有多多个队列管理器承载同一队列的实例。例如，两个分别名为 `MASTER_MQ1` 和 `SLAVE_MQ1` 的队列管理器都承载集群队列 `CQ1`。之后这两个队列管理器都加入集群 `INV_CQ1`，从而造成集群 `INV_CQ1` 中存在 `CQ1` 集群队列的两个实例。

要指定远程队列管理器，请在 `endpoint` 语法段中使用 `remote_qmgr`。如果要将消息发送到承载多个实例的集群队列，并且不关心目标是哪个实例或者不需要在各集群队列实例之间平衡工作负荷，请忽略此 `remote_qmgr` 选项。在此类情况下，WebSphere MQ 独自平衡工作负荷：

- 如果所连接的队列管理器中有实例，则 WebSphere MQ 会自动选择该实例。
- 如果所连接的队列管理器中无实例，则 WebSphere MQ 会决定选择哪个实例合适。

如果不想使用缺省算法，请定义一个集群工作负荷出口。出口是 WebSphere MQ 的一项功能，与数据库中的触发器类似。有关出口及其定义方式的详细信息，请参见 IBM WebSphere MQ 文档。

通过将集群与同一队列的多个实例结合使用，可以将消息传送到承载恰当队列的副本的任何队列管理器。不过，这对需要保持多条消息的顺序完整性的用户有不利影响。例如，某一客户将以下消息发送给供应商：

- 1 “送 100 件小器具”，发送于上午 9:00
- 2 “送 50 件小器具”，发送于上午 9:30
- 3 “取消第一个请求”，发送于上午 10:00

在本示例中，这些消息必须保持正确的顺序，才能使供应商知道客户最终要购买的小器具数量为 50 件（即 $100 + 50 - 100 = 50$ ）。如果第 2 条消息先于第 1 条消息到达，则供应商会误认为客户要购买 100 件小器具。

用户可以通过将这些消息放入同一实例中来解决此问题，具体方法是：在 `msgsend` 函数中指定 `clustqBinding`（一种 `option_string` 类型）。`clustqBinding` 的选项为 `bind`、`nobind` 和 `default`。有关这些选项及相应示例的完整说明，请参见第 93 页的 `msgsend` 的参考页。

使用远程队列对象

在以下情况中，可以通过使用 `msgsend remote_qmgr` 选项指定远程队列管理器的名称向远程队列对象发送消息：

- 本地队列管理器和远程队列管理器位于单个集群中，并且本地队列管理器存储着远程队列管理器的集群队列管理器定义。
- 本地队列管理器中有一个传输队列，且此传输队列与远程队列管理器中的传输队列同名。
- 本地队列管理器中有一个队列管理器别名，且此队列管理器别名与远程队列管理器中的队列管理器别名同名。

注释 Adaptive Server 将远程队列管理器设置为目标队列管理器，并将相应的队列设置为目标队列。Adaptive Server 向相关的传输队列发送一条消息后，即使它尚未向目标队列发送消息，Adaptive Server 也会立即以成功状态返回。

有关 WebSphere MQ 如何在队列管理器之间传输消息的详细信息，请参见 IBM 文档。

消息放入传输队列中后，本地队列管理器会在自己的子网中查找远程队列管理器定义。如果本地队列管理器是：

- 集群的完整存储库 — 则本地队列管理器应包含远程集群队列管理器的定义。
- 部分存储库 — 则本地队列管理器可能不知道远程定义的位置，在这种情况下 WebSphere MQ 会返回错误。不过，发生此情况时，本地队列管理器随后不会寻找远程队列管理器的位置。

如果本地队列管理器找到远程队列管理器定义，则本地队列管理器会通过集群传输队列向远程队列管理器发送一条消息，然后远程队列管理器会向目标队列发送一条消息。这样，即使目标队列不是集群队列，操作也会成功完成。

对于其它情况，相关传输队列的通道会接收消息并将它们发送到该通道连接到的队列管理器。如果不存在此类通道或该通道尚未启动，则传输队列会将消息存储下来，直到该通道启动为止。

使用文本消息传送

JMS 和 WebSphere MQ 都可以处理字节消息和文本消息。

文本消息和 JMS

在 JMS 中发送或接收消息时，Active Messaging 会自动检测消息有效负载的数据类型，并相应地将其作为字节消息或文本消息进行处理。发送消息时，JMS 将 `char`、`varchar`、`unichar`、`univarchar`、`text` 和 `unitext` 识别为有效的文本消息类型。

示例

示例 1 将文本消息发送至 JMS 消息传送总线：

```
declare @msg varchar(1024)
select @msg = 'abcd'
select msgsend(@msg,
  'tibco_jms:tcp://my_jms:7222?queue=sample,user=admin')
```

示例 2 从 JMS 消息传送总线接收文本消息：

```
select msgrecv('tibco_jms:tcp://my_jms:7222?
  queue=sample,user=admin', returns varchar(1024))
```

示例 3 将字节消息发送至 JMS 消息传送总线：

```
declare @msg varbinary(1024)
select @msg = 'abcd'
select msgsend(@msg,
  'tibco_jms:tcp://my_jms:7222?queue=sample,user=admin')
```

示例 4 从 JMS 消息传送总线接收字节消息：

```
select msgrecv('tibco_jms:tcp://my_jms:7222?
  queue=sample,user=admin', returns varbinary(1024))
```

文本消息和 MQ

在 WebSphere MQ 中接收消息时，MQ 仅在“formatName”消息属性设置为“MQSTR”时才将消息视作文本消息。否则，MQ 会将消息作为字节消息进行处理。

示例

示例 1 将文本消息发送到 WebSphere MQ。

```
declare @msg varchar(1024)
select @msg = 'abc'
select msgsend(@msg, 'ibm_mq:channel1/TCP/host1(7654)?
  qmgr=QM,queue=Q1,alter_user=yes',message property "formatName=MQSTR")
```

示例 2 从 WebSphere MQ 接收文本消息:

```
select msgrecv('ibm_mq:channel1/TCP/host1(7654)?
  qmgr=QM,queue=Q1,alter_user=yes',
  option 'bufferLength=20000k,timeout=60000',
  returns varchar(1024))
```

示例 3 将字节消息发送到 WebSphere MQ:

```
declare @msg varbinary(1024)
select @msg = 'abc'
select msgsend(@msg,'ibm_mq:channel1/TCP/host1(7654)?
  qmgr=QM,queue=Q1,alter_user=yes')
```

示例 4 从 WebSphere MQ 接收字节消息:

```
select msgrecv('ibm_mq:channel1/TCP/host1(7654)?
  qmgr=QM,queue=Q1,alter_user=yes',
  option 'bufferLength=20000k,timeout=60000',
  returns varbinary(1024))
```

示例 5 只要字节有效负载是以 UTF8 格式编码的,就可以在 WebSphere MQ 中将其作为文本消息进行发送。在本示例中,文本消息“abc”基于字节有效负载 0x616263,这是因为文本“abc”的 UTF8 编码是 0x616263:

```
declare @msg varbinary(1024)
select @msg = 0x616263
select msgsend(@msg,'ibm_mq:channel1/TCP/host1(7654)?
  qmgr=QM,queue=Q1,alter_user=yes',
  message property "formatName=MQSTR")
```

Adaptive Server Cluster Edition 支持

Active Messaging 支持 Adaptive Server Cluster Edition 的以下客户端技术:

- [第 28 页](#)的“登录重定向” — 实例在确认登录前将传入的客户端连接重定向到其它实例的功能。登录重定向在登录序列期间进行。客户端应用程序不会收到关于已将其重定向的通知。
- [第 29 页](#)的“扩展的高可用性” — 在扩展的故障切换配置中, Adaptive Server 会在“支持高可用性的”客户端连接时为它们提供一个故障切换地址列表。这样,如果支持高可用性的客户端或应用程序连接到的实例出现故障,则这些客户端或应用程序可以进行多次故障切换。

这些客户端的 `interfaces` 文件或目录服务中不需要有 `HAFAILOVER` 条目。不过，如果它们的 `interfaces` 文件或目录服务中确实有 `HAFAILOVER` 条目，则这些客户端会继续使用此条目，直到 Adaptive Server 向它们发送要连接到的故障切换地址或服务器的列表。这些客户端始终使用 Adaptive Server 提供的最新列表。

注释 Active Messaging 功能不支持连接迁移，当现有客户端从集群的一个实例转移到另一个实例时，会发生连接迁移。

登录重定向

登录重定向由 Adaptive Server 的工作负荷管理器用来根据逻辑集群配置和集群的当前工作负荷将传入的连接发送到特定实例。

在登录时，如果一个实例由于负载的原因通知客户端登录到另一个实例，就会发生登录重定向。

不需要为客户端重定向执行任何其它配置；它会自动进行。

下面的示例包括节点 “blade1” 和 “blade2” 上的实例 “ase1” 和 “ase2”，这两个节点在集群 “mycluster” 中运行。

```
ase1
  query tcp ether blade1 19786
ase2
  query tcp ether blade2 19786

mycluster
  query tcp ether blade1 19786
  query tcp ether blade2 19786
```

例如，如果在 “ase1” 和 “ase2” 上启用了 Active Messaging，并且应用程序服务器尝试连接到 “ase1”，但 “ase1” 不可用，则此登录会重定向到 “ase2” 实例以执行消息传送操作。

请参见《集群用户指南》。

扩展的高可用性

在“支持 HA 的”客户端连接时，Adaptive Server 会为其提供一个故障切换地址列表。这样，当支持高可用性的客户端或应用程序连接到的实例变得不可用时，这些客户端或应用程序可以进行多次故障切换。如果该实例尚未向客户端发送故障切换列表，则客户端会使用 `interfaces` 文件中的 `HAFILOVER` 条目信息。

在下面的示例中，如果在登录期间出现网络故障且实例尚未发送扩展的高可用性列表，则允许支持 HA 的客户端进行故障切换。

```
ase1
  query tcp ether blade1 19786

ase2
  query tcp ether blade2 19786

mycluster
  query tcp ether blade1 19786
  query tcp ether blade2 19786
  hafailover mycluster
```

`HAFILOVER` 条目应使用集群别名作为服务器名称，原因是客户端应用程序会尝试每个 `query` 行，直到它与集群中的实例建立连接为止。有关如何在集群环境中启用扩展的高可用性的信息，请参见《集群用户指南》。

扩展的故障切换需要 Open Client 15.0 ESD #3 或更高版本。Cluster Edition 中的客户端库包含 ESD #8。

Open Client 使用 `CS_PROP_EXTENDEDFAILOVER` 属性进行扩展的故障切换。请参见《Client-Library/C 参考手册》。

对线程化内核的 Active Messaging 支持

如果要将 Active Messaging 与线程化内核一起使用，则不要配置以下参数：

- `max online Q engines`
- `number of Q engines at startup`
- `max online engines`
- `number of engines at startup`

必须将 `syb_blocking_pool` 配置为具有至少 1 个线程，而且，出于性能原因，应该至少和运行 Active Messaging 的 Adaptive Server 会话一样大。例如，如果在配置 Active Messaging 之前 `syb_blocking_pool` 的线程计数为 2，则在配置 Active Messaging 之后，有 3 个附加 Adaptive Server 会话使用 Active Messaging。Sybase® 建议您将 `syb_blocking_pool` 的线程计数配置为 5，因为 $2+3=5$ 。

注释 此配置与低于 15.7 版的 Adaptive Server 兼容。

国际化支持

Adaptive Server 15.0.2 ESD #1 及更高版本为 Adaptive Server 与消息传送总线之间的消息发送和接收提供国际化支持，例如：

- 当发送方的服务器字符集配置为采用 GB18030（简体中文）时 — 发送方可以向消息传送总线发送中文消息。
- 当接收方的服务器字符集配置为采用 Big5（繁体中文）时 — 接收方可以从消息传送总线接收中文消息。

示例

示例 1 设置当前字符集，然后向一个采用 GB18030 字符集的 Adaptive Server 中的消息传送总线发送一个中文词语：

```

1> sp_configure "default character set id"
2> go

Parameter Name          Default Memory Used Config Value Run Value Unit  Type
-----
default character set id 1              0          173      173  id  static

(1 row affected)
(return status = 0)

1> declare @msg varchar(1024)
2> select @msg = 0xd6d0cec4
3> select msgsend(@msg, 'ibm_mq:channel1/TCP/host1(7654)?
      qmgr=QM,queue=Q1,alter_user=yes',message property "formatName=MQSTR")
    
```

示例 2 从运行 Big5 字符集的另一个 Adaptive Server 中的消息传送总线接收中文消息：

```
1> sp_configure "default character set id"
2> go

Parameter Name          Default  Memory Used  Config Value  Run Value  Unit  Type
-----
default character set id  1          0           161          161 id static
(return status = 0)

1> declare @msg varchar(1024)
2> select @msg = msgrecv('ibm_mq:channel1/TCP/host1(7654)?
   qmgr=QM,queue=Q1,alter_user=yes',
3> option 'bufferLength=100k,timeout=60000',
4> returns varchar(16384))
5> select convert(varbinary(1024), @msg)
6> go

-----
0xa4a4a4e5
```

输出“0xa4a4a4e5”是采用 Big5 字符集的汉语中“中文”一词的二进制表示形式。

事务型消息行为

缺省情况下，如果数据库事务回退，则所有消息传送操作（`msgsend`、`msgrecv`、`msgpublish`、`msgconsume`、`msgsubscribe` 和 `msgunsubscribe`）都会回退。不过，如果使用 `msgsend` 或 `msgrecv` 执行的消息传送操作失败，则不会影响父级数据库事务。

- 如果某一事务中包含的进程执行 `msgsend` 或 `msgpublish`，则在该进程提交该事务前，所产生的消息在消息总线中一直是不可见的。这与执行 SQL `update` 或 `insert` 不同 — 在事务中执行 SQL `update` 和 `insert` 命令的进程在事务提交前便会立即看到这些命令产生的效果。
- 通过在事务中执行 `msgsend` 或 `msgpublish` 来发送消息的进程在提交该事务前，无法使用 `msgrecv` 或 `msgconsume` 读取该消息。

事务型消息传送的 set 选项

事务型行为由 `set transactional messaging` 命令控制，该命令提供三种操作模式，从而使您在事务中使用消息传送函数时可以选择首选的行为：

```
set transactional messaging [ none | simple | full]
```

- **none** — 使消息传送操作和数据库操作不会相互影响。在下面的示例中，不论 `insert` 成功与否，都会执行 `msgsend` 并将消息发送到消息总线：

```
begin tran
    msgsend (...)
    insert (...)
rollback
```

- **simple**（缺省设置）— 使得数据库操作影响消息传送操作，但消息传送操作不会影响数据库事务。在下面的示例中，如果 `msgsend` 失败，则 `insert` 不会中止：

```
begin tran
    insert (...)
    msgsend (...)
commit
```

在下面的示例中，会回退 `msgsend`：

```
begin tran
    insert (...)
    msgsend (...)
rollback
```

- **full** — 提供完全的事务型行为。在此模式下，消息传送操作和数据库操作会相互影响。如果消息传送操作失败，相应的事务将回退。如果数据库事务失败，则消息传送操作将回退。

```
begin tran
    select @message=msgrecv(Q1,...)
    insert t2 values (@message,...)
    select msgsend ( t2.status,...)
commit tran
```

- 当 `transactional messaging` 设置为 **full** 或 **simple** 时，无法在同一事务内读取用来发送或发布消息的未提交事务。

Transact-SQL 应用程序可以指定一种首选模式，具体取决于其应用程序需求。

注释 不能在事务内使用 `set transactional messaging`。

MQ 安全性

本节讨论安全性与 MQ。

连接到 MQ 队列管理器

使用 MQ 结束点时，不能像使用 JMS 那样指定用户名和口令。到 MQ 队列管理器的所有连接都以用来运行 Adaptive Server 进程的用户身份建立。与 MQ 队列管理器建立连接后，Adaptive Server 随后会尝试以当前执行操作的 Adaptive Server 登录名身份打开队列。因此，用户必须：

- 在运行着 MQ 队列管理器的计算机上拥有一个用户帐号。如果没有此类帐号，则用户必须使用 `msgsend` 函数的 `alter_user=yes` 选项来执行消息传送操作。请参见 `msgsend` 参考页中的第 101 页的表 3-11。
- 获得了第 34 页的表 2-2 中列出的 MQ 授权。

注释 要执行实时数据服务的内置函数，仍然需要 Adaptive Server `messaging_role`。

此外，如果指定的结束点是队列管理器，则 `sp_msgadmin` 的 `'register, login'` 和 `'default, login'` 函数不允许注册 Adaptive Server 登录名，亦不允许创建缺省的 Adaptive Server 登录名。请参见第 57 页的 `sp_msgadmin`。

在 Adaptive Server 主机上安装 MQ 客户端

必须在 Adaptive Server 主机上安装 MQ 客户端软件。

Adaptive Server 会动态装载 MQ 客户端的共享库。表 2-1 显示了要将共享库安装到的位置。

表 2-1: MQ 客户端的共享库及其目录

平台	目录	库名
Solaris 32	<code>/opt/mqm/lib</code>	<code>libmqmcs.so</code> 、 <code>libmqic.so</code>
Solaris 64	<code>/opt/mqm/lib64</code>	<code>libmqmcs.so</code> 、 <code>libmqic.so</code>
Solaris AMD64	<code>/opt/mqm/lib64</code>	<code>libmqmcs.so</code> 、 <code>libmqic.so</code>
Linux 32	<code>/opt/mqm/lib</code>	<code>libmqic_r.so</code>
Linux AMD64	<code>/opt/mqm/lib64</code>	<code>libmqic_r.so</code>
HPPA 64	<code>/opt/mqm/lib64</code>	<code>libmqic.sl</code>

平台	目录	库名
HPIA 64	/opt/mqm/lib64	libmqic.so
AIX 64	/usr/mqm/lib64	libmqic_r.a(mqic_r.o)
Windows 32	c:\Program Files\IBM\WebSphere MQ\bin	MQIC32.DLL

- HP、HPIA、Linux、Linux AMD、Solaris 和 Solaris AMD — Adaptive Server 会从 /opt/mqm/lib 装载库，因此您无需为 MQ 设置 LD_LIBRARY_PATH。
- IBM — 先对 \$LIBPATH 进行设置以使其包含 /usr/mqm/lib64，再启动 Adaptive Server。
- Windows — 先对 %PATH% 进行设置以使其包含此库，再启动 Adaptive Server。

MQ 授权

要进行 MQ 配置，需要获得以下 UNIX 用户帐号和用户组（主体 / 组）授权：

表 2-2: MQ 主体 / 组及其授权

MQ 主体 / 组	MQ 授权
当前运行数据服务器可执行文件的 OS 登录名	对队列管理器的 connect、altusr、inq 和 setid
执行任意消息传送操作时备选用户的 OS 登录名	对队列的 inq
执行消息传送 read 操作时备选用户的 OS 登录名	对队列的 get
执行消息传送 browse 操作时备选用户的 OS 登录名	对队列的 browse
执行消息传送 send 操作时备选用户的 OS 登录名	对队列的 put
指定为 replyToQueue 的动态队列的备选用户的 OS 登录名	对队列管理器的 crt、dlt，以及对模型队列的 get、inq

注释 向远程队列发送消息时，WebSphere MQ 会检查传输队列上的用户鉴定。

如果在 msgsend 中指定 alter_user=yes，则备选用户为当前运行 Adaptive Server 的操作系统登录名。如果不指定 alter_user，则备选用户为当前执行 MQ 操作的 Adaptive Server 登录名。

查询 MQ 信息

如果运行的是 Adaptive Server 15.0.2 ESD #1 版本或更高版本，则可以通过使用 `sp_msgadmin` 存储过程的 `show` 选项，向 Adaptive Server 查询有关指定队列管理器中的 MQ 对象的以下信息：

- 该队列管理器的名称
- 属于该队列管理器的所有队列及其队列类型
- 属于该队列管理器的所有通道及其类型

❖ 为使用 `sp_msgadmin 'show'` 准备 WebSphere MQ

- 1 在 WebSphere MQ 中，启动要查询的队列管理器。
- 2 确保针对该队列管理器的 MQ 监听器正在运行。
- 3 启动队列管理器的命令服务器。
- 4 确保队列管理器中有一个名为 `SYSTEM.ADMIN.COMMAND.QUEUE` 的队列。

有关如何执行以上步骤的信息，请参见 IBM Publication Center (<http://www.elink.ibmink.ibm.com/publications/servlet/pbi.wss>) 中针对 WebSphere MQ 提供的文档。

SQL 参考

本章介绍可用于管理 Active Messaging 的全局变量、存储过程、函数和语法段。

主题	页码
与消息相关的全局变量	38
<msgheader> 和 <msgproperties> 文档	44
特定于 Adaptive Server 的消息属性	46
关键字	47
存储过程	47
内置函数	48
语法段	48
sp_configure 'enable real time messaging'	49
sp_engine	53
sp_msgadmin	57
msgconsume	68
msgpropcount	71
msgproplist	72
msgpropname	73
msgproptype	74
msgpropvalue	76
msgpublish	77
msgrecv	80
msgsend	93
msgsubscribe	122
msgunsubscribe	123
endpoint	125
option_string	128
sizespec	129
timespec	130

与消息相关的全局变量

应用程序可通过这些全局变量访问最近发送或接收的消息中的消息信息。

`@@msgcorrelation`

包含上次发送或读取的消息中的相关信息。

- (MQ) MQ 不会验证 `@@msgcorrelation` 是否包含可输出字符。应用程序不应依赖于采用当前服务器字符集的 `@@msgcorrelation`，而只应将 `@@msgcorrelation` 用作后续消息的选择器。如果要将 `@@msgcorrelation` 返回给应用程序，则应将其转换为 `varbinary` 数据类型。
- (JMS) `@@msgcorrelation` 包含最近发送或接收的消息中的 correlation ID。

`@@msgheader`

包含最近收到的消息中的消息标头信息。此变量采用 XML 格式。有关此格式的详细信息，请参见第 44 页的“`<msgheader>` 和 `<msgproperties>` 文档”。

设置 `@@msgheader` 的函数包括 `msgrecv` 和 `msgconsume`。

表 3-1 显示了 `@@msgheader` 全局变量在 MQ 环境中的有效字段名称及其说明。第 40 页的表 3-2 则列出了 `@@msgheader` 在 JMS 环境中的字段及其说明。

表 3-1: MQ `@@msgheader` 字段和说明

属性名	说明
<code>ApplIdentityData</code>	与标识相关的应用程序数据。
<code>ApplOriginData</code>	与源相关的应用程序数据。
<code>CodedCharSetId</code>	数值编码的字符集标识符。
<code>CorrelId</code>	相关标识符。
<code>Encoding</code>	消息中二进制数据的编码。Encoding 字段中标志的位掩码。
<code>DecimalEncoding</code>	这是消息有效负载中十进制数的编码，此合成属性派生自 <code>Encoding</code> 字段。如果： <ul style="list-style-type: none"> • <code>BigEndian</code> — 十进制数为 <code>big-endian</code>。 • <code>LittleEndian</code> — 十进制数为 <code>little-endian</code>。 • <code>Undefined</code> — 不将十进制数定义为 <code>big-endian</code> 或 <code>little-endian</code>。
<code>Feedback</code>	反馈状态。
<code>FloatEncoding</code>	这是有效负载中浮点数的编码，此合成属性派生自 <code>Encoding</code> 字段。如果： <ul style="list-style-type: none"> • <code>BigEndian</code> — 浮点数为 <code>big-endian</code>。 • <code>LittleEndian</code> — 浮点数为 <code>little-endian</code>。 • <code>Undefined</code> — 不将浮点数定义为 <code>big-endian</code> 或 <code>little-endian</code>。
<code>Format</code>	消息数据的格式名称，既可以是 MQ 定义的格式名称，也可以是应用程序定义的格式名称。
<code>GroupId</code>	组标识符。

属性名	说明
IntegerEncoding	有效负载中整数的编码，此合成属性派生自 <code>Encoding</code> 字段。如果： <ul style="list-style-type: none"> • <code>BigEndian</code> — 整数为 big-endian。 • <code>LittleEndian</code> — 整数为 little-endian。 • <code>Undefined</code> — 不定义整数的字节顺序。
LastMsgInGroup	如果： <ul style="list-style-type: none"> • <code>true</code> — 消息是组中的最后一条消息。 • <code>false</code> — 消息不是组中的最后一条消息。
MsgId	消息标识符。
MsgInGroup	如果： <ul style="list-style-type: none"> • <code>true</code> — 消息属于某个组。 • <code>false</code> — 消息不属于任何组。
MsgSeqNumber	消息序列号。
MessageType	十进制数格式的消息类型，但采用下列值时除外： <ul style="list-style-type: none"> • <code>request</code> — 消息为请求消息。 • <code>reply</code> — 消息为答复消息。 • <code>datagram</code> — 消息为数据报消息。 • <code>report</code> — 消息为报告消息。
NegativeActionNotification	这是一个派生自 <code>Report</code> 字段的合成属性。接收应用程序应生成一个负作用通知 (NAN) 报告。 <ul style="list-style-type: none"> • <code>yes</code> — 接收应用程序应生成 NAN 报告消息并将其发送给 <code>ReplyToQ</code> 和 <code>ReplyToQMgr</code> 字段中指定的目标。 • <code>no</code> — 接收应用程序不应生成 NAN 报告消息。
Persistence	消息的持久性。如果： <ul style="list-style-type: none"> • <code>persistent</code> — 消息为持久性消息。 • <code>non-persistent</code> — 消息不是持久性消息。
PositiveActionNotification	这是一个派生自 <code>Report</code> 字段的合成属性。接收应用程序应生成一个正作用通知 (PAN) 报告。如果： <ul style="list-style-type: none"> • <code>yes</code> — 接收应用程序应生成 PAN 报告消息并将其发送给 <code>ReplyToQ</code> 和 <code>ReplyToQMgr</code> 字段中指定的目标。 • <code>no</code> — 接收应用程序不应生成 PAN 报告消息。
PutAppName	这是将消息放入队列的应用程序的名称。
PutAppType	这是将消息放入队列的应用程序的类型。
PutDate	这是将消息放入队列的日期。
PutTime	这是将消息放入队列的时间。

属性名	说明
ReplyCorrelationId	这是一个派生自 Report 字段的合成属性。它表示要用作报告消息的相关 ID 的内容。 <ul style="list-style-type: none"> • msgId — 报告消息的相关 ID 应设置为已接收消息的消息 ID。 • correlationId — 报告消息的相关 ID 应设置为已接收消息的相关 ID。
ReplyMsgId	这是一个派生自 Report 字段的合成属性。它表示要用作报告消息的消息 ID 的内容。 <ul style="list-style-type: none"> • new — 使用新消息 ID 作为报告消息的消息 ID。 • original — 使用收到的消息 ID 作为报告消息的消息 ID。
ReplyToQ	答复队列的名称。
ReplyToQMgr	答复队列管理器的名称。
Report	消息中的报告选项。这是 MQRO * 标志的位图。
UserIdentifier	用户标识符。

表 3-2: JMS @@msgheader 字段和说明

属性名	说明
correlation	消息中的相关 ID
destination	消息中目标的名称
encoding	消息的编码名称
messageid	消息中的消息 ID
mode	消息的传递模式: <ul style="list-style-type: none"> • persistent • non-persistent
priority	消息优先级
redelivered	消息中的重新传递状态
replyto	消息中的 replyto 名称
timestamp	消息的时间戳
ttl	消息中的生存时间值，用于指示消息可以存在的时间长度
type	消息的类型

@@msgId

包含最近发送或接收的消息的 ID。

MQ 不会验证 **@@msgId** 是否包含可输出字符。应用程序不应依赖于采用当前服务器字符集的 **@@msgId**，而只应将 **@@msgId** 用作后续消息的选择器。如果要 **@@msgId** 返回给应用程序，则应将其转换为 **varbinary** 数据类型。

设置此变量的函数包括：

- (JMS) **msgsend**、**msgpublish**、**msgrecv**、**msgconsume**
- (MQ) **msgsend**、**msgrecv**

@@msgproperties

包含最近收到的消息中的消息属性信息。此变量采用 XML 格式。有关此格式的详细信息，请参见第 44 页的“<msgheader> 和 <msgproperties> 文档”。

- (JMS) **@@msgproperties** 是消息中的用户属性。
- (MQ):
 - 如果消息包含一个或多个 MQRF 标头，则将 MQRF 标头中的名称 — 值对插入 **@@msgproperties**。
 - 由于 MQRF 标头中的名称 — 值对可以具有非唯一名称，因此将为这些名称追加 “_ddd” 以使其变成唯一名称，其中 ddd 是为使其唯一而添加的整数扩展名。例如，对于具有下列主题的 MQRF 标头：

```
MQPSTopic    */baseball
MQPSTopic    */baseball/world series
MQPSTopic    */sports
```

在 **@@msgproperties** 中将得到下列属性：

```
MQPSTopic    */baseball
MQPSTopic_1  */baseball/world series
MQPSTopic_2  */sports
```

设置 **@@msgproperties** 的函数包括：

- (JMS) msgrecv、msgconsume
- (MQ) msgrecv

RF 标头中有则提取的值对包括：

MQPSCommand	MQPSIntData	MQPSReason	MQPSSubIdentity
MQPSCompCode	MQSParmId	MQPSReasonText	MQPSSubName
MQPSCorrelId	MQSPubOpts	MQPSRegOpts	MQPSSubUserData
MQPSDelOpts	MQSPubTime	MQPSSeqNum	MQPSSubUserData
MQPSErrorId	MQPSQMgrName	MQPSStreamName	MQPSTopic
MQPSErrorPos	MQPSQName	MQPSStringData	MQPSUserld

无法识别的名称将被忽略。如果 RF 标头中的值带有引号 (“ ”)，则删除两侧的引号。在带引号的值中，如果存在转义引号 (“ ”)，则将双引号替换为单引号。

@@msgreplyqmgr

仅限 MQ — 包含上次读取的消息的 ReplyToQmgr 名称。

@@msgreplytoinfo

包含用于直接发送和答复消息的主题名称 (*provider_url*、*queue_name*、*topic_name*、*user_name*) 或队列名称。可以是永久目标或临时目标。

设置 `@@msgreplytoinfo` 的函数包括：

- (JMS) `msgconsume`、`msgpublish`、`msgrecv`、`msgsend`
- (MQ) `msgrecv`、`msgsend`

仅限 JMS — 口令不包括在 `@@msgreplytoinfo` 的值中。若要将此目标用作后续 `msgsend` 或 `msgrecv` 调用的参数，请添加 `password=<your password>`。

仅限 MQ — 可包含 `remote_qmgr` 的语法；`@@msgreplytoinfo` 显示请求 / 答复消息传送，它使用 `@@msgcorrelation` 表明对集群队列管理器的支持：

例如，假设某台 Adaptive Server 连接至 MASTER_MSCAI 队列管理器，并在将 `replyToQueue` 属性指定为 MASTERQ 的情况下，向 SLAVE_MSCAI 远程队列管理器上的 Q1 发送一条消息。在发送 `msgsend` 后，其值将变为 `@@msgreplytoinfo` 的值：

```
select msgsend('d','ibm_mq:CH1/tcp/host1(1105)?
qmgr=MASTER,remote_qmgr=SLAVE,queue=Q1,alter_user=yes',
  message property 'replyToQueue=MASTERQ')
go
select @@msgreplytoinfo
go
```

```
IBM_MQ:CH1/tcp/host1(1105)?qmgr=MASTER,queue=MASTERQ
```

另一台 Adaptive Server 连接至队列管理器 SLAVE，并收到之前由 Q1 发送的消息。`@@msgreplytoinfo` 全局变量随后会包括 `remote_qmgr` 的语法，这样本例中的远程队列就变为答复队列。

```
select msgrecv('ibm_mq:CH2/tcp/host2(4810)?
qmgr=SLAVE,queue=Q1,alter_user=yes', option 'timeout=100')
go
select @@msgreplytoinfo
go
```

```
ibm_mq:CH2/tcp/host2(4810)?qmgr=SLAVE,remote_qmgr=MASTER,queue=MASTERQ
```

注释 在使用包含 `remote_qmgr` 语法的 `@@msgreplytoinfo` 发送答复消息 `msgrecv` 时，答复消息能否抵达正确的远程队列管理器取决于 WebSphere MQ 的配置情况。请参见第 25 页的“使用远程队列对象”。

`@@msgschema`

仅限 JMS — 包含消息的 `schema` 或空值。包含 Adaptive Server 属性 `ase_message_body_schema` 的值。请参见有关 `msgsend` 和 `msgpublish` 中的 `schema` 选项的说明。

设置 `@@msgschema` 的函数包括：`msgsend`、`msgpublish`。

`@@msgstatus`

包含服务提供程序例外的整数错误代码；如果上一操作未引发例外，则包含零。

设置 `@@msgstatus` 的函数包括：`msgsend`、`msgpublish`、`msgrecv`、`msgconsume`。

`@@msgstatusinfo`

如果上一 `msgsend`、`msgpublish`、`msgrecv` 或 `msgconsume` 引发了例外，则包含列出服务提供程序例外的错误消息；如果是一个空字符串，则包含零。

(MQ) 包含上一消息处理操作的提供程序错误消息。MQ Client Library 不提供已本地化的错误消息，因此显示的错误消息将类似于以下形式：

```
MQ API call failed with reason code '%s' (%d)
```

“%s”将替换为 MQ 原因代码的 MQ 助记符。

“%d”将替换为十进制的 MQ 原因代码。

设置此变量的函数包括：

- (JMS) `msgsend`、`msgpublish`、`msgrecv`、`msgconsume`。
- (MQ) `msgsend`、`msgrecv`。

`@@msgtimestamp`

包含上次发送的消息中的 `timestamp`。

设置该变量的函数为：`msgsend`、`msgpublish`。

示例

示例 1 仅限 MQ — 显示使用 `@@msgreplytoinfo` 和 `@@msgcorrelation` 进行请求 / 答复消息处理：

会话 1 (请求方)	会话 2 (接收方)
<pre>select msgsend('sender mmessage', 'ibm_mq:channel1/TCP/host1(5678)' + '?qmgr=QM1' + ',queue=Q100', option 'msgType=request', message property 'correlationId=0x123456' + 'replyToQueue=Q200')</pre>	

会话 1 (请求方)	会话 2 (接收方)
	<pre>select msgrecv('ibm_mq:channel1/TCP/host1(5678)' + '?qmgr=QM1' + ',queue=Q100') select msgsend('receiver reply', @msgreplytoinfo, option 'msgType=reply' message property 'correlationId=' + @@msgcorrelation)</pre>
<pre>select msgrecv('ibm_mq:channel1/TCP/host1(5678)' + '?qmgr=QM1' + ',queue=Q200' option 'timeout=30ss', + 'correlationID=0x123456')</pre>	

在下面的示例中：

- 1 会话 1 向 Q100 发送请求消息，并预期 Q200 发送答复消息。它将相关设置为 0x123456。
- 2 会话 2 读取 Q100 中的消息，然后向 Q200 发送答复消息，并将相关指定为 0x123456。答复队列从刚才读取的消息中获取。
- 3 会话 1 读取 Q200 中的答复消息，并只获取相关为 0x123456 的消息。

用法

- 这些全局变量的数据类型为 char，长度为 16384。
- 使用 rtrim 可删除尾随空白。

<msgheader> 和 <msgproperties> 文档

说明

包含返回消息的标头和属性的 XML <msgheader> 和 <msgproperties> 文档需要设置全局变量 @@msgheader 和 @@msgproperties。本节说明这些文档的格式。

对于名为 PROPERTY_1、PROPERTY_2 等的属性，<msgheader> 和 <msgproperties> 文档的一般格式为以下语法节中的 DTD 模板所描述的格式。

语法

```

<!DOCTYPE msgheader [
<!ELEMENT msgheader EMPTY>
<!ATTLIST property_1 CDATA>
<!ATTLIST property_2 CDATA>
etc.
<!DOCTYPE msgproperties [
<!ELEMENT msgproperties EMPTY>
<!ATTLIST property_1 CDATA>
<!ATTLIST property_2 CDATA>

```

示例 这些示例显示两个 `select` 语句的 `<msgheader>` 或 `<msgproperties>` 文档:

```

select msgsend('Sending message with properties',
              'my_jms_provider?queue=queue.sample',
              message property 'color=red, shape=square')

select msgrecv('my_jms_provider?queue=queue.sample')

select rtrim (@@msgproperties)

<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<msgproperties
  RTMS_MSGBODY_FORMAT='&apos;string&apos;';
  ASE_RTMS_CHARSET='1'
  ASE_RTMS_VERSION='&apos;1.0&apos;';
  ASE_VERSION='&apos;12.5.0.0&apos;';
  shape='&apos;square&apos;';
  color='&apos;red&apos;'; >
</msgproperties>

select rtrim (@@msgheader)

<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<msgheader
  type='&apos;null&apos;';
  timestamp='1080092021000'
  replyto='&apos;queue.sample&apos;';
  redelivered='false'
  priority='4'
  messageId='&apos;ID:E4JMS-SERVER.73018656B39:1&apos;';
  ttl='0'
  destination='&apos;queue.sample&apos;';
  mode='2'
  correlation='&apos;null&apos;';
  encoding='&apos;null&apos;'; >
</msgheader>

```

用法

- 对于消息标头和消息属性的每个属性，指定消息的 `<msgheader>` 或 `<msgproperties>` 文档都包含一个特性。该特性的名称为相应属性的名称，该特性的值则为相应属性的字符串值。

- `<msgheader>` 或 `<msgproperties>` 文档中的属性值由 XML 实体替换。`msgpropvalue` 和 `msgpropname` 可将 XML 实体隐式替换为属性值。
- 由 `msgrecv` 或 `msgconsume` 生成的 `<msgheader>` 或 `<msgproperties>` 文档具有指定属性字符集的 XML 声明。

特定于 Adaptive Server 的消息属性

（仅限 JMS）为帮助执行调试和监控等任务，JMS 消息的属性部分中包括特定于 Adaptive Server 的预定义属性。这些属性通常处理源自另一台 Adaptive Server 的消息或可能有助于调试的消息。

其中的许多消息属性只有在运行 `diagserver` 或打开某些跟踪标志时才会包括在内。所有以“ASE_”开头的属性均为保留属性；这些属性不能使用 `msgsend` 或 `msgpublish` 进行设置。表 3-3 讲述了这些消息属性。

表 3-3: 特定于 Adaptive Server 的消息（针对 JMS）

属性	说明	使用场合
ASE_RTMS_CHARSET	已发送数据的字符集编码。	始终
ASE_MSGBODY_SCHEMA	描述消息正文的 schema 或空值。只有用户将消息 schema 作为 <code>msgsend</code> 的一部分发送时，此 schema 才不是空值。如果 ASE_MSGBODY_FORMAT 为 <code>xml</code> ，此属性将包含描述有效负载的 XML schema。即使此值超过 16 K，也不会截断此 schema。	始终
ASE_MSGBODY_FORMAT	消息正文的格式： <code>xml</code> 、 <code>string</code> （采用服务器字符集时）、 <code>binary</code> 和 <code>unicode</code> （在网络顺序中为 <code>unichar</code> ）。	始终
ASE_ORIGIN	源 Adaptive Server 的名称。	与 <code>diagserver</code> 一起使用
ASE_RTMS_VERSION	使用 Active Messaging 的 Adaptive Server 的版本。	始终
ASE_SPID	发送消息的 SPID。	与 <code>diagserver</code> 一起使用
ASE_TIMESTAMP	显示消息发送时间的 Adaptive Server 时间戳。	与 <code>diagserver</code> 一起使用
ASE_VERSION	发布消息的 Adaptive Server 的版本。	始终
ASE_VERSIONSTRING	Adaptive Server 的版本字符串。提供有关平台和版本类型等方面的信息。对调试十分有用。	与 <code>diagserver</code> 一起使用

注释 上面显示的这些属性仅供参考。它们在以后可能会发生更改。

关键字

表 3-4 显示特定于 ASE Active Messaging 的关键字，以及可合法使用这些关键字的函数。

表 3-4: ASE Active Messaging 中的双重关键字和三重关键字

JMS 或 MQ	关键字	使用关键字的合法命令和函数
同时适用	message header	select msgsend(..., message header,...) select msgpublish(...,message header,...)
同时适用	message property	select msgsend(..., message property,...) select msgpublish(...,message property,...)
JMS	message selector	select msgrecv(...,message selector,...) select msgconsume(...,message selector,...)
JMS	with retain	select msgunsubscribe(...,with retain,...)
JMS	with remove	select msgunsubscribe(...,with remove,...)
同时适用	transactional messaging none	set transactional messaging none
同时适用	transactional messaging simple	set transactional messaging simple
同时适用	transactional messaging full	set transactional messaging full

存储过程

用于 ASE Active Messaging 的存储过程包括：

- 第 49 页的 [sp_configure 'enable real time messaging'](#)
- 第 53 页的 [sp_engine](#)
- 第 57 页的 [sp_msgadmin](#)

`sp_msgadmin` 及其选项不配置或管理基础消息提供程序。例如，您仍须在消息传送提供程序级别创建、删除和访问队列与主题。

注释 `sp_addexclass` 的 `anyengine` 和 `lastonline` 参数中不包含 MQ Q 引擎。

内置函数

本章的这一节与内置函数有关，介绍用于管理 Active Messaging 的 SQL 函数，以及选项字符串的一般格式。有关特定于 Adaptive Server 的消息属性，请参见第 46 页的表 3-3。本章中的 SQL 函数可执行下列操作：

- 向队列发送消息以及从队列接收消息
- 发布、预订和使用与消息主题有关的消息
- 处理消息属性

本章中列出的函数及其所在页码如下：

- [第 68 页的 msgconsume](#)
- [第 71 页的 msgpropcount](#)
- [第 72 页的 msgproplist](#)
- [第 73 页的 msgpropname](#)
- [第 74 页的 msgproptype](#)
- [第 76 页的 msgpropvalue](#)
- [第 77 页的 msgpublish](#)
- [第 80 页的 msgrecv](#)
- [第 93 页的 msgsend](#)
- [第 122 页的 msgsubscribe](#)

语法段

本节与语法段有关，介绍用于管理实时消息传送的 SQL 语法和约束的各个部分。

本章中列出的语法段及其所在的页码如下：

- [第 125 页的 endpoint](#)
- [第 128 页的 option_string](#)
- [第 129 页的 sizespec](#)
- [第 130 页的 timespec](#)

sp_configure 'enable real time messaging'

说明	启用或禁用实时消息传送，或者显示当前的消息传送配置。
语法	<pre>sp_configure "enable real time messaging" [, enable_or_disable] [, rtm_provider drop instance] [, instance_name]</pre>
参数	<p><i>enable_or_disable</i> 指定是否启用 "real time messaging" 选项。有效值包括：</p> <ul style="list-style-type: none"> • 1 — 启用实时消息传送。 • 0 — 禁用实时消息传送。 <p>如果省略，则返回当前的 "real time messaging" 配置。</p> <p><i>rtm_provider</i> 指定要启用或禁用的 Active Messaging 提供程序的类型。指定 JVM 和集群服务器时需使用此参数。有效值包括：</p> <ul style="list-style-type: none"> • eas_jms — 启用或禁用 "real time messaging"（仅限 EAServer）。 • ibm_mq — 启用或禁用 "real time messaging"（仅限 IBM MQ）。 • sonicmq_jms — 启用或禁用 "real time messaging"（仅限 SonicMQ JMS）。 • tibco_jms — 启用或禁用 "real time messaging"（仅限 Tibco JMS）。 <p><i>drop instance</i> 删除某一实例的与消息传送相关的配置选项。</p> <p><i>instance_name</i> 创建集群服务器环境时指定的实例名称。如果未指定此选项，则由当前的实时消息传送配置指定该全集群范围的选项。</p>
示例	<p>示例 1 为所有提供程序启用实时消息传送：</p> <pre>sp_configure "enable real time messaging",1</pre> <p>随后可通过以下语法禁用此配置：</p> <pre>sp_configure "enable real time messaging",0</pre> <p>示例 2 仅为 MQ 启用实时消息传送：</p> <pre>sp_configure "enable real time messaging", 1 ,ibm_mq</pre> <p>随后可通过以下语法禁用此配置：</p> <pre>sp_configure "enable real time messaging", 0, ibm_mq</pre>

用法

使用此存储过程不会覆盖以前的设置。例如，如果先启用 `tibco_jms`，再运行此存储过程来启用 MQ，则 MQ 和 `tibco_jms` 都将变为启用状态。禁用 `tibco_jms` 不会影响 MQ，后者会继续处于启用状态。

只有在正确安装和设置以下内容后，`enable_or_disable` 参数才有效：

- 与所用平台相应的 `LD_LIBRARY_PATH`
- 提供程序 DLL 库
- SYBASE 许可证
- 产品 CD 中的 SYBASE 接口库

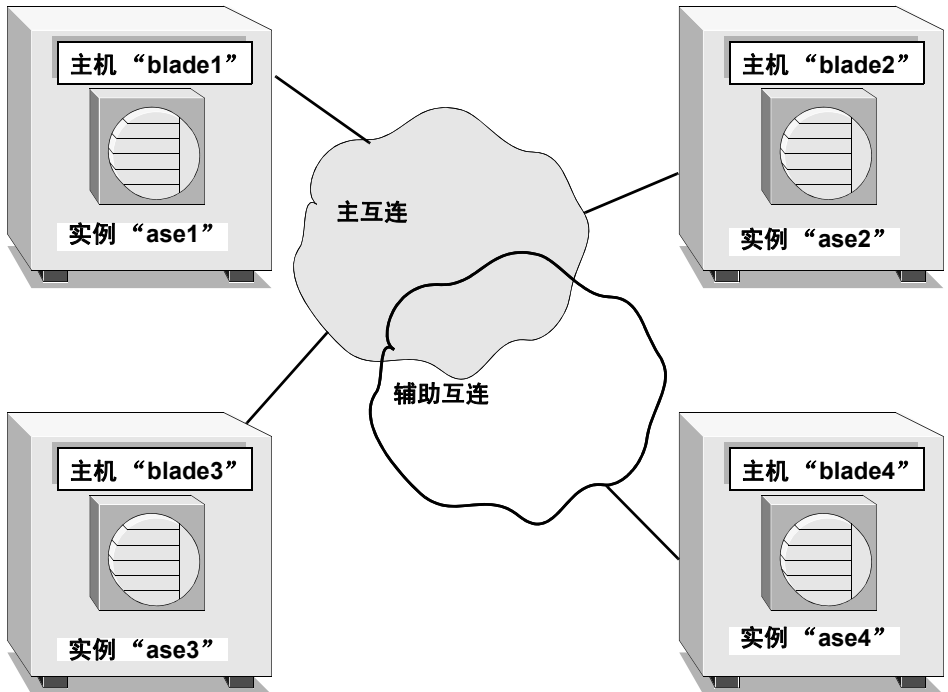
有关路径和文件名的详细信息，请参见 *Adaptive Server Installation Guide*（《Adaptive Server 安装指南》）。

在集群环境中使用 `sp_configure`

Cluster Edition 允许通过配置使多台 Adaptive Servers 作为一个共享磁盘集群运行。多台计算机连接到一组共享磁盘和一个高速私有互连（例如千兆以太网），使 Adaptive Server 可以通过多个物理和逻辑主机进行扩展。

在以下示例使用的集群系统中，客户端连接至名为“mycluster”的共享磁盘集群，该集群包括“ase1”、“ase2”、“ase3”和“ase4”实例，这些实例分别运行在“blade1”、“blade2”、“blade3”和“blade4”计算机上。在这些示例中，每个节点驻留一个实例。

图 3-1: sp_configure 示例中描述的集群环境。



示例 1 为集群中所有实例上的所有 Active Messaging 提供程序启用实时消息传送:

```
sp_configure "enable real time messaging", 1
```

示例 2 如果客户端登录 "ase1", 则禁用仅 IBM MQ 的实例 "ase1":

```
sp_configure "enable real time messaging", 0, "ibm_mq", "ase1"
```

为 "ase1" 生成的特定于实例的配置选项值为 26。"real time" 的配置值为 byte, 它具有不同的位, 用于表示不同的实时功能:

- 0x1 — 全部启用。
- 0x2 — 启用 tibco_jms。
- 0x4 — 启用 ibm_mq。
- 0x8 — 启用 eas_jms。
- 0x10 — 启用 sonicmq_jms。

示例 3 显示实例 “ase1” 上特定于实例的 `enable real time messaging` 配置选项状态。其值为 26:

```
sp_configure "enable real time messaging", null, null, ase1
```

示例 4 删除实例 “ase1” 上特定于实例的 `enable real time messaging` 配置选项。运行此过程后，“ase1” 将开始使用全集群范围的 `enable real time messaging` 配置选项，并且状态变为 1。

```
sp_configure "enable real time messaging", 0, "drop instance", "ase1"
```

运行 `drop instance` 时，新的配置值与当前运行的值相同。

示例 5 显示全集群范围的 `enable real time messaging` 配置选项状态。其值为 1:

```
sp_configure "enable real time messaging"
```


sp_engine

说明	可用于使 Q 引擎变为联机或脱机。
语法	<pre>sp_engine "online offline can_offline shutdown q_online q_offline q_can_offline q_shutdown", [engine_id]</pre>
参数	<p>can_offline 返回有关是否可以使某个引擎变为脱机的信息。如果无法使该引擎变为脱机，则显示阻止该引擎变为脱机的 Adaptive Server 会话的 spid。此参数不能用于指定 Q 引擎。</p> <p>engine_id 引擎的 ID。 指定的引擎类型必须与命令（online、q_online 等）匹配。例如，不能使用 q_offline 指定非 Q 引擎，也不能使用 offline 指定 Q 引擎。 此参数是 offline、q_offline、can_offline、q_can_offline、shutdown 和 q_shutdown 所必需的。 此参数不是 online、q_online 所必需的。</p> <p>online 使引擎变为联机。sp_configure "max online Q engines" 的值必须大于当前联机 Q 引擎的数量。必须使用引号，因为 online 是一个保留关键字。此参数不能用于指定 Q 引擎。</p> <p>offline 使引擎变为脱机。也可以使用 engine_id 指定要使其变为脱机的引擎。此参数不能用于指定 Q 引擎。</p> <p>q_can_offline 返回有关是否可以使某个 Q 引擎变为脱机的信息。如果无法使该引擎变为脱机，则显示阻止该引擎变为脱机的 Adaptive Server 会话的 spid。必须使用 engine_id 指定是否可以将 Q 引擎变为脱机。</p> <p>q_offline 使 Q 引擎变为脱机。必须使用 engine_id 指定要使其变为脱机的引擎。</p> <p>q_online 使下一 Q 引擎变为联机。</p> <p>q_shutdown 强制 Q 引擎变为脱机。如果有任何任务与此引擎关联，则在等待 5 分钟后注销这些任务。必须使用引号，因为 shutdown 是一个保留关键字。必须使用 engine_id 指定是否可以关闭该 Q 引擎。</p>

shutdown

强制引擎变为脱机。如果有任何任务与此引擎关联，则在等待 5 分钟后注销这些任务。必须使用引号，因为 **shutdown** 是一个保留关键字。此参数不能用于指定 Q 引擎。

示例**示例 1 手动使 Q 引擎变为联机：**

```
sp_engine 'q_online'
go

(return status=0)

02:00000:00000:2005/06/08 12:52:21.09 kernel Network and device connection limit is
1014.
02:00000:00000:2005/06/08 12:52:21.24 server Initialized Unilib version 7.2.
02:00000:00000:2005/06/08 12:52:21.24 kernel Q engine 2, os pid 20025 online
02:00000:00000:2005/06/08 12:52:21.33 kernel LDAP dynamic libraries successfully
loaded.
02:00000:00000:2005/06/08 12:52:21.38 kernel IBM MQ dynamic libraries successfully
loaded.
```

示例 2 使 Q 引擎变为脱机：

```
1> select engine, status from sysengines
2> go

engine status
-----
      0 online
      1 online_q
      2 online_q
(3 rows affected)

1> sp_engine 'q_offline', 1
2> go

(return status = 0)
00:00000:00000:2005/06/08 12:55:54.25 kernel engine
2, os pid 20025 offline

1> select engine, status from sysengines
2> go

engine status
-----
      0 online
      1 online_q
(2 rows affected)
```

示例 3 检查是否可以使某个 Q 引擎变为脱机:

```

1> select engine, status from sysengines
2> go

engine status
-----
          0 online
          1 online_q
(2 rows affected)

1> sp_engine 'q_can_offline', 1
2> go

spid: 13 has outstanding rtms-connection connections.

```

示例 4 关闭 Q 引擎:

```

1> select engine, status from sysengines
2> go

engine status
-----
          0 online
          1 online_q
(2 rows affected)

1> sp_engine 'q_shutdown', 1
2> go

(return status = 0)

1> select engine, status from sysengines
2> go

engine status
-----
          0 online

(1 row affected)

```

用法

- `online`、`offline`、`can_offline` 和 `shutdown` 仅影响非 Q 引擎。如果使用这些参数指定了 Q 引擎，则会显示错误。
- `q_online`、`q_offline`、`q_can_offline` 和 `q_shutdown` 仅影响 Q 引擎。如果使用这些参数指定了非 Q 引擎，则会显示错误。
- 不能关闭引擎 0 或使其变为脱机。
- 使用以下查询可确定某一引擎的状态以及当前处于联机状态的引擎:

```

select engine, status from sysengines
where status = "online"

```

- 只有 `max online Q engines` 大于当前处于 `online` 状态的引擎数量，并且有足够的可用 CPU 支持添加任何额外的引擎时，才能使引擎变为联机。
- 如果有服务器进程与该引擎关联，则 `engine offline` 可能会失败或无法立即生效。
- 在集群环境中，`sp_engine` 仅适用于本地实例的引擎。

权限

您必须是系统管理员才能使引擎变为联机或脱机。

sp_msgadmin

说明

配置和管理与消息传送相关的信息。

语法

```

sp_msgadmin 'config', ['jvmlogging', logging_level
    | 'jvmpropertyfile', filepath
    | 'jvmlogfile', filepath
    | 'jvmmaxthreads', thread_number
    | 'jvmmintthreads', thread_number
    | 'jvmthreadtimeout', thread_timeout
    | 'jvm' , jvm_parameter]

sp_msgadmin 'default', 'login', provider_name, provider_login,
    provider_password

sp_msgadmin 'help'
    [, 'list' | 'register' | 'default' | 'remove']

sp_msgadmin 'list',
    [| 'login'[, provider_name, [login_name]
    | 'provider' [, provider_name]
    | 'subscription' [, subscription_name]]

sp_msgadmin 'register',
    ['provider', provider_name, provider_class,
    messaging_provider_URL
    | 'login', provider_name, local_login, provider_login,
    provider_password [, role_name]
    | 'subscription', subscription_name, endpoint[, selector
    [, delivery_option [, durable_name, client_id]]]]

sp_msgadmin 'remove',
    ['provider', provider_name
    | 'login', provider_name, local_login [, role]
    | 'subscription', subscription_name

sp_msgadmin 'show',
    showtype, provider[, options_clause]

```

参数

sp_msgadmin 'config'

允许为 Java 虚拟机 (JVM) 或密钥存储库文件路径 (供 Adaptive Server 使用 MQ SSL) 指定各种配置。配置的值在重新启用 ASE Active Messaging 后生效。sp_msgadmin 'config' 的选项包括:

- 'jvmlogging', *logging_level* — 允许将消息传送服务配置为仅显示代码中高于已配置级别的跟踪信息。

logging_level 指定使用 Apache log4j 记录系统的级别。*logging_level* 的值包括:

- 'all' — 返回代码中的所有跟踪信息。
- 'debug' — 返回 JVM 调试信息。
- 'fatal' — 返回 JVM 致命信息。
- 'off' — 关闭记录。
- 'info' — *logging_level* 的缺省值, 用于返回信息级别的日志信息。
- 'error' — 仅返回错误日志信息。

请参见 Apache log4j 网站, 位于 <http://logging.apache.org/log4j/>。

- 'jvmpropertyfile', *filepath* — 指定 JVM 用于配置的属性文件。

filepath 可以使属性文件的任何有效路径, 包括使用环境变量。属性文件的缺省值为 *\$\$SYBASE/\$\$SYBASE_ASE/lib/rtms.properties*。

- 'jvmlogfile', *filepath* — 定义 JVM 用于配置的日志文件的路径。

JVM 的日志信息显示在控制台上, 并写入单个日志文件。每当日志文件达到最大大小 (5MB) 时, JVM 都会自动创建一个新的日志文件, 并在文件名末尾追加一个新的编号 (例如 XXX.2、XXX.3, 依此类推)。

JVM 日志文件 *filepath* 的缺省值为 *\$\$SYBASE/\$\$SYBASE_ASE/rtms.log*。

当在集群环境中启动 Java Active Messaging 服务器时, 实际日志文件为该值与 @@nodename 的组合。例如, 如果对节点 "s1" 运行 sp_msgadmin, 则实际 JVM 日志文件为

\$\$SYBASE/\$\$SYBASE_ASE/jrtms_s1.log:

```
1> sp_msgadmin 'config', 'jvmlogfile', '$$SYBASE/$$SYBASE_ASE/jrtms.log'
```

如果已配置的 JVM 日志文件：

- 具有文件扩展名（例如 `$$SYBASE/$$SYBASE_ASE/jrtms.log`，其中 `jrtms.log` 的文件名包含 `log` 文件扩展名），则实例 “ase1” 的实际文件名为 `$$SYBASE/$$SYBASE_ASE/jrtms_ase1.log`。
- 不具有文件扩展名（例如 `$$SYBASE/$$SYBASE_ASE/jrtms`，其中文件名是没有扩展名的 `jrtms`），则实例 “ase1” 的实际文件名为 `$$SYBASE/$$SYBASE_ASE/jrtms_ase1`。
- 'jvmmxthreads', *thread_number* — 指定要在 JVM 服务器的线程池同时运行的 Java 线程的最大数量。
thread_number 的值必须大于 `jvminthreads` 的值。缺省值为 10。
- 'jvminthreads', *thread_number* — 指定要在 JVM 服务器的线程池同时运行的 Java 线程的最小数量。 *thread_number* 的值可以是 0 或更多，但必须小于 `jvmmxthreads` 的值。缺省值为 0。
- 'jvmthreadtimeout', *thread_timeout* — 允许线程在非活动状态达到指定时段后自动销毁。
thread_timeout 是销毁线程前经过的秒数。缺省值为 600（10 分钟）。
- 'jvm', *jvm_parameter* — 定义启动 JVM 时传递给 Java 的参数。
jvm_parameter 是任意有效 Java 参数字符串的名称。缺省值为 “-Xmx500m”，这是一个通用的 Java 标志，用于指定 Java 在已分配 RAM 为 500 MB 时启动。有关 Java -Xmx 标志的详细信息，请参见 Java 网站，位于 <http://java.sun.com>。
- 'ibmmq_keystore', *keystore_name* — 配置密钥存储库文件路径，以便 Adaptive Server 通过 SSL 在自身与 WebSphere MQ 之间收发消息。
keystore_name 是存储密钥和证书的密钥数据库文件的位置。

sp_msgadmin 'default'

指定缺省值。对于 `sp_msgadmin 'list'`，则列出用于为指定的消息提供程序指定缺省登录名的语法。选项包括：

- 'login' — 在与 'default' 一起使用时，可指定缺省登录名。

注释 如果 endpoint 为 MQ 队列管理器，则不能使用 `sp_msgadmin 'default', 'login'`。

- *provider_name* — 要注册的消息传送提供程序，长度可达 30 个字符。
- *provider_login* — *local_login* 在连接消息提供程序时映射到的消息传送提供程序的登录名。 *provider_login* 也是该提供程序在使用 *provider_name* 发送或接收消息时使用的缺省登录名。
- *provider_password* — *provider_login* 的口令。

注释 （仅限集群环境）如果使用 `sp_msgadmin default` 在集群环境中定义缺省登录名，则可对整个集群使用该配置。

sp_msgadmin 'help', 'list' | 'register' | 'default' | 'remove'

提供有关 `sp_msgadmin` 或其参数的语法信息。

sp_msgadmin 'list'

列出有关消息提供程序、登录名或预订的语法信息：

- 'login'[, *provider_name*, [*login_name*]] — 列出有关特定消息传送提供程序登录名映射或所有消息传送提供程序登录名的信息。
- 'provider'[, *provider_name*] — 指定消息提供程序，并列出有关特定消息传送提供程序或所有消息提供程序的信息。
- 'subscription'[, *subscription_name*] — 列出有关特定预订或所有预订的信息。

sp_msgadmin 'register'

注册某个消息传送提供程序、登录名或预订。选项包括：

- **sp_msgadmin 'register' provider** — 注册消息传送提供程序，其中：
 - *provider_name* — 消息传送提供程序的名称。
 - *provider_class* — 要添加的消息传送提供程序的类。有效值包括：
 - EAS_JMS
 - TIBCO_JMS
 - IBM_MQ
 - SONIC_MQ
 - *messaging_provider_URL* — 要注册的消息传送提供程序的 URL。
- **sp_msgadmin 'register' 'login'** — 注册登录名映射，其中：
 - *provider_name* — 以前注册的某个提供程序的名称，长度可达 30 个字符。
 - *local_login* — 映射到本地登录名的 Adaptive Server 登录名。
 - *provider_login* — 连接消息提供程序时由 *local_login* 映射到的消息传送提供程序的登录名。
 - *provider_password* — *provider_login* 的消息传送提供程序口令。
 - *role_name* — 一个 SQL 角色名称。如果指定 *role_name*，则会忽略 *local_login*，并且 *provider_login* 和 *provider_password* 将应用于 *role_name*。

注释 如果 endpoint 为 MQ 队列管理器，则不能使用 `sp_msgadmin 'register', 'login'`。

- **sp_admin 'register' 'subscription'** — 注册预订，其中：
 - *subscription_name* — 预订名称。
 - *endpoint* — 预订所指向的主题。有关 *endpoint* 的信息，请参见 [第 93 页的 msgsend](#)。

注释 如果 endpoint 为 MQ 队列管理器，则不能使用 `sp_msgadmin 'register', 'subscription'`。

- **selector** — 允许客户端选择感兴趣消息的消息过滤器。有关过滤器的信息，请参见第 80 页的 **msgrecv**。
- **delivery_option** — 指定 SQL 会话是否可以使用它所发布的消息。有效值包括：
 - **local** — SQL 会话可以使用它所发布的消息。
 - **nonlocal** — SQL 会话不能使用它所发布的消息。
 - **null** — 假定值为 **local**。
- **durable_name** — 是一个字符串值。请参见 **client_id**。
- **client_id** — 消息传送提供程序用于将预订标识为永久性预订的标识。**client_id** 是一个字符串值。对于 **client_id** 和 **durable_name**，如果指定了其中之一，则还必须指定另一个，才能将预订指定为永久性预订。否则，预订为非永久性预订。

client_id 和 **durable_name** 共同标识消息提供程序的永久性预订，其组合必须是唯一的。

client_id 的唯一性涵盖了消息传送提供程序。JMS 每次只允许连接一个特定的 **client_id**。例如，如果某一应用程序已具有使用指定 **client_id** 的持久性预订，则要与此应用程序同时连接的其它应用程序将不能指定同一 **client_id**。

即使在未连接客户端时，持久性预订也能存在，因为消息传送提供程序会保存收到的消息。

只有在连接客户端后，非持久性预订才能存在。如果未连接客户端，消息传送提供程序将放弃收到的消息。

注释 （仅限集群环境）如果使用 **sp_msgadmin 'register'** 在集群环境中注册提供程序、登录名和预订信息，则此项注册将应用于整个集群。

sp_msgadmin 'remove'

删除某一消息提供程序、登录名或预订。

- **'provider', provider_name** — 删除之前使用以下语法定义的消息传送提供程序：

```
sp_msgadmin 'register', 'provider', provider_name
```

provider_name 是要删除的消息传送提供程序的别名。

- 'login', *provider_name*, *local_login* [, *role*] — 删除之前在 Adaptive Server 登录名与服务提供程序登录名（由以下调用定义）之间创建的映射：

```
sp_msgadmin 'register', 'login', local_login, ...
```

其中：

- *local_login* — 映射到本地登录名的 Adaptive Server 登录名。
- *role* — 角色。
- 'subscription', *subscription_name* — 删除之前使用以下语法创建的预订：

```
sp_msgadmin 'register' 'subscription', subscription_name, ...
```

sp_msgadmin 'show'

需要 Adaptive Server 15.0.2 ESD #1 版或更高版本，可显示有关指定队列管理器上的某些 MQ 对象的信息，其中：

- *showtype* — 允许指定要显示的 WebSphere MQ 进程或对象：
 - *qmgr* — 队列管理器的名称。
 - *queues* — 属于该队列管理器的所有队列及其类型。
 - *channels* — 属于该队列管理器的所有通道及其类型。
- *provider* — 指定消息传送提供程序。请使用完整路径格式，详见第 125 页的 [endpoint](#)。
- *option_string* — 选项列表，如表 3-5 所示。

表 3-5: option_string 类型和值

类型	值	缺省值	说明
timeout	介于 0 与 $(2^{31}-1)$ 之间的 timespec 值	30000（30 秒）	指定 WebSphere MQ 管理接口应等待每个答复消息的最长时间（单位为毫秒）。
replyqueue	string	无	命令服务器将答复消息返回给队列。如果不定义此选项，命令服务器会将消息返回给通过打开 SYSTEM.DEFAULT.MODEL.QUEUE 创建的动态队列。

示例

示例 1 (JMS) 记录 JVM 的级别：

```
sp_msgadmin 'config', 'jvmlogging', 'info'
```

示例 2 (JMS) 将 /usr/l.prop 指定为属性文件：

```
sp_msgadmin 'config', 'jvmpropertyfile', '/usr/l.prop'
```

示例 3 (JMS) 将日志文件路径定义为 `$$SYBASE/$$SYBASE_ASE/rtms.log`:

```
sp_msgadmin 'config', 'jvml logfile', '$$SYBASE/$$SYBASE_ASE/rtms.log'
```

示例 4 (JMS) 将 JVM 服务器线程池中的最大线程数指定为 100:

```
sp_msgadmin 'config', 'jvmmaxthreads', 100
```

示例 5 (JMS) 将 10 分钟指定为闲置线程自动销毁前的时间长度:

```
sp_msgadmin 'config', 'jvmthreadtimeout', 600
```

示例 6 (JMS) 使用 `-Xmx500m` 标志指定在 RAM 为 500 MB 时启动 JVM:

```
sp_msgadmin 'config', 'jvm', '-Xmx500m'
```

示例 7 (JMS) 注册类为 `EAS_JMS` 且 URL 为 `iiop://localhost:7222` 的“`eas_1`”消息提供程序:

```
sp_msgadmin 'register', 'provider',  
            'eas_1', 'eas_jms', 'iiop://localhost:7222'
```

示例 8 (JMS) 指定在使用指定消息传送提供程序发送或接收消息时适用于所有未映射 Adaptive Server 登录名的缺省登录名:

```
sp_msgadmin 'default', 'login', 'my_eas', 'eas_user', 'eas_password'
```

注释 必须先通过调用 `sp_msgadmin 'register', 'provider'` 注册 `provider_name`。

示例 9 (JMS) 指定缺省登录名。

```
sp_msgadmin 'default', 'login', 'one_jms_provider', 'loginsa',  
            'abcdef123456'
```

示例 10 (JMS) 列出登录名为“`loginsa`”的用户的详细信息:

```
sp_msgadmin 'list', 'login', 'my_jms_provider', 'loginsa'
```

示例 11 (JMS) 使用消息传送提供程序登录名“`jms_user1`”和消息传送提供程序名称“`my_jms_provider`”注册登录名“`ase_login1`”:

```
sp_msgadmin 'register', 'login', 'my_jms_provider', 'ase_login1',  
            'jms_user1', 'jms_user1_password'
```

示例 12 (JMS) 使用消息传送提供程序登录名“`jms_user1`”以及用于所有具有 `sa_role` 权限的 Adaptive Server 登录名的指定口令注册登录名:

```
sp_msgadmin 'register', 'login', 'my_jms_provider', null, 'jms_user1',  
            'jms_user1_password', 'sa_role'
```

示例 13 (JMS) 注册类为 TIBCO_JMS 且将 IP 10.23.233.32:4823 作为自己地址的 “my_jms_provider” 消息传送提供程序:

```
sp_msgadmin 'register', 'provider', 'my_jms_provider', 'TIBCO_JMS',
'tcp://10.23.233.32:4823'
```

示例 14 (JMS) 注册名为 “durable_sub1” 的持久性预订，然后使用 sp_msgadmin 'list' 显示有关新预订的信息。

```
sp_msgadmin 'register', 'subscription', 'durable_sub1',
'my_jms_provider?topic=topic.sample', null, null, 'durable1', 'client1'
sp_msgadmin 'list', 'subscription', 'durable_sub1'
```

示例 15 (JMS) 注册非持久性预订 “subscription_1”。

```
sp_msgadmin 'register', 'subscription', 'subscription_1',
'my_jms_provider?topic=topic.sample'
```

注释 必须先使用 sp_msgadmin register, provider 注册 “my_jms_provider”。

示例 16 (JMS) 删除缺省登录名:

```
sp_msgadmin 'remove', 'login', 'my_jms_provider'
```

示例 17 (JMS) 删除与消息传送提供程序 “my_jms_provider” 关联的 Adaptive Server 登录名 “ase_login1”:

```
sp_msgadmin 'remove', 'login', 'my_jms_provider', 'ase_login1'
```

示例 18 (JMS) 删除 “my_jms_provider” 上 sa_role 角色的所有登录名:

```
sp_msgadmin 'remove', 'login', 'my_jms_provider', null, 'sa_role'
```

示例 19 (MQ) 配置 Adaptive Server 的密钥存储库以便使用 SSL，其中密钥数据库文件的路径为 /var/mqm/clients/ssl/KeyringClient.kdb:

```
sp_msgadmin 'config', ibmq_keystore,
'var/mqm/clients/ssl/KeyringClient'
```

示例 20 (MQ) 注册类为 IBM_MQ 且 URL 为 chan11/TCP/host1(5678) 的 “mq_provider_1” 消息传送提供程序:

```
sp_msgadmin 'register', 'provider', 'mq_provider_1', 'ibm_mq',
'chan11/TCP/host1(5678)'
```

示例 21 (MQ) 显示监听端口为 3150 的计算机 “bigcrunch” 中的队列管理器名称:

```
sp_msgadmin 'show', 'QMGR', 'ibm_mq:/tcp/bigcrunch(3150)'
```

```
Name
-----
TEST
```

示例 22 (MQ) 显示队列管理器名称。该队列管理器位于监听端口为 3150 的计算机 “bigcrunch” 上。答复消息放置在 Q1 队列中，Adaptive Server 等待答复消息的最长时间为 20 毫秒：

```
sp_msgadmin 'show', 'QMGR', 'ibm_mq:channel1/tcp/bigcrunch(3150)',
  'timeout=20, replyqueue=Q1'
```

示例 23 (MQ) 显示队列管理器上的所有队列。答复消息放置在 Q1 队列中，Adaptive Server 等待答复消息的最长时间为 20 毫秒：

```
sp_msgadmin 'show', 'queues', 'ibm_mq:/tcp/bigcrunch(3150)',
  'timeout=20, replyqueue=Q1'
```

Name	Type
Q1	LOCAL
SYSTEM.MQSC.REPLY.QUEUE	MODEL
RQ1	REMOTE
AQ1	ALIAS
...	

示例 24 (MQ) 显示队列管理器上的所有通道：

```
sp_msgadmin 'show', 'channels', 'ibm_mq:/tcp/bigcrunch(3150)'
```

Name	Type
SNCH1	SENDER
SECH2	SERVER
RCCH3	RECEIVER
CHL5	SRVCONN
...	

示例 25 (SonicMQ) 向指定结束点注册名为 “sub1” 的预订，并将其放置在 Q1 队列中：

```
sp_msgadmin register, subscription, sub1,
  'sonicmq_jms:tcp://mysonic:7223??topic=T1,user=sonic_usr, password=sonic_pwd'
```

用法

不能在事务内部使用 sp_msgadmin。

sp_msgadmin 'register'

- 在使用登录名连接消息提供程序时，将按以下顺序解析登录名：
 - a 在结束点中指定的显式登录名和口令（如果提供）。
 - b 当前 Adaptive Server 登录名的显式登录名映射。
 - c 消息提供程序的缺省登录名和口令（角色与 Adaptive Server 登录名相对应）。
 - d 消息提供程序的缺省登录名和口令（无特定的角色关联）。

e 空登录名和口令（如果上述情况均不适用）。

- 在修改 Adaptive Server 登录名与消息传送提供程序登录名之间的登录名映射时，只能删除该映射，或者使用一组不同的映射重新注册。
- 仅限 MQ — 如果使用已注册的提供程序进入某一结束点，使用 `msgsubscribe`、`msgunsubscribe`、`msgpublish` 和 `msgconsume` 会返回错误。
- 有关 `sp_msgadmin` 不同变体的通用用法，请参见第 57 页的“`sp_msgadmin`”。

`sp_msgadmin 'remove'`

- 删除消息传送提供程序不会影响正传递给该消息提供程序的消息（即正在发送或接收的消息）。
- `sp_msgadmin 'remove'` 不影响任何当前指向该提供程序的连接。这意味着，在当前有连接指向指定消息提供程序的情况下，删除该消息提供程序、登录名或缺省值不会对连接产生影响。但是，Sybase 建议您不要这样做。
- 如果指定 `role_name`，则必须将 `local_login` 指定为 `null`。

`sp_msgadmin 'config'`

- `sp_msgadmin 'config'` 仅适用于 JMS。
- 所有 `sp_msgadmin 'config'` 参数都存储在 `sysattributes` 表中。若要检索这些值，请执行：

```
1> select * from sysattributes where class = 21
```

有关 `sysattributes` 的信息，请参见《参考手册：表》。

- 适用于 `sp_msgadmin 'config'` 的所有参数均会得到动态配置（'jvm' 除外）。

权限

您必须具有 `messaging_role` 才能运行 `msgsend` 和 `msgrecv` 函数。

您必须具有 `messaging_role` 和 `sso_role` 权限才能发出以下命令：

- `sp_msgadmin 'default'`
- `sp_msgadmim 'register'`
- `sp_msgadmin 'remove'`

任何用户均可发出以下命令：

- `sp_msgadmim 'help'`
- `sp_msgadmin 'list'`

msgconsume

说明 仅限 EAServer JMS — 提供 SQL 接口以使用发布到不同主题的消息。

语法

```
msgconsume_call ::=
    msgconsume (subscription_name, option_and_returns)
        subscription_name := basic_character_expression
        option_and_returns ::= [option_clause] [returns_clause]
        option_clause ::= [,] option option_string
        returns_clause ::= [,] returns sql_type
        subscriber_name ::= basic_character_expression
        sql_type ::=
            varchar(integer) | java.lang.String | text
            | varbinary(integer) | image
```

参数

basic_character_expression
数据类型为 char、varchar 或 java.lang.String 的 Transact-SQL 查询表达式。

option_string
option_string 的一般格式，详见第 128 页的“option_string”。表 3-6 中介绍了使用消息时要使用特殊选项：

表 3-6: msgconsume 的 option 和 option_string 值

option 值	option_string 值	缺省值	说明
timeout	介于 -1, 0 – (2 ³¹ – 1) 之间的 timespec 值	-1	缺省情况下，msgconsume 会阻塞消息，直至从消息总线中读取到下一条消息为止。如果 timeout 不是 -1，则当经过 timeout 时间间隔而仍未读取到消息时，msgconsume 将返回空值。值以毫秒为单位。 timeout 使用 timespec 选项。请参见第 130 页的 timespec。
requeue	string	无	要在其上对 Adaptive Server 无法处理的消息重新排队的目标、队列或主题的名称。如果未指定 requeue，则当消息无法处理时，将会显示一条错误消息。指定的结束点必须与 msgconsume 和 msgrecv 位于同一消息传送提供程序上。

subscription_name
要使用其消息的预订的名称。

returns
指定要返回的子句。

SQL_type

SQL 语句中使用的数据类型。

如果未指定要返回的数据类型，则缺省值为 `varchar(16384)`。合法的 SQL 数据类型包括：

- `varchar(n)`
- `text`
- `java.lang.String`
- `varbinary(n)`
- `image`
- `univarchar(n)`

示例

示例 1 在使用消息前在客户端服务器上定义预订：

```
sp_msgadmin 'register', 'subscription', 'subscription_1',
  'my_jms_provider?topic=topic.sample,user=user1,password=pwd',
  'Supplier=12345',null,'durable1', 'client1'
```

Sybase 建议，在使用预订中的消息前，应先进行预订：

```
select msgsubscribe('subscription_1')
declare @mymsg varchar(16384)
select @mymsg = msgconsume('subscription_1')
```

示例 2 声明变量并从指定的预订接收消息：

```
declare @mymsg varchar(16384)
select @mymsg = msgconsume('subscription_1',
  option 'timeout=0')
```

转发消息：

```
select msgsend
  (msgconsume('subscription_1'), 'my_jms_provider?queue=queue.sample')
```

读取消息并将其作为 `varbinary` 返回：

```
select msgconsume('subscription_1' returns varbinary(500))
```

用法

- 无法识别的 `option` 名称将导致错误。
- `msgconsume` 从 `end_point` 和 `message_filter`（由 `subscription_name` 指定）定义的主题读取消息。如果出现超时或错误，则返回空值；否则返回读取的消息正文。

- Adaptive Server 仅处理 `message`、`text` 或 `bytes` 类型的消息。如果 Adaptive Server 遇到无法处理的消息，并且未指定 `requeue`，则消息将留在初始队列中。后续读取操作在遇到该消息时，会采取相同的处理方式。若要防止出现此行为，请指定 `requeue`。指定 `requeue` 后，Adaptive Server 无法处理的消息将放入指定的队列。

指定的结束点必须与 `msgconsume` 中使用的结束点位于同一消息传送服务提供程序上。

- 如果该消息传送提供程序发出的消息不是 `message`、`text` 或 `bytes` 类型，并且未指定 `requeue`，Adaptive Server 将发出一条错误消息。
- 如果未进行预订，Adaptive Server 会在运行 `msgconsume` 期间自动预订。
- 调用 `msgconsume` 会产生三种结果：
 - 返回值是消息提供程序返回的 `message_body` 值（已转换为指定的 `returns` 类型）。
 - `@@msgheader` 和 `@@msgproperties` 的值设置为 `<msgheader>` 和 `<msgproperties>` 文档，其中包含由 `msgconsume` 返回的消息的属性。

有关 `<msgheader>` 和 `<msgproperties>` 的详细信息，请参见第 38 页的“与消息相关的全局变量”。
 - 可使用 `msgpropvalue` 从 XML 文档 `<msgheader>` 和 `<msgproperties>` 以及其它相关函数中提取特定属性的值。请参见第 76 页的 `msgpropvalue`。

权限

您必须具有 `messaging_role` 才能运行 `msgconsume`。

msgpropcount

说明 从 `<msgheader>` 和 `<msgproperties>` 文档中提取并返回 `msg_doc` 中属性或特性的数量。

语法

```
msgpropcount_call ::= msgpropcount([msg_doc]
    msg_doc ::= basic_character_expression
    prop_name ::= basic_character_expression
```

参数 `msgpropcount_call`
发出要使用 `msgpropcount` 函数的请求。

msg_doc
`basic_character_expression` 格式的 `<msgheader>` 或 `<msgproperties>` XML 文档。如果未指定 `msg_doc`，`msgpropcount` 将使用 `@@msgproperties` 的当前值。

prop_name
要以 `basic_character_expression` 格式从中提取值或类型的属性名。

示例 以下示例假定 `msgrecv` 中的某一调用返回一条具有单个属性 `trade_name` 且属性值为 “Acme Maintenance (“Quick & Safe”)” 的消息。这样，`@@msgproperties` 全局变量的值为：

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
  <msgproperties
    trade_name='Acme Maintenance (&quot;Quick &amp; Safe&quot;)'>
  </msgproperties>
```

根据 XML 约定的要求，与符号和短语 `Quick & Safe` 两侧的引号应替换为 XML 实体 `"` 和 `&`。

从上次检索到的消息中检索属性数量：

```
select msgpropcount (@@msgproperties)
```

msgproplist

说明 从 `<msgheader>` 和 `<msgproperties>` 文档中提取并返回一个 `option_string` 格式的字符串以及 `msg_doc` 的所有属性特性。

语法 `msgproplist_call ::= msgproplist([msg_doc] [returns varchar | text])`
`msg_doc ::= basic_character_expression`
`prop_name ::= basic_character_expression`

参数

`msgproplist_call`
发出要使用 `msgproplist` 函数的请求。

`msg_doc`
`<msgheader>` 或 `<msgproperties>` XML 文档。一个 `basic_character_expression`。如果未指定 `msg_doc`，则使用 `@@msgprproperties` 的当前值。

`prop_name`
要从中提取值或类型的属性名。一个 `basic_character_expression`。

`returns varchar | text`
指定返回消息的格式。

示例 以下示例假定 `msgrecv` 中的某一调用返回一条具有单个属性 “`trade_name`” 且属性值为 “`Acme Maintenance (“Quick & Safe”)`” 的消息。这样，`@@msgproperties` 全局变量的值为：

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
  <msgproperties
    trade_name='Acme Maintenance (&quot;Quick &amp; Safe&quot;)'>
  </msgproperties>
```

根据 XML 约定的要求，与符号和短语 `Quick & Safe` 两侧的引号应替换为 XML 实体 `"` 和 `&`。

以下两个命令均可检索属于某一消息的属性列表：

```
select msgproplist
select msgproplist(@@msgproperties)
```

用法

- 如果 `msgproplist` 调用的结果超过 16 K，则结果值将包含 “`TRUNCATED`” 一词。出现此情况时，指定 `returns text` 可防止结果遭到截断。必须使用其它 `msgprop` 函数来迭代访问该属性列表并获取属性的名称和值。
- 如果在未指定返回长度的情况下运行 `msgproplist`，所有超过缺省返回值 (32) 的输出都将被截断。若要避免这种情况，请指定返回内容的长度。例如，以下语句将被截断：

```
declare @properties varchar(1000)
select @properties = msgproplist(@@msgproperties returns varchar)
```

但以下语句将不会截断：

```
declare @properties varchar (1000)
select @properties= msgproplist(@@msgproperties returns varchar(1000))
```

msgpropname

说明 从 `<msgheader>` 和 `<msgproperties>` 文档中提取并返回属性名。如果 `integer` 参数的值小于 1 或大于 `msg_doc` 中的属性数量，则结果将为空值。

语法

```
msgpropname_call ::= msgpropname(integer[,msg_doc],)
msg_doc ::= basic_character_expression
prop_name ::= basic_character_expression
```

参数 *integer*

值的索引。

msgpropname_call

发出要使用 `msgpropname` 函数的请求。

msg_doc

`<msgheader>` 或 `<msgproperties>` XML 文档。一个 *basic_character_expression*。如果未指定 *msg_doc*，则使用 `@@msgprproperties` 的当前值。

prop_name

要从中提取值或类型的属性名。一个 *basic_character_expression*。

示例 **示例 1** 假定 `msgrecv` 中的某一调用返回一条具有单个属性 `trade_name` 且属性值为 “Acme Maintenance (“Quick & Safe”)” 的消息。这样，`@@msgproperties` 全局变量的值为：

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<msgproperties
  trade_name='Acme Maintenance (&quot;Quick &amp; Safe&quot;)'>
</msgproperties>
```

根据 XML 约定的要求，与符号和短语 `Quick & Safe` 两侧的引号应替换为 XML 实体 `"` 和 `&`。

示例 2 由于第 9 个属性不存在，因此返回一个空值：

```
select msgpropname(9, @@msgproperties)
```

msgproptype

说明	从 <code><msgheader></code> 和 <code><msgproperties></code> 文档中提取并返回名为 <code>prop_name</code> 的 <code>msg_doc</code> 属性的消息提供程序属性类型。如果 <code>msg_doc</code> 没有名为 <code>prop_name</code> 的属性，则结果为空值。
语法	<pre>msgproptype_call ::= msgproptype(prop_name [, msg_doc] msg_doc ::= basic_character_expression prop_name ::= basic_character_expression</pre>
参数	<p><code>msgproptype_call</code> 发出要使用 <code>msgproptype</code> 函数的请求。</p> <p><code>msg_doc</code> <code><msgheader></code> 或 <code><msgproperties></code> XML 文档。一个 <code>basic_character_expression</code>。如果未指定 <code>msg_doc</code>，则使用 <code>@@msgprproperties</code> 的当前值。</p> <p><code>prop_name</code> 要从中提取值或类型的属性名。一个 <code>basic_character_expression</code>。</p>
示例	先发送一条消息，它具有两个属性 “integer_prop” 和 “string_prop”，前者是一个值为 1234 的整数，后者是一个值为 “cat” 的字符串：

```
select msgsend('msgproptype example',
              'tibco_jms:tcp://localhost:7222?queue=queue.sample'
              MESSAGE_PROPERTY "integer_prop=1234,string_prop='cat'")
go
```

```
-----
ID:E4JMS-SERVER.82CC311EC:1
(1 row affected)
```

随后读回该消息：

```
select msgrecv('tibco_jms:tcp://localhost:7222?queue=queue.sample')
go
```

```
-----
msgproptype example
(1 row affected)
```

选择 `@@msgproperties` 全局变量以显示刚才接收的消息中有哪些属性：

```
select @@msgproperties
go
```

```
-----
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
  <msgproperties
    string_prop="&apos;cat&apos;"
```

```

ASE_RTMS_CHARSET="1"
ASE_ORIGIN="'&apos;francis_pinot_2&apos;";
ASE_SPID="15"
ASE_MSGBODY_FORMAT="'&apos;string&apos;";
ASE_TIMESTAMP="'&apos;2005/06/22 15:01:36.91&apos;";
ASE_MSGBODY_SCHEMA="'&apos;NULL&apos;";
ASE_RTMS_VERSION="'&apos;1.0&apos;";
ASE_VERSION="'&apos;12.5.0.0&apos;";
integer_prop="1234">
</msgproperties>

```

(1 row affected)

第一个 msgproptype 调用请求 “integer_prop” 属性的类型并返回 “Integer”：

```

1> select msgproptype('integer_prop')
2> go

```

```

-----
Integer
(1 row affected)

```

第二个 msgproptype 调用请求 “string_prop” 属性的类型并返回 “String”：

```

1> select msgproptype('string_prop')
2> go

```

```

-----
String
(1 row affected)

```

用法

- (MQ) 在使用 msgproptype 查询 MQ 消息标头中包含的下列二进制字段之一时，将返回字符串 “Hex”：
 - MsgId
 - CorrelId
 - GroupId
 - Encoding

例如，以下命令返回 “Hex”：

```
select msgproptype ('Encoding', @@msgheader)
```

msgpropvalue

说明	从 <code><msgheader></code> 和 <code><msgproperties></code> 文档中提取并返回名为 <code>prop_name</code> 的 <code>msg_doc</code> 属性的值。结果是转换为 <code>varchar</code> 的属性值；如果 <code>msg_doc</code> 没有名为 <code>prop_name</code> 的属性，则结果是空值。
语法	<pre>msgpropvalue_call ::= msgpropvalue(prop_name [, msg_doc]) msg_doc ::= basic_character_expression prop_name ::= basic_character_expression</pre>
参数	<p><code>msgpropvalue_call</code></p> <p>发出要使用 <code>msgpropvalue</code> 函数的请求。</p> <p><code>msg_doc</code></p> <p><code><msgheader></code> 或 <code><msgproperties></code> XML 文档。一个 <code>basic_character_expression</code>。如果未指定 <code>msg_doc</code>，则使用 <code>@@msgprproperties</code> 的当前值。</p> <p><code>prop_name</code></p> <p>要从中提取值或类型的属性名。一个 <code>basic_character_expression</code>。</p>
示例	<p>示例 1 下列示例假定 <code>msgrecv</code> 中的某一调用返回一条具有单个属性 “<code>trade_name</code>” 且属性值为 “Acme Maintenance (“Quick & Safe”)” 的消息。这样，<code>@@msgproperties</code> 全局变量的值为：</p>

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<msgproperties
  trade_name='Acme Maintenance (&quot;Quick &amp; Safe&quot;)'>
</msgproperties>
```

根据 XML 约定的要求，与符号和短语 `Quick & Safe` 两侧的引号应替换为 XML 实体 `"` 和 `&`。以下命令检索消息属性 `trade_name`：

```
select msgpropvalue (@@msgproperties, 'trade_name')
-----
('Quick & Safe') Acme Maintenance
```

这是存储在 `Transact-SQL` 变量或列中的原始字符串。

示例 2 由于检索的消息没有名为 “`discount`” 的属性，因此返回一个空值：

```
select msgpropvalue ('discount', @@msgproperties)
```

示例 3 检索第 8 个属性的值：

```
select msgpropvalue (msgpropname(8, @@msgproperties))
```


msgpublish

说明 仅限 JMS — 提供 SQL 接口以向主题发布消息。

语法

```

message_publish_call ::=
    msgpublish(message_body, subscription_name
               [options_and_properties])
    options_and_properties ::=
        [option_clause] [properties_clause]
        [header_clause]
    option_clause ::= [,] option option_string
    header_clause ::= [,] message header
        option_string
    properties_clause ::=
        [,] message property option_string
    message_body ::= scalar_expression |
        (select_for_xml)
  
```

参数

message_body

要发送的消息。消息正文可包含任意字符串，它可以是二进制数据、字符数据或 SQLX 数据。

subscription_name

要想起发布消息的预订的名称。

option_clause

option 名称和 *option_string* 的一般格式，详见第 128 页上的相关小节。

第 78 页的表 3-7 提供了可为 msgsend 指定的选项。

properties_clause

option_string 或以下表中列出的选项之一。表 3-7 和第 79 页的表 3-8 中描述的选项将设置为消息标头或消息属性中的属性，如表中的放置列所述。选项值即属性值。

属性名区分大小写。

scalar_expression

如果消息是一个 SQL *scalar_expression*，则它可以是任意数据类型。

如果未指定 *type* 选项，则当 *scalar_expression* 的计算结果为 *character* 数据类型时，消息类型为 *text*；否则消息类型为 *bytes*。

如果 *scalar_expression* 的数据类型不是 *character*，则使用关于隐式转换的常规 SQL 规则将其转换为 *varbinary*。数据类型的 *binary* 值将按照主机的字节顺序包括在消息中。

select_for_xml

一个指定 *for xml* 子句的 *select* 表达式。

header_clause

允许用户仅指定标头属性。如果输入了无法识别的标头属性，则会显示错误。

如果在 *message property* 和 *message header* 子句中都指定了一个已经识别的标头属性，则优先使用 *message header* 子句中的标头属性。

如果在 *option_clause* 中指定了无法识别的选项，则会显示错误。

message header 子句接受以前识别的所有标头属性。

示例

若要发布消息，必须在客户端要连接的服务器上定义预订：

```
sp_msgadmin 'register', 'subscription', 'subscription_1',
  'my_jms_provider?topic=topic.sample,user=user1,password=pwd',
  'Supplier=12345',null, 'durable1', 'client'
```

这样，客户端 / 服务器即可向指定的预订发布消息：

```
select msgpublish
  ('Sending order', 'subscription_1',
  MESSAGE PROPERTY 'Supplier=12345')
```

用法

- 如果使用 *message property*，则忽略无法识别的选项。如果在 *msgsend* 或 *msgpublish* 函数中使用 *message header*，则当指定了无法识别的选项时，将会显示错误。
- 在对以下命令的调用中必须指定 *subscription_name*:


```
sp_msgadmin 'register', 'subscription'
```

 在对以下命令的后续调用中不要指定 *subscription_name*:


```
sp_msgadmin 'remove', 'subscription'
```
- 表 3-7 列出了可为 *msgpublish* 指定的选项（适用于 JMS）。

表 3-7: msgpublish option_string 参数的值

选项	值	缺省值	注释
schema	<ul style="list-style-type: none"> no yes "user_schema" 	no	输入下列值之一： <ul style="list-style-type: none"> <i>user_schema</i> — 用户提供的 schema，用于描述 <i>message_body</i>。 no — 指示不生成 schema，也不将其作为消息的一部分发送。 yes — 指示 Adaptive Server 将为消息生成 XML schema。yes 仅在使用 <i>select_for_xml</i> 参数的 <i>message_body</i> 中有意义。<i>select_for_xml</i> 会生成以 SQLX 格式表示的 SQL 结果集。生成的 XML schema 是一个 SQLX 格式的 schema，用于描述结果集文档。 该 schema 将作为 ASE_MSGBODY_SCHEMA 属性包括在消息中。
type	text 或 bytes	text	要发送的消息类型。

- 表 3-8 列出了 *properties_clause* 参数的选项和值。如果使用未列在表 3-8 中的属性，则将其设置为已发送消息的消息属性中的属性。

表 3-8: *msgpublish properties_clause* 参数的值

选项	值	缺省值	放置	注释
correlation	<i>string</i>	无	标头	客户端应用程序设置相关 ID 以将各个消息关联在一起。Adaptive Server 设置由应用程序指定的相关 ID。
mode	<ul style="list-style-type: none"> persistent non-persistent 	persistent	标头	如果输入： <ul style="list-style-type: none"> persistent — JMS 提供程序使用稳定存储支持消息。将 mode 设置为 persistent 时，如果消息传送提供程序在消息受到使用前失败，则消息很可能会保存下来。 non-persistent — 当消息传送提供程序失败时，消息可能会在到达所需目标前丢失。
priority	1 到 9	4	标头	priority 的行为由基础消息总线控制。此处提到的值适用于 JMS。 1–4 的 priority 为正常区间；5–9 的 priority 为加速区间。
replyqueue	一个包含 <i>queue_name</i> 的字符串	无	标头	如果 <i>queue_name</i> 或 <i>topic_name</i> 的值为： <ul style="list-style-type: none"> syb_temp — Adaptive Server 会创建一个临时目标，并将新建临时目标的相关信息作为标头信息的一部分发送。Adaptive Server 随后会将 @@msgreplytoinfo 更新为临时目标。临时目标的类型（队列或主题）取决于指定的是 replyqueue 还是 replytopic。仅使用最后列出的选项。 已存在的目标 — Adaptive Server 不创建新目标，而是使用用户指定的目标。
replytopic	一个包含 <i>topic_name</i> 的字符串	无	标头	
ttd	$0 - (2^{63}-1)$	0	标头	ttd 指在消息总线中的生存时间。Adaptive Server 不受此属性影响。 到期信息是消息保持有效的持续时间，单位为毫秒。例如，60 指示消息的生存期为 60 毫秒。 值 0 指示消息永不到期。 ttd 使用 timespec 选项。请参见第 130 页的 timespec 。

权限

您必须具有 **messaging_role** 才能运行 **msgpublish**。

msgrecv

说明

提供 SQL 接口以从不同的服务结束点（必须是队列）接收消息。

`msgrecv` 从指定的 `service_provider` 和 `service_destination` 接收消息并返回该消息。返回值为服务提供程序返回的消息正文（已转换为指定的返回类型）。

语法

```
msgrecv_call ::=
    msgrecv (end_point options_filter_and_returns)
            options_filters_and_return ::=
            [option_clause] [filter_clause] [returns_clause]
            option_clause ::= [,] option_option_string
            filter_clause ::= [,] message_selector message_filter
            message_filter ::= basic_character_expression
            returns_clause ::= [,] returns_sql_type
            end_point ::= basic_character_expression
            sql_type ::=
                varchar(integer) | java.lang.String | text
                | varbinary(integer) | image
            message_filter ::= basic_character_expression
```

参数

basic_character_expression

数据类型为 `char`、`varchar` 或 `java.lang.String` 的 SQL 查询表达式。

end_point

一个 *basic_character_expression*，其中运行期值为 `service_provider_uri`。
end_point 为消息的目标。

filter_clause

将 *message_filter* 直接传递给指定的消息提供程序，并由后者决定它的使用方式。

message_filter

一个过滤器参数和 *basic_character_expression*。过滤器值将直接传递给消息提供程序。其使用方式取决于消息提供程序。有关消息过滤器的讨论，请参见下面的“用法”一节。

如果提供程序类为“`ibm_mq`”，则忽略为 `msgrecv` 指定的所有 *message_filter*。

`msgrecv`

从指定的 `service_provider` 和 `service_destination` 接收消息并返回该消息。返回值为服务提供程序返回的消息正文（已转换为指定的返回类型）。

option

第 84 页的表 3-9（针对 MQ）和第 89 页的表 3-10（针对 JMS）中显示的值。

注释 无法识别的 *option* 名称将导致错误。

option_string

option_string 的一般格式，详见第 128 页。第 84 页的表 3-9（针对 MQ）和第 89 页的表 3-10（针对 JMS）描述了 *msgrecv* 的选项。

returns_clause

要返回的数据类型。如果未指定 *returns_clause*，则缺省值为 `varchar(16384)`。

如果指定 `varbinary` 或 `image` 类型的 *returns_clause*，则以消息的字节顺序返回数据。

sql_type

下列有效的 SQL 数据类型之一：

- `varchar(n)`
- `text`
- `java.lang.String`
- `varbinary(n)`
- `image`
- `univarchar(n)`

示例

示例 1 (MQ) 按指定的超时设置从队列 Q1 中读取消息。如果 Q1 在超时时间（3 秒）内没有可用消息，则返回空值：

```
select msgrecv(
    'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
    option 'timeout=3ss')
```

示例 2 (MQ) 指定一个 `correlationId`，但不指定超时设置。当队列中出现与 `correlationId` 匹配的消息时，调用将返回相应信息：

```
select msgrecv(
    'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
    option 'correlationId=x67a12z99')
```

示例 3 (MQ) 指定一个 `groupId` 和 `allMsgsInGroup`，但不指定超时设置。此调用将阻塞，直至队列中出现具有指定 `groupId` 的所有消息：

```
select msgrecv(  
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
  option 'groupId=g7853b77,allMsgsInGroup=yes')
```

示例 4 (MQ) 队列中已存在下列消息：

AA BB CC DD EE FF GG HH

首先在浏览器模式下读取前三条消息 (AA – CC) 并删除 CC。接着将浏览游标设置为返回开头，随后在浏览器模式下读取前三条消息 (AA – DD) 并删除 DD。最后，将位置设置为 `next` 并执行读取操作，这将读取并删除 AA。当此示例完成时，队列中将不再有消息 AA、CC 和 DD。

```
-- Browse cursor at the beginning, this will return 'AA'  
select msgrecv(  
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
  option 'inputMode=browse+Qdefault,browse=first')  
  
-- Browse the next message, this will return 'BB'  
select msgrecv(  
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
  option 'inputMode=browse+Qdefault,browse=next')  
  
-- Browse the next message, this will return 'CC'  
select msgrecv(  
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
  option 'inputMode=browse+Qdefault,browse=next')  
  
-- Remove the message under the browse cursor, this will return 'CC'  
select msgrecv(  
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
  option 'inputMode=browse+Qdefault,position=cursor')  
  
-- Reposition browse cursor at the beginning, this will return 'AA'  
select msgrecv(  
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
  option 'inputMode=browse+Qdefault,browse=first')  
  
-- Browse the next message, this will return 'BB'  
select msgrecv(  
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',  
  option 'inputMode=browse+Qdefault,browse=next')  
  
-- Browse the next message, this will return 'DD'
```

```

select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,browse=next')

-- Read the message under the cursor, this will return 'DD'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,position=cursor')

-- Read the next message in queue order, this will return 'AA'
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q1',
  option 'inputMode=browse+Qdefault,position=next')

```

示例 5 Tibco JMS — 从指定的 *end_point* 接收消息:

```

select msgrecv
  ('tibco_jms:tcp://my_jms_host:7222?queue=queue.sample,'
  +'user=jms_user1,password=jms_user1_password')

```

示例 6 SonicMQ JMS — 使用 *timeout* 选项从指定的 *end_point* 接收队列 Q1 中的消息:

```

select msgrecv
  ('sonicmq_jms:tcp://mysonic:7223?queue=Q1,user=sonic_usr,
  password=sonic_pwd',option 'timeout=1000')

```

示例 7 (JMS) 在使用 *timeout* 选项并指定消息选择器的情况下从指定的 *end_point* 接收消息:

```

declare @mymsg varchar (16384)
select @mymsg = msgrecv('my_jms_provider?queue=queue.sample',
  option 'timeout=1000'
  message selector 'correlationID = 'MSG_001''')

```

示例 8 (JMS) 仅在消息属性 “Name” 为 “John Smith” 时, 此 *msgrecv* 调用才使用 *queue.sample* 中的消息:

```

select msgrecv('my_jms_provider?queue=queue.sample',
  message selector 'Name='John Smith''')

```

示例 9 (JMS) 演示如何在表中插入一条 *text* 消息:

```

create table T1(c1 numeric(5,0)identity, m text)
insert into T1
select msgrecv('my_jms_provider?queue=queue.sample',
  returns text)

```

示例 10 (JMS) 以下示例读取一条消息并将其作为 `varbinary` 返回：

```
select msgrecv('my_jms_provider?queue=queue.sample'
              returns varbinary(500))
```

用法

(MQ) 表 3-9 列出了 `msgrecv` 属性可用的 `option` 和 `option_string` 值。

表 3-9. `msgrecv` 的 MQ `option` 和 `option_string` 值

option 值	option_string 值	缺省值	说明
<code>allMsgsInGroup</code>	<ul style="list-style-type: none"> • <code>yes</code> • <code>no</code> 	<code>no</code>	如果未指定 <code>groupId</code> ，则忽略此选项。 如果指定： <ul style="list-style-type: none"> • <code>yes</code> — 一个组的所有逻辑消息必须都出现在队列中，才会返回该组的第一条消息。 • <code>no</code> — 一个组的所有逻辑消息不必都出现在队列中，即可返回该组的第一条消息。
<code>allSegments</code>	<ul style="list-style-type: none"> • <code>yes</code> • <code>no</code> 	<code>no</code>	如果指定： <ul style="list-style-type: none"> • <code>yes</code> — 一个分段消息的所有消息必须都出现在队列中，才会返回第一个消息段。 • <code>no</code> — 一个分段消息的所有消息不必都出现在队列中，即可返回第一个消息段。
<code>browse</code>	<ul style="list-style-type: none"> • <code>next</code> • <code>next+Lock</code> • <code>first</code> • <code>first+Lock</code> • <code>cursor</code> • <code>cursor+Lock</code> • <code>reopen</code> • <code>reopen+Lock</code> • <code>unlock</code> • <code>null</code> 	<code>null</code>	如果将浏览器属性设置为： <ul style="list-style-type: none"> • <code>null</code> — 从队列中读取并删除消息。 <code>position</code> 选项控制要读取的消息。 • <code>null</code> 之外的任何其它值 — 从队列中读取消息，但不删除消息。读取顺序取决于队列的缺省顺序（先进先出、后进先出或按照优先级）。 如果同时： <ul style="list-style-type: none"> • 指定 <code>msgId</code>、<code>correlationId</code>、<code>groupId</code>、<code>sequenceId</code> 或 <code>offset</code> — MQ 将浏览或读取与指定选择标准匹配的下一条消息。 • 指定 <code>timeout</code>，并且未找到与选择标准匹配的消息 — 将返回一个空值。 • 未指定 <code>timeout</code> — <code>msgrecv</code> 操作将阻塞，直至队列中出现与选择标准匹配的消息。

option 值	option_string 值	缺省值	说明
browse (续)			<p>如果为 browse 指定以下值:</p> <ul style="list-style-type: none"> • next — 返回下一条消息。 • next+Lock — 返回并锁定消息, 以便其它阅读器无法将其删除。 • first — 返回第一条消息。如果在发出一个或多个 browse=next 选项后指定 browse=first, 浏览游标将重新定位到打开队列的起始位置。 • first+Lock — 返回并锁定第一条消息, 以便其它阅读无法将其删除。 • cursor — 返回浏览游标所在的消息。请勿在未执行 browse=first, browse=first+Lock、browse=next 或 browse=next+Lock 的情况下使用 browse=cursor。重复执行 browse=cursor 会返回同一条消息。 • cursor+Lock — 返回并锁定游标所在的消息, 以便其它阅读器无法将其删除。 • reopen — 关闭浏览游标, 随后将其重新打开并放置在开头。对于优先级队列, 如果自上次打开游标后有较高优先级的消息传入, 该消息将出现在队列的开头。 • reopen+Lock — 关闭浏览游标, 随后将其重新打开并放置在开头, 同时锁定第一条消息, 以便其它阅读器无法将其删除。 • unlock — 解锁并返回游标所在的消息。

option 值	option_string 值	缺省值	说明
bufferLength	sizespec 0 或 1 – 值		<p>使用 bufferLength 大小的缓冲区读取消息。</p> <ul style="list-style-type: none"> 消息传送内置函数会尝试分配具有此长度的缓冲区。如果没有足够内存来分配该缓冲区，此命令将失败。 将 msgrecv 指定为返回 text 或 image 时，msgrecv 假定消息大小为指定队列能够容纳的最大消息，并使用 maxMsgLength 队列属性。将 maxMsgLength 设置为以下值时，应增大消息传送内存： <ul style="list-style-type: none"> 缺省值 4 MB，或者 远大于消息实际长度的值。 <p>Sybase 建议将 maxMsgLength 队列属性设置为应用程序所允许的最小值，以便 Adaptive Server 使用最少的内存来读取消息。若要设置 maxMsgLength，请使用 MQ 命令 (MQSC) 工具更改队列的 MAXMSGL 属性。</p> <p>缺省值 bufferLength 缺省值为以下值之一：</p> <ul style="list-style-type: none"> 为队列管理器和目标队列定义的 maxMsgLength 的最小值，或者 返回类型的长度（如果返回类型不是 text、image 或 java.lang.String）。 <p>0 指示使用缺省值。</p> <p>对于发布 / 预订消息，bufferLength 必须包括消息主题的长度（包括 MQRFB 标头）</p>
closeAfterRecv	<ul style="list-style-type: none"> yes no 	no	<p>如果：</p> <ul style="list-style-type: none"> yes — 队列在当前 msgrecv 操作完成后关闭，以便在后续 msgrecv 调用中以其它输入模式重新打开。 no — 队列在当前 msgrecv 操作完成后继续保持打开状态。
completeMsg	<ul style="list-style-type: none"> yes no 	yes	<p>如果：</p> <ul style="list-style-type: none"> yes — 将分段消息作为一条消息返回。 no — 如果存在分段消息，则将每个段作为一条单独的消息返回。 <p>对于同一结束点，completeMsg 应与所有 msgrecv 调用具有相同的设置。</p>
correlationId	<ul style="list-style-type: none"> null string 	null	<p>要读取的相关 ID，在 select 语句中用于选择队列中的特定消息。</p> <p>MQ 将此字段定义为支持 binary 值的“unsigned char”。若要输入一个二进制字符串作为 correlationId，请使用“0x...”作为值。请勿在值两侧添加引号。</p>

<i>option</i> 值	<i>option_string</i> 值	缺省值	说明
<i>formatName</i>	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>预期消息格式的名称。如果指定此选项，而消息的名称字段 <i>formatName</i> 与之不匹配，则不读取消息。有关详细信息，请参见此表中的 <i>requeue</i> 选项。</p> <p>MQ 将此字符串的长度限制为 8 字节。</p>
<i>groupid</i>	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>要读取的消息的组 ID。这是一个 <i>select</i> 选项。MQ 将此字段定义为 “<i>unsigned char</i>”，这意味着它支持 <i>binary</i> 值。若要输入一个二进制字符串作为 <i>msgId</i>，请使用 “0x...” 作为值。请勿在值两侧添加引号。</p>
<i>inputMode</i>	<ul style="list-style-type: none"> • <i>browse</i> • <i>Qdefault</i> • <i>shared</i> • <i>exclusive</i> • <i>browse+Qdefault</i> • <i>browse+shared</i> • <i>browse+exclusive</i> 	<i>Qdefault</i>	<p><i>inputMode</i> 的值按以下方式打开 MQ 队列：</p> <ul style="list-style-type: none"> • <i>browse</i> — 打开队列仅供浏览。如果尝试进行破坏性读取，队列管理器将生成错误。 • <i>Qdefault</i> — 使用为队列定义的缺省输入模式打开队列。 • <i>shared</i> — 在共享输入模式下打开队列。如果队列已由另一个 MQ 句柄在独占模式下打开，则会收到错误。 • <i>exclusive</i> — 在独占输入模式下打开队列。如果队列已由另一个 MQ 句柄在共享或独占模式下打开，则会收到错误。 • <i>browse+Qdefault</i> — 在浏览和共享输入模式下打开队列。 • <i>browse+shared</i> — 在浏览和共享输入模式下打开队列。如果队列已由另一个 MQ 句柄在独占模式下打开，则会收到错误。 • <i>browse+exclusive</i> — 在浏览和独占输入模式下打开队列。如果队列已由另一个 MQ 句柄在共享或独占模式下打开，则会收到错误。 <p><i>inputMode</i> 仅适用于 <i>msgrecv</i>。</p> <p>对于每个结束点，均须在以下时间指定 <i>inputMode</i>：</p> <ul style="list-style-type: none"> • 在第一次 <i>msgrecv</i> 操作时指定，或者 • 在指定 <i>closeAfterRecv</i> 后指定。 <p>尝试跨调用更改 <i>inputMode</i> 的值可能会导致意外结果。</p>
<i>msgId</i>	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>要读取的消息的消息 ID。</p> <p>作为一个选择性选项，可使用 <i>msgId</i> 选择队列中的特定消息。</p> <p>MQ 将此字段定义为支持 <i>binary</i> 值的 “<i>BYTE</i> 数组”。若要输入一个二进制字符串作为 <i>msgId</i>，请使用 “0x...” 作为值。请勿在值两侧添加引号，否则会将其解释为带引号的字符串。</p>
<i>offset</i>	-1 以及介于 0 – <i>maxint</i> 之间的 <i>integer</i>		<p>要读取的消息的偏移量。</p> <p>如果为 -1，则不指定偏移量。</p> <p>作为 <i>select</i> 选项，可使用 <i>offset</i> 选择队列中的特定消息。</p>

option 值	option_string 值	缺省值	说明
ordering	<ul style="list-style-type: none"> logical physical 	physical	<p>如果 ordering 为：</p> <ul style="list-style-type: none"> logical — 按照 groupId、sequenceId 和 offsets 以逻辑顺序读取消息。 physical — 按照消息在队列中出现的顺序读取消息。
position	<ul style="list-style-type: none"> next cursor 	next	<p>position 控制要返回的消息。根据指定的 inputMode 值，有 1 到 2 个“读取”位置：</p> <ul style="list-style-type: none"> “Normal” — 通常发生破坏性读取的缺省读取位置。打开队列时，“normal”读取位置位于队列的第一条消息上。 “Browse cursor” — 由之前指定 browse 的调用定位的读取位置。在打开队列进行浏览时，“browse cursor”位于队列中的第一条消息之前。“Browse cursor”仅适用于 browse+Qdefault、browse+shared 和 browse+exclusive <p>如果：</p> <ul style="list-style-type: none"> next — 返回当前位于“normal”读取位置的消息。返回消息后，“normal”读取位置将移到前一条消息。 cursor — 返回当前位于“browse cursor”处的消息。如果尚未定位“browse cursor”，MQ 队列管理器将引发错误。返回消息后，“browse cursor”将移到前一条消息。 <p>MQ 队列管理器应用以下规则来确定要返回的消息：</p> <ul style="list-style-type: none"> 队列的缺省顺序（优先级、先进先出、后进先出） 指定的任何选择标准（messageId、correlationId、groupId、sequenceId 或 offset）
requeue	<ul style="list-style-type: none"> null string 	null	<p>它必须是结束点的完整 URI。</p> <p>在下列情况中，读取的消息将重新排入指定的队列：</p> <ul style="list-style-type: none"> msgrecv 在指定了 formatName 的情况下读取消息。 读取的消息具有不同的 formatName。 requeue 不是 null。 <p>如果无法将消息重新排入指定的队列，则将其留在读取该消息时所在队列，并引发例外。</p> <p>MQ 将重新排入队列的消息长度限制为 48 字节。</p>
sequenceId	介于 -1 – 9,999,999 之间的 integer	-1	<p>要读取的消息的序列 ID。</p> <p>如果为 -1，则不指定序列 ID。</p> <p>作为一个选择性选项，可使用 sequenceId 选择队列中的特定消息。</p>

<i>option</i> 值	<i>option_string</i> 值	缺省值	说明
truncationAllowed	<ul style="list-style-type: none"> yes no 	no	在下列情况中可截断消息： <ul style="list-style-type: none"> 用于读取消息的缓冲区（bufferLength 或返回的数据类型的长度）。 缓冲区小于消息的长度。 指定为： <ul style="list-style-type: none"> yes — 允许截断。 no — 不允许截断。当值为 no 而消息已被截断时，读取操作将失败。
timeout	-1 以及介于 0 – (2 ³² -1) 之间的 timespec	-1	指定超时设置。如果： <ul style="list-style-type: none"> -1 — 没有超时设置。 timeout 指定为整数，值的单位为毫秒。 有关详细信息，请参见第 130 页的 timespec。

(JMS) 表 3-10 列出了 msgrecv 属性可用的 *option* 和 *option_string* 值。

表 3-10: msgrecv 的 JMS option 和 option_string 值

<i>option</i> 值	<i>option_string</i> 值	缺省值	说明
requeue	string	无	要在其上对 Adaptive Server 无法处理的消息重新排队的目标、队列或主题的名称。如果未指定 requeue，则当消息无法处理时，将会显示一条错误消息。指定的结束点必须与 msgconsume 和 msgrecv 位于同一消息传送提供程序上。
timeout	timespec -1, 0 - (2 ³¹ - 1)	-1	缺省情况下，msgrecv 会阻塞消息，直至从消息总线中读取到下一条消息为止。如果 timeout 不是 -1，则当经过 timeout 时间间隔而仍未读取到消息时，msgrecv 将返回空值。值以毫秒为单位。请参见第 130 页的 timespec。

- 无法识别的 option 名称将导致错误。
- 有关从消息标头中读取的属性，请参见第 38 页的 @msgheader。
- msgrecv 从指定的 service_provider 和 service_definition 接收消息并返回该消息。
- 缺省情况下，msgrecv 是一个阻塞命令，它会阻塞消息，直至从消息总线中读取到下一条消息为止。如果 timeout 不是 -1，则当经过 timeout 时间间隔而仍未读取到消息时，msgrecv 将返回空值。其值以毫秒为单位。
- Adaptive Server 仅处理 message、text 或 bytes 类型的消息。如果 Adaptive Server 遇到无法处理的消息，并且未指定 requeue，则消息将留在初始队列中。后续读取操作在遇到该消息时，会采取相同的处理方式。若要防止出现此行为，请指定 requeue。指定 requeue 后，Adaptive Server 无法处理的消息将放入指定的队列。

指定的结束点必须与 `msgrecv` 中使用的结束点位于同一个消息传送服务提供程序上。

- 消息会按照主机的字节顺序包括数据类型的 `binary` 值。
- 调用 `msgrecv` 会产生三种结果：
 - 返回值是消息提供程序返回的 `message_body` 值（已转换为指定的 `returns` 类型）。
 - `@@msgheader` 和 `@@msgproperties` 的值设置为 `<msgheader>` 和 `<msgproperties>` 文档的值，其中包含由 `msgrecv` 返回的消息的属性。
 - 可使用 `msgpropvalue` 从 `<msgheader>` 和 `<msgproperties>` 文档中提取特定属性的值。请参见第 76 页的“`msgpropvalue`”。
 - 第 38 页的“与消息相关的全局变量”介绍了 `<msgheader>` 和 `<msgproperties>` 的一般格式。

MQ 和 msgrecv

下列语句仅在提供程序类为“`ibm_mq`”时有效：

- `msgid`、`correlationId`、`groupId`、`sequenceId` 和 `offset` 选项充当选择消息的匹配标准。如果指定，则返回与指定值匹配的下一条消息。此限定由 WebSphere MQ 队列管理器执行。
- 如果已接收消息的 `MQMD.Format` 字段为“`MQSTR`”，则假定数据为字符数据，并可将数据作为 `text` 或 `varchar` 返回。所有其它格式名称只能作为 `image` 或 `binary` 返回。一种特殊的情况是 `MQMD.Format` 为“`MQHRF`”。在此情况下，将改用 `MQRFH.Format` 字段。如果无法以指定的返回类型返回消息正文，则当指定了 `requeue` 选项时，会将消息发送给 `requeue` 选项；否则读取操作将失败。当 `MQMD.Format` 为“`MQSTR`”并且消息正文仅包含字符数据时，MQ 不会强制如此。程序员应始终指定 `image` 或 `varbinary` 返回类型。

引号属性或 option 值

- 在 `option` 值两侧添加单引号 (') 可将值作为字符串处理。如果省略单引号，则会将 `option` 值作为其它属性名处理，只有两个属性具有相同的值时，表达式才为 `true`。

如果应用程序使用带引号的标识符，则必须在消息选择器两侧添加单引号 (')。这意味着，如果选择器中有字符串值，则必须用双引号 (") 将这些值括起来。例如：

```
set quoted_identifier on
select msgrecv ('my_jms_provider?queue=queue.sample',
  message selector 'color = ''red''')
```

如果应用程序不使用带引号的标识符，则可在消息选择器两侧添加普通双引号。例如：

```
set quoted_identifier off
select msgrecv('my_jms_provider?queue=queue.sample',
  message selector "color='red'")
```

在下一示例中，**消息传送客户端**应用程序发送一条消息，指示“color”属性的值将为“red”并且“red”属性的值将为“color”。

```
select msgsend ('Sending message with property color',
  'my_jms_provider?queue=queue.sample'
  message selector 'color=red, red=color')
```

对于仅希望使用包含值为“red”的“color”属性的客户端应用程序，必须在选择器值两侧添加双引号(")。例如：

```
select msgrecv('my_jms_provider?queue=queue.sample'
  message selector 'color=''red''')
```

但是，如果客户端应用程序使用以下语法，则不会收到该消息，因为“red”将作为属性名处理：

```
select msgrecv('my_jms_provider?queue=queue.sample',
  message selector 'color=red')
```

在另一个示例中，客户端发送一条选择并过滤出多个属性的消息：

```
select msgsend('Sending message with properties',
  'my_jms_provider?queue=queue.sample',
  message selector 'color=red, shape=square')
```

如果另一个客户端希望选择属性“color”等于“red”且属性“shape”等于“square”的消息，该客户端必须执行以下语法：

```
select msgrecv('my_jms_provider?queue=queue.sample',
  message selector 'color=''red'' and shape=''square''')
```

消息过滤器

- 如果指定了过滤器参数，则会直接将过滤器值传递给消息提供程序。该值的具体使用方式取决于消息提供程序。
- 消息过滤器中指定的比较运算会使用消息提供程序指定的排序顺序，该顺序可能不同于 Adaptive Server 使用的排序顺序。
- JMS 消息提供程序使用 JMS 消息选择器作为过滤器。JMS 消息选择器的规则如下：
 - 消息选择器的语法是条件表达式的子集，包括 not、and 或 between 和 like。

- 标识符区分大小写。
- 标识符必须指定消息标头字段和属性名。
- 仅限 JMS — 如果 *message_filter* 指定为 *msgrecv*，则会将其忽略。
- 仅限 MQ — 通过在消息选项中指定相关和消息 ID，可选择特定的消息。

权限

您必须具有 *messaging_role* 才能运行 *msgrecv*。

msgsend

说明 提供 SQL 接口，以将消息发送给 `queue` 类型的其它服务结束点。

语法

```

message_send_call ::=
    msgsend(message_body, end_point [options_and_properties])
    options_and_properties ::= [option_clause]
    [properties_clause] [header_clause]
    option_clause ::= [,] option option_string
    properties_clause ::= [,] message property
    property_option_string
    header_clause ::= [,] message header
    header_option_string
    message_body ::= scalar_expression |
    (select_for_xml)
    end_point ::= basic_character_expression
  
```

参数 `message_body`

要发送的消息。消息正文可包含任意字符串，并可以是二进制、字符或 SQLX 数据。

`endpoint`

要将消息指向的队列。`endpoint` 是一个 `basic_character_expression`，其中运行期值为 `service_provider_uri`。

`option`

允许指定 `msgsend` 的选项。如果使用的是 JMS，请使用第 101 页的表 3-11 中的选项。如果使用的是 MQ，请使用第 101 页的表 3-12 中的选项。

`option_string`

指定 `option_string` 的一般语法和处理方式。各个选项在引用它们的函数中均有介绍。

```

option_string ::= basic_character_expression
option_string_value ::= option_and_value [,] option_and_value]
option_and_value ::= option_name = option_value
option_name ::= simple_identifier
option_value ::= simple_identifier
| quoted_string | integer_literal | float_literal | byte_literal
| true | false | null
  
```

参数	说明
<code>option_string</code>	描述要指定的选项的字符串
<code>simple_identifier</code>	标识 <code>option</code> 值的字符串
<code>quoted_string</code>	使用关于嵌入式引号的常规 SQL 约定形成的字符串
<code>integer_literal</code>	按常规 SQL 约定指定的文字

参数	说明
<i>float_literal</i>	按常规 SQL 约定指定的文字
true	布尔型文字
false	布尔型文字
null	空文字
<i>byte_literal</i>	格式为 0xHH，其中每个 H 都是一个十六进制数字

properties_clause

一个 *property_option_string*，或第 104 页的表 3-13（针对 MQ）和第 112 页的表 3-14（针对 JMS）中列出的选项之一。这两个表中描述的选项均设置为消息标头或消息属性中的属性，如表中的放置列所述。选项值即属性值。

属性名区分大小写。

仅限 Tibco JMS — 如果使用未列在第 112 页的表 3-14 中的属性，则将其设置为已发送消息的消息属性中的属性。

仅限 MQ — *properties_clause* 的值因指定的 rhfCommand 选项值而异：

- 只有 rhfCommand 为 deletePublication 时，第 113 页的表 3-15 中的属性才有效。

发送至发布流的 deletePublication 命令消息会指示 MQ 发布 / 预订代理删除其在发布流中为指定主题保留的所有发布的副本。

将忽略 msgsend 的 message_body 参数。

- 只有 rhfCommand 为 deregisterPublisher 时，第 113 页的表 3-16 中的属性才有效。
- 发送至 MQ 发布 / 预订代理控制队列的 deregisterPublisher 命令消息会通知该代理不再发布有关指定主题的信息。
- 只有 rhfCommand 为 deregisterSubscriber 时，第 114 页的表 3-17 中的属性才有效。

将忽略 msgsend 的 message_body 参数。

如果 msgType 为 request，则将答复消息发送给 replyToQmgr 和 replyToQueue。

- 只有 rhfCommand 为 publish 时，第 115 页的表 3-18 中的属性才有效。

publish 命令消息将发送给发布流队列，以发布有关特定主题的信息。发布数据将指定为 msgsend 的 message_body 参数。

如果 `msgType` 为 `request`，则将答复消息发送给 `replyToQmgr` 和 `replyToQueue`。

- 只有 `rhfCommand` 为 `registerSubscriber` 时，第 118 页的表 3-20 中的属性才有效。

发送至 MQ 发布 / 预订代理控制队列的 `registerSubscriber` 命令消息会通知该代理，发布者将发布或可以发布有关一个或多个指定主题的数据。如果该发布者已经注册，并且没有任何其它错误，则对发布者的注册信息进行相应的修改。

如果 `msgType` 为 `request`，则将答复消息发送给 `replyToQmgr` 和 `replyToQueue`。

- 只有 `rhfCommand` 为 `requestUpdate` 时，第 119 页的表 3-21 中的属性才有效。

发送至 MQ 发布 / 预订代理控制队列的 `requestUpdate` 命令消息会通知该代理，预订者希望代理转发与指定主题匹配的所有保留发布。

如果 `msgType` 为 `request`，则将答复消息发送给 `replyToQmgr` 和 `replyToQueue`。

scalar_expression

如果消息是一个 SQL *scalar_expression*，则它可以是任意数据类型。

如果未指定 `type` 选项，则当 *scalar_expression* 的计算结果为 `character` 数据类型时，消息类型为 `text`；否则消息类型为 `bytes`。

如果 *scalar_expression* 的数据类型不是 `character`，则使用关于隐式转换的常规 SQL 规则将其转换为 `varbinary`。数据类型的 `binary` 值将按照主机的字节顺序包括在消息中。

basic_character_expression

数据类型为 `char`、`varchar` 或 `java.lang.String` 的 Transact-SQL 查询表达式。

(select_for_xml)

指定 `for xml` 子句的 `select` 表达式。

在作为 *select_for_xml* 参数的 *message_body* 中，*select_for_xml* 会生成一个以 SQLX 格式表示的 SQL 结果集。

只有在配置 Adaptive Server 以获取本机 XML 功能时，才能指定 *select_for_xml*。*select_for_xml* 只能作为标量表达式从 `msgsend` 调用中引用。

必须在 *select_for_xml* 两侧添加小括号。

header_clause

仅允许用户指定第 104 页的表 3-13（针对 MQ）和第 112 页的表 3-14（针对 Tibco JMS）中指定的标头属性。如果输入无法识别的标头属性，则会显示一条错误消息。

如果在 *message property* 和 *message header* 子句中都指定了一个已经识别的标头属性，则优先使用 *message header* 子句中的标头属性。

如果在 *message header* 参数中指定了任何无法识别的名称，则会显示一条错误消息。

示例 **示例 1** SonicMQ JMS 一向指定的结束点发送消息 “hello”：

```
select msgsend('hello',
  'sonicmq_jms:tcp://mysonic:7223?queue=testq,user=xyz')
```

示例 2 (JMS) 向指定的结束点发送消息 “Hello Messaging World!”：

```
declare @mymsg varchar (255)
set @mymsg = 'Hello Messaging World!'
select msgsend(@mymsg,
  +'my_jms_provider?queue=queue.sample,user=jms_user1,'
  +'password=jms_user1_password')
```

示例 3 Tibco JMS 一向发送一条消息，其正文为 SQL 查询返回给指定结束点的 SQL 结果集（采用 SQLX 格式的表达形式）：

```
select msgsend ((select * from pubs2..publishers FOR XML),
  'tibco_jms:tcp://my_jms_host:7222?queue=queue.sample,'
  +'user=jms_user1,password=jms_user1_password')
```

示例 4 (JMS) 设置两个属性并为消息生成 XML schema:

```
select msgsend
((select pub_name from pubs2..publishers where pub_id = '1389' FOR XML),
  my_jms_provider?queue=queue.sample',
  message property 'priority=6, correlationID=MSG_001',
  option 'schema=yes')
```

示例 5 (JMS) 显示用户为消息属性指定的值:

```
select msgsend ('hello', 'my_jms_provider?queue=queue.sample'
  message property 'ttl=30,category=5, rate=0.57, rank='top',
  priority=6')
```

ttl 和 *priority* 在内部设置为标头属性。 *category*、*rate* 和 *rank* 设置为用户指定的消息属性。

示例 6 (MQ) 发送一条请求消息，并预期由同一队列管理器中的指定队列提供答复。

```
select msgsend('do something',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  option 'msgType=request'
  message property 'replyToQueue=QUEUE.REPLY')
```

示例 7 (MQ) 发送一条答复消息。相关 ID 和答复队列已从之前收到的请求消息中提取：

```
select @correlationId = msgpropvalue("CorrelId", @@msgheader)
select @replyQ = @@msgreplytoinfo
select msgsend('i'm done', @replyQ
  option 'msgType=report'
  message property 'correlationId=' + @correlationId)
```

示例 8 (MQ) 发送一条报告消息。相关 ID、答复队列和报告消息数据标头已从之前收到的请求消息中提取：

```
select @correlationId = msgpropvalue("CorrelId", @@msgheader)
select @replyQ = @@msgreplytoinfo
select msgsend(@reportData, @replyQ
  option 'msgType=report'
  message property 'correlationId=' + @correlationId)
```

示例 9 (MQ) 发送 4 条数据报消息。每条消息都是 “theGroup” 组中的一部分，且序列号依次递增：

```
begin tran
select msgsend('message 1',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'groupId=theGroup,sequenceId=1')
select msgsend('message 2',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'groupId=theGroup,sequenceId=2')
select msgsend('message 3',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'groupId=theGroup,sequenceId=3')
select msgsend('message 4',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'groupId=theGroup,sequenceId=4,lastMsgInGroup=yes')
commit
```

示例 10 (MQ) 发送一条数据报消息。请求各种确认报告并发送给“myReplyQueue”：

```
select msgsend('I want a confirmation',
  'ibm_mq:channel1/TCP/host1(5678)?queue=QUEUE.COMMAND',
  message property 'replyToQueue=myReplyQueue'
  + ',exceptionReport=yes,'
  + ',arrivalReport=withData'
  + ',deliveryReport=withFullData')
```

示例 11 (MQ) 发布一条包含主题“A”、“A/B”、“A/B/C”的数据报消息。注册发布者以发布“A”、“A/B”和“A/B/C”主题，且发布内容包含有关“A/B”主题的信息。使用缺省的MQ发布/预订代理队列和流队列：

```
-- First register the publisher
select msgsend(null,
  'ibm_mq:channel1/TCP/host1(5678)?queue=SYSTEM.BROKER.CONTROL.QUEUE'
  option 'msgType=datagram,rfhCommand=registerPublisher'
  message property 'topics='a:A/B:a/b/c''')

-- Now publish the publication
select msgsend('something about A/B',
  'ibm_mq:channel1/TCP/host1(5678)?queue=SYSTEM.BROKER.DEFAULT.STREAM'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics=A/B')
```

示例 12 (MQ) 按一个组发送多条消息。由于ordering设置为logical，因此仅指定msgInGroup、lastMsgInGroup、msgSegment、msgLastSegment选项。由于未指定组名，因此队列管理器将选择一个组名：

```
begin tran
select msgsend('first logical message of the group',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes')

select msgsend('second logical message of the group',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes')

select msgsend('third logical message of the group, first segment',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes,msgSegment=yes')

select msgsend('third logical message of the group, second segment',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
```

```

message property 'ordering=logical,msgInGroup=yes,msgSegment=yes')

select msgsend('third logical message of the group, third segment',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,msgInGroup=yes,msgLastSegment=yes')

select msgsend('fourth logical message of the group',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=QUEUE.COMMAND',
  message property 'ordering=logical,lastMsgInGroup=yes')
commit

```

示例 13 (MQ) 使用 `msgsend` 中的 `alter_user=yes` 选项允许用户 Joe (SQL 登录名为 “joe”) 通过 Adaptive Server 与计算机 “host1” 上运行的 MQ 应用程序之间发送和接收消息，即使 host1 上没有名为 “joe” 的用户 ID 也是如此。

```

select msgsend('Hello world',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=joeQM,queue=QUEUE1,alter_user=yes')

```

示例 14 (MQ) 使用 `msgsend` 注册一个预订者，然后取消对该预订者的注册。预订者对与主题 “A” 或 “A/B/*” 匹配的所有发布内容均感兴趣。匹配的发布内容将由 MQ 发布 / 预订代理转发给队列 “Q2”：

```

-- 注册预订者
select msgsend(null,
  'ibm_mq:channel1/TCP/host1(5678)'
  + '?qmgr=QM,queue=SYSTEM.BROKER.CONTROL.QUEUE'
  option 'msgType=datagram,rfhCommand=registerSubscriber'
  message property 'topics='A:A/B/*',streamName=stream1,queueName=Q2')

-- 向流队列中发布消息，并使其执行隐式注册
select msgsend('happy birthday',
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,
  queue=stream1'
  option 'msgType=datagram,rfhCommand=publish'
  message property 'topics='A''')

-- 读取由 MQ 发布 / 预订转发给我们的消息
select msgrecv(
  'ibm_mq:channel1/TCP/host1(5678)?qmgr=QM,queue=Q2'
  option 'timeout=50ss')

-- 取消对预订者的注册
select msgsend(null,
  'ibm_mq:channel1/TCP/host1(5678)'
  + '?qmgr=QM,queue=SYSTEM.BROKER.CONTROL.QUEUE'
  option 'msgType=datagram,rfhCommand=deregisterSubscriber'
  message property 'topics='A:A/B/*',streamName=stream1,queueName=Q2')

```

示例 15 (MQ) 显示 msgsend 中的 clustQBinding=bind 选项。本地 “INVC” 队列管理器是 Q1 集群队列的成员，Q1 是集群队列。

```
select msgsend(
    "M", "ibm_mq:CH1/TCP/box1(5599)?qmgr=INVC,queue=Q1,alter_user=yes",
    option "clustQBinding=bind")
```

最初运行此 select 语句时，MQOPEN 调用会选择集群队列管理器来接收消息。在同一 SQL 会话中发出的后续语句会自动路由到同一队列管理器。

示例 16 (MQ) 显示 msgsend 中的 clustQBinding = nobind 选项。每次都要选择接收消息的集群队列管理器：

```
select msgsend(
    "M", "ibm_mq:CH1/TCP/box1(5599)?qmgr=INVC,queue=Q1,alter_user=yes",
    option "clustQBinding=nobind")
```

示例 17 (MQ) 显示 msgsend 中的 clustQBinding = default 选项，其行为由队列的 “DEFBIND” 属性决定。如果值为 “open”，则行为与 clustQBinding=bind 相同；否则值与 clustQBinding=nobind 相同：

```
select msgsend(
    "M", "ibm_mq:CH1/TCP/box1(5599)?qmgr=INVC,queue=Q1,alter_user=yes",
    option "clustQBinding=default")
```

用法

- 如果目标的格式为 queue=queue_name，则将消息发送至此队列。
- service_provider_class 以及单词 “user” 和 “password” 不区分大小写。local_name、hostname、port、queue_name、user_name 和 password 参数区分大小写。
- 您可以按照第 46 页的表 3-3 中的说明设置特定于 Adaptive Server 的消息属性。
- msgsend 中的选项字符串用法：
 - 将忽略空选项字符串。
 - 可用逗号或空格分隔各个选项字符串（在第一个选项之前、最后一个选项之后、两个选项之间以及等号两侧，均没有对空格数量的限制）。
 - 带引号的字符串按照关于嵌入式引号的 SQL 约定生成。
 - 如果指定了同名的多个选项，则只处理最后列出的选项。例如，在以下语句中，只对 'priority' 使用或验证值 7；其它值将被忽略：

```
select msgsend('Hello Messaging World!',
    'my_jms_provider?queue=queue.sample',
    MESSAGE PROPERTY 'priority='high', priority=yes, priority=7')
```


- 执行 `msgsend` 后，将使用该调用的信息设置全局变量的值。请参见第 38 页的“与消息相关的全局变量”。
- 带引号的选项或属性值两侧应使用单引号 (') 而非双引号 (")。

注释 如果将 `topic=topic_name` 指定为目标，`msgsend` 允许将消息发送到某一主题。但是，Sybase 建议不要这样做，因为这可能会导致意外行为。

- 无法识别的选项或属性将被忽略，但无法识别的选项值或属性值将标记为错误。

注释 与早期版本不同，此行为在 Adaptive Server 12.5.3a 版中已发生更改。

`msgsend option option_string` 参数值

表 3-11 列出了适用于 JMS 的 `msgsend` 选项参数。

表 3-11: `msgsend` 的有效 JMS option option_string 类型和值

类型	值	缺省值	说明
schema	<ul style="list-style-type: none"> • no • yes • "user_schema" 	no	<ul style="list-style-type: none"> • <code>user_schema</code> 是用户提供的 schema，用于描述 <code>message_body</code>。 • no 指示不生成 schema，也不将其作为消息的一部分发送。 • yes 指示 Adaptive Server 将为消息生成 XML schema。yes 仅在使用 <code>select_for_xml</code> 参数的 <code>message_body</code> 中有意义。<code>select_for_xml</code> 会生成以 SQLX 格式表示的 SQL 结果集。生成的 XML schema 是一个 SQLX 格式的 schema，用于描述结果集文档。 <p>该 schema 将作为 ASE_MSGBODY_SCHEMA 属性包括在消息中。</p>
type	text、bytes	text	要发送的消息的类型。

表 3-12 列出了适用于 MQ 的 `msgsend option` 参数。

表 3-12: `msgsend` 的有效 MQ option option_string 类型和值

类型	值	缺省值	说明
msgType	<ul style="list-style-type: none"> • datagram • request • reply • report 	datagram	<p>如果消息的类型为：</p> <ul style="list-style-type: none"> • request — 还必须指定 <code>replyQueue</code> 属性。 • report — 还必须指定 <code>reportDataHeader</code> 和 <code>feedback</code> 属性。

类型	值	缺省值	说明
rfhCommand	<ul style="list-style-type: none"> • null • deletePublication • deregisterPublisher • deregisterSubscriber • publish • registerPublisher • registerSubscriber • requestUpdate 	null	<p>MQ 发布 / 预订的 MQRF 标头是发送给队列并由 MQ 发布 / 预订代理读取的控制消息。该代理对从队列中读取的消息进行操作。</p> <p>如果 rfhCommand 为 null，则消息不包括 MQRF 标头。如果 rfhCommand 为任何其它值，并且 MQPSCCommand 设置为以下选项之一，则消息包括 MQRF 标头：</p> <ul style="list-style-type: none"> • deletePublication — 设置为 DeletePub。结束点为要发布流队列的结束点。请参见第 113 页的表 3-15。 • deregisterPublisher — 设置为 DeregPub。请参见第 113 页的表 3-16。 • deregisterSubscriber — 设置为 DeleteSub。请参见第 114 页的表 3-17。 • publish — 设置为 Publish。结束点为要发布流队列的结束点。请参见第 115 页的表 3-18。 • registerPublisher — 设置为 RegPub。请参见第 113 页的表 3-15。 • registerSubscriber — 设置为 RegSub。请参见第 118 页的表 3-20。 • requestUpdate — 设置为 ReqUpdate。请参见第 119 页的表 3-21。 <p>消息将发送给指定的结束点。对于下列选项，请将结束点指定为发布流队列：</p> <ul style="list-style-type: none"> • publish • deletePublication <p>对于下列选项，请将结束点指定为 MQ 发布 / 预订代理控制队列：</p> <ul style="list-style-type: none"> • deregisterPublisher • deregisterSubscriber • registerPublisher • registerSubscriber • requestUpdate

类型	值	缺省值	说明
alter_user	<ul style="list-style-type: none"> • yes • no 	null	<p>alter_user=yes 选项允许获得 messaging_role 权限的用户与运行 MQ 的计算机之间发送和接收消息，即使他们在该计算机上没有操作系统（登录）ID 也不例外。</p> <p>如果未设置此选项，且用户在运行 MQ 的计算机上没有登录 ID，MQ 鉴定和消息传送操作都将失败。</p> <hr/> <p>注释 如果运行 MQ 的计算机未同时运行 Adaptive Server Enterprise，则即使在运行 alter_user=yes 后，用户也会看到错误消息。若要避免此情况，请在 MQ 计算机上创建一个新登录名，并使之与启动 Adaptive Server 的用户的用户 ID 完全相同。</p>
clustQBinding	<ul style="list-style-type: none"> • bind • nobind • default 	default	<p>clustQBinding 选项允许用户指定是否要将消息放入同一个实例中。如果不将消息发送至集群队列，则会忽略此选项。如果指定：</p> <ul style="list-style-type: none"> • bind — WebSphere MQ 在首次打开消息时会选择其目标以及承载自身的队列管理器，以确定对执行 MPOPEN 调用时确定的目标的所有 MQPUT 调用。 • nobind — 每次请求 MQ 将消息放入队列时，WebSphere MQ 都会使用每次通过 MPOPEN 调用获取的集群队列句柄执行 MQPUT 时所选择的名称，为消息选择不同的目标。消息的去向取决于负载均衡考虑（如果已启用此选项）以及队列管理器的可用性。 • default — 由在集群队列定义级别定义的绑定属性驱动的目标。在使用集群系统但没有指定 clustQBinding 选项时，也会发生此行为。

msgsend *properties_clause* 参数值

表 3-13 列出了适用于 MQ 的 msgsend *properties_clause* 参数。

表 3-13: msgsend 的有效 MQ message property property_option_clause 类型和值

类型	值	缺省值	说明
arrivalReport	<ul style="list-style-type: none"> • yes • withData • withFullData • no 	no	<p>此消息达到最终目标时应生成一个到达确认 (COA) 报告。</p> <p>必须指定 replyToQueue。如果指定：</p> <ul style="list-style-type: none"> • yes — 生成 COA 报告但不包括已接收消息中的数据。 • withData — 生成 COA 报告并包括已接收消息中前 100 字节的数据。 • withFullData — 生成 COA 报告并包括已接收消息中的全部数据。 • no — 不生成 COA 报告。
correlationId	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>客户端设置相关 ID 以将各个消息关联在一起。MQ 将此字符串的长度限制为 24 字节。</p> <p>MQ 将此字段定义为 unsigned char，以指示其支持 binary 值。若要输入一个 binary 字符串作为 correlationId，请使用 “0x...” 作为值；请勿在值两侧添加引号。</p> <p>如果 rfhCommand 不是 null：</p> <ul style="list-style-type: none"> • 如果 correlationId 不是 null，则不请求新的相关 ID。如果 correlationAsId 为 yes，并且 correlationId 为 null，则这是一个单独的传统标识（相关 ID 为空）。 • 对于 deletePublication、deregisterPublisher、publish 和 registerPublisher 的 rfhCommands，指定的相关 ID 是发布者传统标识的一部分。
deliveryReport	<ul style="list-style-type: none"> • yes • withData • withFullData • no 	no	<p>从最终目标传递此消息时会生成一个到达确认 (COA) 报告。</p> <p>必须指定 replyToQueue。如果：</p> <ul style="list-style-type: none"> • yes — 生成 COA 报告但不包括已接收消息中的数据。 • withData — 生成 COA 报告并包括已接收消息中前 100 字节的数据。 • withFullData — 生成 COA 报告并包括已接收消息中的全部数据。 • no — 不生成 COA 报告。

类型	值	缺省值	说明
exceptionReport	<ul style="list-style-type: none"> • yes • withData • withFullData • no 	no	<p>此消息到期或此发送失败均会生成例外报告。必须指定 <code>replyToQueue</code>。如果：</p> <ul style="list-style-type: none"> • yes — 生成例外报告但不包括已接收消息中的数据。 • withData — 生成例外报告并包括已接收消息中前 100 字节的数据。 • withFullData — 生成例外报告并包括已接收消息中的全部数据。 • no — 不生成例外报告。
expirationReport	<ul style="list-style-type: none"> • yes • withData • withFullData • no 	no	<p>此发送失败将生成例外报告。必须指定 <code>replyToQueue</code>。如果：</p> <ul style="list-style-type: none"> • yes — 生成例外报告但不包括已接收消息中的数据。 • withData — 生成例外报告并包括已接收消息中前 100 字节的数据。 • withFullData — 生成例外报告并包括已接收消息中的全部数据。 • no — 不生成例外报告。
expiry	介于 -1 和 214748364799 之间的 timespec	-1、永不到期	<p>消息在队列管理器中的生存时间。如果 <code>timespec</code> 是一个整数，则单位为毫秒。值包括：</p> <ul style="list-style-type: none"> • 0 — 消息永不到期。 • -1 — 使用为队列定义的缺省值。 <hr/> <p>注释 <code>expiry</code> 的单位为秒的十分之一，因此在将该数字传递给 MQ 之前，会将其舍入到秒的十分之一。</p> <hr/> <p>请参见第 130 页的 <code>timespec</code>。</p>
feedback	<p><i>integer</i></p> <p>范围必须介于 MQFB_APPL_FIRST (65536) 与 MQFB_APPL_LAST (99999999) 之间</p>	0	<p>对于报告消息，<code>feedback</code> 为指示报告消息性质的代码。MQ 会分别为以下消息定义一个 <code>feedback</code> 代码：</p> <ul style="list-style-type: none"> • 系统报告消息 • 应用程序报告消息

类型	值	缺省值	说明
formatName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>应用程序定义的属性，用于传递有关消息格式的信息。</p> <p>此属性允许发送应用程序设置一个描述消息数据的格式名称。</p> <p>接收应用程序可通过检查 <code>@@msgheader</code> 中的 <code>formatName</code> 来确定消息数据的处理方式。</p> <p>以“MQ”开头的名称为保留名称。</p> <p>MQ 将此字符串的长度限制为 8 字节。</p>
groupId	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>用户定义的组。</p> <p>MQ 将此字符串的长度限制为 24 字节。</p> <p>MQ 将此字段定义为 <code>unsigned char</code>，以指示其支持 <code>binary</code> 值。若要输入一个二进制字符串作为 <code>groupId</code>，请使用“0x...”作为值。请勿在值两侧添加引号，否则会将该值解释为带引号的字符串。</p> <p>如果未指定 <code>groupId</code>，但指定了某个分组属性，则队列管理器将生成组名。</p> <p>如果 <code>ordering</code> 设置为 <code>logical</code>，则将其忽略。</p> <p>一个组中的所有消息必须在同一事务中发送。</p>
lastMsgInGroup	<ul style="list-style-type: none"> • yes • no 	no	<p>如果值为 <code>yes</code>，则将消息标记为组中的最后一条逻辑消息。</p> <p>若要使一个组中只有一条逻辑消息，必须将 <code>lastMsgInGroup</code> 设置为 <code>yes</code>。</p> <p>一个组中的所有消息必须在同一事务中发送。</p>
mode	<ul style="list-style-type: none"> • <code>persistent</code> • <code>non-persistent</code> • <code>default</code> 	default	<p>如果 <code>mode</code> 为：</p> <ul style="list-style-type: none"> • <code>persistent</code> — 消息处理提供程序使用稳定存储支持消息。如果消息传送提供程序在消息受到使用前失败，则消息很可能会保存下来。 • <code>non-persistent</code> — 当消息传送提供程序失败时，消息可能会在到达所需目标前丢失。 • <code>default</code> — 使用为队列定义的缺省值。
msgId	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>指定此类型时，WebSphere MQ 会使用为 <code>msgId</code> 指定的值替换所有现有消息 ID。</p> <p>MQ 将此字符串的长度限制为 24 字节。</p> <p>MQ 将此字段定义为“<code>unsigned char</code>”，以指示其支持 <code>binary</code> 值。</p> <p>若要输入一个二进制字符串作为 <code>msgId</code>，请使用“0x...”作为值。请勿在值两侧添加引号。</p>

类型	值	缺省值	说明
msgInGroup	<ul style="list-style-type: none"> • yes • no 	no	<p>如果值为 yes，则此消息为消息组的逻辑消息。对于组中的所有逻辑消息，必须将此属性设置为 yes；但最后一条消息除外，应将该消息的 lastMsgInGroup 设置为 yes。</p> <p>一个组中的所有消息必须在同一事务中发送。</p>
msgLastSegment	<ul style="list-style-type: none"> • yes • no 	no	<p>如果值为 yes，则此消息为分段消息的最后一段。若要使某个逻辑消息自成一段消息，必须将该消息的 msgLastSegment 设置为 yes。</p> <p>当值为 yes 且 ordering 设置为 physical 时，还必须设置 offset 属性。</p> <p>一个组中的所有消息必须在同一事务中发送。</p>
msgSegment	<ul style="list-style-type: none"> • yes • no 	no	<p>如果值为 yes，则此消息为分段消息中的一段。对于单个段中的消息，必须将所有段的这一属性设置为 yes；但最后一段除外，应将该段的 msgLastSegment 设置为 yes。</p> <p>当值为 yes 且 ordering 设置为 physical 时，还必须设置 offset 属性。</p> <p>一个组中的所有消息必须在同一事务中发送。</p>
negativeActionReport	<ul style="list-style-type: none"> • yes • no 	no	<p>必须指定 replyToQueue。如果：</p> <ul style="list-style-type: none"> • yes — 当检索应用程序读取此消息并对其进行负面处理后，将生成一个负作用 (NAN) 报告。 • no — 不生成 NAN 报告。
offset	-1 以及介于 0–maxint 之间的 <i>integer</i>	-1	<p>当消息为分段消息中的一段时，应将 offset 设置为当前消息在逻辑消息中的字节偏移量。</p> <p>-1 指示不指定 offset。</p> <p>如果未同时指定 msgSegment 或 msgLastSegment，将忽略 offset。</p> <p>msgpublish 会忽略此属性。</p> <p>如果 ordering 设置为 logical，则将其忽略。</p> <p>一个组中的所有消息必须在同一事务中发送。</p>
onNoDelivery	<ul style="list-style-type: none"> • deadLetter • discard 	deadLetter	<p>如果：</p> <ul style="list-style-type: none"> • deadLetter — 如果无法传递消息，则将其放置在死字母队列中。 • discard — 队列管理器将放弃该消息。

类型	值	缺省值	说明
ordering	<ul style="list-style-type: none"> logical physical 	physical	<p>如果此属性为：</p> <ul style="list-style-type: none"> physical — 应用程序可按任意顺序发送组（或分段消息）中的消息。如果检测到段缺失或序列标识符之间存在间隔，队列管理器将返回错误。 logical — 应用程序只需对 msgInGroup、lastMsgInGroup、msgSegment 和 lastMsgSegment 选项进行相应设置。队列管理器会自动设置组名、序列标识符和段偏移量。
positiveActionReport	<ul style="list-style-type: none"> yes no 	no	<p>必须指定 replyToQueue。如果：</p> <ul style="list-style-type: none"> yes — 当检索应用程序读取此消息并对其进行正面处理后，将生成一个正作用 (PAN) 报告。 no — 不生成 PAN 报告。
priority	<p><i>integer:</i></p> <ul style="list-style-type: none"> -1, 0 到队列管理器 配置的 max priority 	-1	<p>控制消息的优先级。如果：</p> <ul style="list-style-type: none"> -1 — 使用为队列定义的缺省优先级。 指定的 priority 大于为队列管理器定义的 max priority — 使用为队列管理器定义的 max priority。此行为由 MQ 实现。
replyCorrelationId	<ul style="list-style-type: none"> msgId correlationId 	msgId	<p>如果：</p> <ul style="list-style-type: none"> msgId — 报告消息中的相关 ID 使用已接收消息的消息 ID。 correlationId — 报告消息中的相关 ID 使用已接收消息的相关 ID。
replyMsgId	<ul style="list-style-type: none"> new original 	new	<p>如果：</p> <ul style="list-style-type: none"> new — 生成的报告消息包含新的消息 ID。 original — 报告消息使用与已接收消息相同的消息 ID。

类型	值	缺省值	说明
replyToInputMode	<ul style="list-style-type: none"> • browse • Qdefault • shared • exclusive • browse+Qdefault • browse+shared • browse+exclusive 	Qdefault	<p>打开 replyToQueue 时所处的模式。</p> <p>如果指定 replyToQueue，队列会自动打开以进行后续输入。此模式指定打开 replyToQueue 时所处的输入模式。</p> <p>如果未指定 replyToQueue，则忽略此属性。</p> <p>这些模式的含义如下：</p> <ul style="list-style-type: none"> • browse — 打开队列仅供浏览。如果尝试进行破坏性读取，队列管理器将发出错误消息。 • Qdefault — 使用为队列定义的缺省输入模式打开队列。 • shared — 在共享输入模式下打开队列。如果队列已由另一个 MQ 句柄在独占模式下打开，则会显示错误消息。 • exclusive — 在独占输入模式下打开队列。如果队列已由另一个 MQ 句柄在共享或独占模式下打开，则会显示错误。 • browse+Qdefault — 打开队列以进行浏览，并使用为队列定义的缺省输入模式。 • browse+shared — 打开队列以进行浏览，并使用共享输入模式。如果队列已由另一个 MQ 句柄在独占模式下打开，则会显示错误消息。 • browse+exclusive — 打开队列以进行浏览，并使用独占输入模式。如果队列已由另一个 MQ 句柄在共享或独占模式下打开，则会显示错误。
replyToModel	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>基于其创建答复队列的模型队列的名称（当 replyToQueue 为动态队列时）。</p> <p>如果未指定 replyToQueue，则忽略此属性。</p> <p>MQ 将此字符串的长度限制为 48 字节。</p>
replyToQmgr	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>为 replyToQueue 未来所在的队列管理器保留。</p> <p>目前，replyToQueue 始终位于已连接的队列管理器上。</p>

类型	值	缺省值	说明
replyToQueue	<ul style="list-style-type: none"> • null • <i>string</i> 	null	<p>应用程序预期可从中获取对请求消息的答复的队列。</p> <hr/> <p>注释 已发送的消息类型不必为 <code>request</code>，因为 MQ 并不强制如此。</p> <hr/> <p>如果指定的队列名称以 “*” 结尾，则系统会使用指定前缀生成一个动态队列名称。</p> <p>如果指定了 <code>replyToModel</code> 和动态队列名称，则基于为 <code>replyToModel</code> 指定的模型队列创建动态队列。</p> <p>执行发送操作后，可通过 <code>@@msgreplytoinfo</code> 会话变量获取系统生成的动态队列名称。</p> <hr/> <p>注释 指定动态队列名称后，当前 Adaptive Server 登录名必须在队列管理器中具有 “crt” 授权才能创建动态队列。</p> <hr/> <p>指定动态队列名称后，如果接收应用程序未删除该动态队列，则必须手动执行此操作。</p> <p>如果 <code>rfhCommand</code> 不是 <code>null</code>，则可以指定 <code>replyToQueue</code> 从 MQ 发布 / 预订代理获取响应。</p>

类型	值	缺省值	说明
rfhCommand	<ul style="list-style-type: none"> • null • deletePublication • deregisterPublisher • deregisterSubscriber • publish • registerPublisher • registerSubscriber • requestUpdate 	null	<p>MQ 发布 / 预订的 MQRF 标头是发送给队列并由 MQ 发布 / 预订代理读取的控制消息。该代理对其从队列中读取的消息进行操作。</p> <p>如果 rfhCommand 为 null，则消息不包括 MQRF 标头。如果 rfhCommand 为任何其它值，并且 MQPSCCommand 设置为以下选项之一，则消息包括 MQRF 标头：</p> <ul style="list-style-type: none"> • deletePublication — 设置为 DeletePub。结束点为要发布流队列的结束点。请参见第 113 页的表 3-15。 • deregisterPublisher — 设置为 DeregPub。请参见第 113 页的表 3-16。 • deregisterSubscriber — 设置为 DeleteSub。请参见第 114 页的表 3-17。 • publish — 设置为 Publish。结束点为要发布流队列的结束点。请参见第 115 页的表 3-18。 • registerPublisher — 设置为 RegPub。请参见第 113 页的表 3-15。 • registerSubscriber — 设置为 RegSub。请参见第 113 页的表 3-15。 • requestUpdate — 设置为 ReqUpdate。请参见第 113 页的表 3-15。 <p>消息将发送给指定的结束点。对于下列选项，请将结束点指定为发布流队列：</p> <ul style="list-style-type: none"> • publish • deletePublication <p>对于下列选项，请将结束点指定为 MQ 发布 / 预订代理控制队列：</p> <ul style="list-style-type: none"> • deregisterPublisher • deregisterSubscriber • registerPublisher • registerSubscriber • requestUpdate
sequenceld	介于 -1 – 9,999,999 之间的 <i>integer</i>	-1	<p>用于排列组中的逻辑消息。</p> <p>-1 指示不指定 sequenceld。</p> <p>如果未同时指定 msgInGroup 或 lastMsgInGroup，则会忽略 sequenceld。</p> <p>msgpublish 会忽略此属性。</p> <p>如果 ordering 设置为 logical，则将其忽略。</p> <p>一个组中的所有消息必须在同一事务中发送。</p>

表 3-14 列出了适用于 JMS 的 `msgsend properties_clause` 参数。

表 3-14: `msgsend` 的有效 JMS message property properties_option_string 类型和值

选项	值	缺省值和放置	说明
ttl	0 – (2 ⁶³ – 1)	0 标头	ttl 指在消息总线中的生存时间。Adaptive Server 不受此属性影响。 到期信息是消息保持有效的持续时间，单位为毫秒。例如，60 指示消息的生存期为 60 毫秒。 值 0 指示消息永不到期。 ttl 使用 timespec 选项。请参见第 130 页的 timespec。
priority	1 – 9	4 标头	priority 的行为由基础消息总线控制。此处提到的值适用于 Tibco JMS。 1 – 4 的 priority 为正常区间；5 – 9 的 priority 为加速区间。
correlation	string	无 标头	客户端应用程序设置相关 ID 以将各个消息关联在一起。Adaptive Server 设置由应用程序指定的相关 ID。
mode	<ul style="list-style-type: none"> • persistent • non-persistent 	persistent 标头	如果模式为： <ul style="list-style-type: none"> • persistent — JMS 提供程序使用稳定存储支持消息。如果消息传送提供程序在消息受到使用前失败，则消息很可能会保存下来。 • non-persistent — 当消息传送提供程序失败时，消息可能会在到达所需目标前丢失。
replyqueue	一个包含 <code>queue_name</code> 的字符串	无 标头	如果 <code>queue_name</code> 或 <code>topic_name</code> 的值为： <ul style="list-style-type: none"> • <code>syb_temp</code> — Adaptive Server 会创建一个临时目标，并将新建临时目标的相关信息作为标头信息的一部分发送。 Adaptive Server 随后会将 <code>@@msgreplytoinfo</code> 更新为新建的临时目标。 临时目标的类型（队列或主题）取决于指定的是 <code>replyqueue</code> 还是 <code>replytopic</code>。仅使用最后列出的选项。 • 已存在的目标 — Adaptive Server 不创建新目标，而是使用用户指定的目标。
replytopic	一个包含 <code>topic_name</code> 的字符串	无 标头	

`msgsend` 属性和 `rhfCommand`

对于 MQ，只有 `rhfCommand` 为 `deletePublication` 时，表 3-15 中的属性才有效。

表 3-15: *rfhCommand* 设置为 *deletePublications* 时的 *msgsend* 属性

属性	值	缺省值	说明
local	<ul style="list-style-type: none"> • yes • no 	no	如果: <ul style="list-style-type: none"> • yes — 仅删除在此代理上本地发布的保留发布。 • no — 从网络中的所有代理上删除全局保留发布。
streamName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	指定主题的发布流的名称。 如果未指定, 则缺省值为要将此 MQRFH 命令消息发送到流队列。 MQ 将此字符串的长度限制为 48 字节。
topics	<i>string</i>	无	使用第 15 页的“主题的语法”中详述的格式。 将删除与此主题匹配的保留消息。 必须至少提供一个主题。 这是一个必需的属性; 如果省略, 将生成错误。

对于 MQ, 只有 *rfhCommand* 为 *deregisterPublisher* 时, 表 3-15 中的属性才有效。

表 3-16: *rfhCommand* 设置为 *deregisterPublisher* 时的 *msgsend* 属性

属性	值	缺省值	说明
deregAll	<ul style="list-style-type: none"> • yes • no 	no	如果: <ul style="list-style-type: none"> • yes — 为此发布者取消对所有主题的注册, 并忽略 <i>topics</i> 属性。 • no — 不取消对任何已注册主题的注册。 如果指定 <i>topics</i> , Adaptive Server 将返回错误。
streamName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	如果: <ul style="list-style-type: none"> • Not null — 这是发布流的名称。 • null — 采用 SYSTEM.BROKER.DEFAULT.STREAM。 MQ 将此字符串的长度限制为 48 字节。
topics	<ul style="list-style-type: none"> • null • <i>string</i> 	null	使用第 15 页的“主题的语法”中详述的格式。 这些都是发布者取消注册的主题。 在下列情况中, Adaptive Server 将返回错误: <ul style="list-style-type: none"> • <i>deregAll</i> 属性设置为 yes。 • <i>topics</i> 不是 null。
qmgrName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	这是发布者的队列管理器名称, 用于建立发布者的传统标识。应将其指定为注册发布者时指定的值。 如果为 null, 则缺省为 <i>replyToQmgr</i> 。
queueName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	这是发布者的队列名称, 用于建立发布者的传统标识。应将其指定为注册发布者时指定的值。 如果为 null, 则缺省为 <i>replyToQueue</i> 。

属性	值	缺省值	说明
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	如果： <ul style="list-style-type: none"> • yes — 将 correlationId 用作发布者传统标识的一部分。必须指定 correlationId，但不能将其指定为 0x00。 • no — 不将 correlationId 用作发布者传统标识的一部分。 • generate — 将系统生成的 correlationId 用作发布者传统标识的一部分。

对于 MQ，只有 rhfCommand 为 deregisterSubscriber 时，表 3-17 中的属性才有效。

表 3-17: rhfCommand 设置为 deregisterSubscriber 时的 msgsend 属性

属性	值	缺省值	说明
deregAll	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — 为此预订者取消对所有主题的注册。同时忽略 topics 属性。 • no — 不取消对任何预订者主题的注册。 如果 topics 不是 null，Adaptive Server 将返回错误
streamName	<ul style="list-style-type: none"> • null • string 	null	如果： <ul style="list-style-type: none"> • Not null — 这是发布流的名称。 • null — 采用 SYSTEM.BROKER.DEFAULT.STREAM。 MQ 将此字符串的长度限制为 48 字节。
topics	<ul style="list-style-type: none"> • null • string 	null	使用第 15 页的“主题的语法”中详述的格式。这些是此预订者取消注册的主题。在下列情况中，Adaptive Server 将返回错误： <ul style="list-style-type: none"> • deregAll 为 Yes。 • topics 不是 null。
qmgrName	<ul style="list-style-type: none"> • null • string 	null	这是预订者的队列管理器名称，用于建立预订者的传统标识。应将其指定为注册预订者时指定的值。如果为 null，则缺省为 replyToQmgr。
queueName	<ul style="list-style-type: none"> • null • string 	null	这是预订者的队列名称，用于建立预订者的传统标识。应将其指定为注册预订者时指定的值。如果为 null，则缺省为 replyToQueue。
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	如果： <ul style="list-style-type: none"> • yes — 将 correlationId 用作发布者传统标识的一部分。必须指定 correlationId，但不能将其指定为 0x00。 • no — 不将 correlationId 用作发布者传统标识的一部分。 • generate — 将系统生成的 correlationId 用作发布者传统标识的一部分。

对于 MQ，只有 rhfCommand 为 publish 时，表 3-18 中的属性才有效。

表 3-18: rhfCommand 设置为 publish 时的 msgsend 属性

属性	值	缺省值	说明
topics	string	无	<ul style="list-style-type: none"> 使用第 15 页的“主题的语法”中详述的格式。 不允许使用通配符。 这些是与发布中的信息相关的主题。 这是一个必需的属性；如果省略，将生成错误。
anon	<ul style="list-style-type: none"> yes no 	no	如果： <ul style="list-style-type: none"> yes — MQ 发布 / 预订代理不公布发布者的标识。如果 noReg 为 yes，则将其忽略。 no — MQ 发布 / 预订代理公布发布者的标识。
local	<ul style="list-style-type: none"> yes no 	no	如果： <ul style="list-style-type: none"> yes — MQ 发布 / 预订代理仅将此发布发送给注册时指定 local 的预订者。如果 noReg 为 yes，则将其忽略。 no — MQ 发布 / 预订代理会将此发布发送给所有预订者。
directReq	<ul style="list-style-type: none"> yes no 	no	如果： <ul style="list-style-type: none"> yes — 发布者接受其它应用程序对发布信息的直接请求。如果 noReg 为 yes，则将其忽略。 如果已将 anon 属性设置为 yes，请勿将此选项也设置为 yes，否则 MQ 发布 / 预订代理会以错误进行响应。 <ul style="list-style-type: none"> no — 发布者不接受其它应用程序对发布信息的直接请求。
noReg	<ul style="list-style-type: none"> yes no 	no	如果未在 MQ 发布 / 预订代理中将发布者注册为此流和主题发布者，并且 NoReg 的值为： <ul style="list-style-type: none"> yes — MQ 发布 / 预订代理不指定隐式注册。将忽略 anon、local 和 directReq 属性。 no — MQ 发布 / 预订代理使用由 anon、local 和 directReq 设置的值执行隐式注册。 如果发布者已经注册，并且 anon、local 或 directReq 设置为 yes，则根据这些属性更改现有注册。
otherSubsOnly	<ul style="list-style-type: none"> yes no 	no	如果： <ul style="list-style-type: none"> yes — 如果此发布者已预订此发布，MQ 发布 / 预订代理会将此发布发送给此发布者。 no — MQ 发布 / 预订代理不会将此发布发送给此发布者，即使此发布者已预订此发布也是如此。

属性	值	缺省值	说明
publishSequenceld	-1 以及介于 0 – $(2^{32} - 1)$ 之间的 <i>number</i>	-1	如果： <ul style="list-style-type: none"> • Not -1 — 这是发布的序列号。该序列号会随每次发布递增，但 MQ 发布 / 预订代理不会对此进行验证。 • If -1 — 不设置序列号。
publishTimeStamp	<ul style="list-style-type: none"> • null • <i>integer</i> 	null	如果： <ul style="list-style-type: none"> • 非 null — 这是使用通用时间的发布时间戳（格式为 YYYYMMDDHHMMSSth）。系统不会对格式进行验证。 • null — 不设置发布时间戳。
qmgrName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	这是用于确定发布者传统标识的队列管理器，也是预订者可向此发布者发送直接请求的位置。 MQ 将此字符串的长度限制为 48 字节。
queueName	<ul style="list-style-type: none"> • null • <i>string</i> 	null	这是用于确定发布者传统标识的队列，也是预订者可向此发布者发送直接请求的位置。 MQ 将此字符串的长度限制为 48 字节。
retainPub	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — MQ 发布 / 预订代理不会将此发布发送给此发布者，即使此发布者已预订此发布也是如此。 • no — 如果此发布者已预订此发布，MQ 发布 / 预订代理会将此发布发送给此发布者。
stringData	<ul style="list-style-type: none"> • null • <i>string</i> 	null	If not null — 这是发布的 MQRF 标头中包括的发布者定义的可选信息。 注释 虽然 MQ 发布 / 预订允许 MQRF 中存在多个 stringData 标记，但 ASE Active Messaging 仅支持一个标记。
integerData	-1 以及介于 0 – $(2^{32} - 1)$ 之间的 <i>number</i>	-1	如果不是 -1，则为发布的 MQRF 标头中包括的发布者定义的可选信息。 注释 虽然 MQ 发布 / 预订允许 MQRF 标头中存在多个 integerData 标记，但 Active Messaging 仅支持一个标记。
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	如果： <ul style="list-style-type: none"> • yes — 将 correlationId 用作发布者传统标识的一部分。必须指定 correlationId，但不能将其指定为 0x00。 • no — 不将 correlationId 用作发布者传统标识的一部分。 • generate — 将系统生成的 correlationId 用作发布者传统标识的一部分。

对于 MQ，只有 rhfCommand 为 registerPublisher 时，表 3-19 中的属性才有效。

表 3-19: rhfCommand 设置为 registerPublisher 时的 MQ msgsend 属性

属性	值	缺省值	说明
anon	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — MQ 发布 / 预订代理不公布发布者的标识。 • no — MQ 发布 / 预订代理公布发布者的标识。
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	如果： <ul style="list-style-type: none"> • yes — 将 correlationId 用作发布者传统标识的一部分。必须指定 correlationId，但不能将其指定为 0x00。 • no — 不将 correlationId 用作发布者传统标识的一部分。 • generate — 将系统生成的 correlationId 用作发布者传统标识的一部分。
directReq	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — 发布者接受其它应用程序对发布信息的直接请求。 如果已将 anon 属性设置为 yes，请勿将此选项也设置为 yes，否则 MQ 发布 / 预订代理会以错误进行响应。 • no — 发布者不接受其它应用程序对发布信息的直接请求。
local	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — MQ 发布 / 预订代理仅将此发布发送给注册时指定 Local 的预订者。 • no — MQ 发布 / 预订代理会将此发布发送给所有预订者。
qmgrName	<ul style="list-style-type: none"> • null • string 	null	这是用于确定发布者传统标识的队列管理器，也是预订者可向此发布者发送直接请求的位置。 MQ 将此字符串的长度限制为 48 字节。
queueName	<ul style="list-style-type: none"> • null • string 	null	这是用于确定发布者传统标识的队列，也是预订者可向此发布者发送直接请求的位置。 MQ 将此字符串的长度限制为 48 字节。
streamName	<ul style="list-style-type: none"> • null • string 	null	如果： <ul style="list-style-type: none"> • Not null — 这是发布者向其中发布信息的流。 • null — 缺省值为 SYSTEM.BROKER.DEFAULT.STREAM。 MQ 将此字符串的长度限制为 48 字节。
topics	string	无	使用第 15 页的“主题的语法”中详述的格式。 不允许使用通配符。 这些是发布者围绕其提供信息的主题。 这是一个必需的属性；如果省略，将生成错误。

对于 MQ，只有 rfhCommand 为 registerSubscriber 时，表 3-20 中的属性才有效。

表 3-20: rfhCommand 设置为 registerSubscriber 时的 MQ msgsend 属性

属性	值	缺省值	说明
topics	string	无	使用第 15 页的“主题的语法”中详述的格式。这些是预订者希望收到的发布的主题。这是一个必需的属性；如果省略，将生成错误。
anon	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — MQ 发布 / 预订代理不公布预订者的标识。 • no — MQ 发布 / 预订代理公布预订者的标识。
local	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — 不将预订分发到网络中的其它代理。只有指定 local 的发布者从此节点发布的信息才会发送给此预订者。 • no — 不在 RFH 命令中指定预订。
newPubsOnly	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — 代理仅将此发布发送给此预订者，而不发送注册时已有的保留发布。 • no — 不在 RFH 命令中指定发布。
pubOnReqOnly	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — 代理仅向此预订者发送新发布，而不发送注册时已有的保留发布。 • no — 不在 RFH 命令中指定发布。
inclStreamName	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — 代理向转发给预订者的每条消息的 MQRF 标头中添加发布流名称。 • no — 不在 RFH 命令中指定发布。
informIfRet	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — 代理通过在发送给预订者的消息的 MQRF 标头中设置 MQPSPubsOptslsRetainedPub 来通知预订者是否保留发布。 • no — 不在 RFH 命令中指定发布。
dupsOk	<ul style="list-style-type: none"> • yes • no 	no	如果： <ul style="list-style-type: none"> • yes — 允许代理偶尔向预订者传递重复的发布。 • no — 不在 RFH 命令中指定发布。

属性	值	缺省值	说明
pubsPersistence	<ul style="list-style-type: none"> non-persistent persistent asPublication asQueue 	asQueue	如果： <ul style="list-style-type: none"> non-persistent — 将发布作为非持久性消息放入预订者队列。 persistent — 将发布作为持久性消息放入预订者队列。 asPublication — 按照原始发布的持久性将发布放入预订者队列。 asQueue — 使用预订者队列的默认持久性将发布放入预订者队列。
streamName	<ul style="list-style-type: none"> null string 	null	如果： <ul style="list-style-type: none"> Not null — 这是发布者向其中发布信息的流。 null — 预订采用传统标识。
qmgrName	<ul style="list-style-type: none"> null string 	null	这是用于确定预订者传统标识的队列管理器。 MQ 将此字符串的长度限制为 48 字节。
queueName	<ul style="list-style-type: none"> null string 	null	这是用于确定预订者传统标识的队列。 MQ 将此字符串的长度限制为 48 字节。
correlationAsId	<ul style="list-style-type: none"> yes no generate 	no	如果： <ul style="list-style-type: none"> yes — 将 correlationId 用作预订者传统标识的一部分。必须指定 correlationId，但不能将其指定为 0x00。 no — 不将 correlationId 用作预订者传统标识的一部分。 generate — 将系统生成的 correlationId 用作预订者传统标识的一部分。

只有 rhfCommand 为 requestUpdate 时，表 3-21 中的属性才有效。

表 3-21: rhfCommand 设置为 requestUpdate 时的 MQ msgsend 属性

属性	值	缺省值	说明
topics	string	无	使用第 15 页的“主题的语法”中详述的格式。 预订者要请求的主题。 只能提供一个主题。 这是一个必需的属性；如果省略，将生成错误。
streamName	<ul style="list-style-type: none"> null string 	null	如果： <ul style="list-style-type: none"> Not null — 这是发布者向其中发布信息的流。 null — 缺省值为 SYSTEM.BROKER.DEFAULT.STREAM。
qmgrName	<ul style="list-style-type: none"> null string 	null	这是用于确定预订者传统标识的队列管理器名称。应将其指定为注册预订者时指定的值。 MQ 将此字符串的长度限制为 48 字节。

属性	值	缺省值	说明
queueName	<ul style="list-style-type: none"> • null • string 	null	这是用于确定预订者传统标识的队列。应将其指定为注册预订者时指定的值。 MQ 将此字符串的长度限制为 48 字节。
correlationAsId	<ul style="list-style-type: none"> • yes • no • generate 	no	如果： <ul style="list-style-type: none"> • yes — 将 correlationId 用作预订者传统标识的一部分。必须指定 correlationId，但不能将其指定为 0x00。 • no — 不将 correlationId 用作预订者传统标识的一部分。 • generate — 将系统生成的 correlationId 用作预订者传统标识的一部分。

- 如果使用 `message property`，则忽略无法识别的选项。如果在 `msgsend` 或 `msgpublish` 函数中使用 `message header`，则当指定了无法识别的选项时，将会显示错误。
- `msgsend` 调用的结果是一个 `varchar` 字符串。如果消息成功发送，则返回值为消息 ID。如果未发送消息，则返回空值。
- 这些限制适用于 `service_provider_uri` 的运行期格式：

```

service_provider_uri ::=
  provider_name ?destination [,user=username, password=password]
  provider_name ::= local_name | full_name
  local_name ::= identifier
  full_name ::= service_provider_class:service_provider_url

```

- `local_name` 是一个提供程序标识符，之前在调用 `sp_msgadmin 'register', 'provider'` 时注册，是该调用中指定的 `full_name` 的缩写形式。
- 当前支持的唯一 `service_provider_class` 为 JMS。
- `service_provider_url` 的格式为 “`tcp://hostname:port`”。主机名可以是名称或 IP 地址。
- `service_provider_url` 不能包含空格。

MQ

- `msgsend` 返回的状态是向指定队列发送消息的完成状态，而非 MQ 发布 / 预订代理中的状态。若要获取 MQ 发布 / 预订代理中的完成状态，请指定 `replyToQueue`，然后发送一条 `request` 消息或请求 `negativeActionReport`。MQ 发布 / 预订代理会向 `replyToQueue` 发送 `response` 或 `report MQRFH` 消息。在这两种情况下，都必须从 `replyToQueue` 中显式读取 `response` 或 `report` 消息，并检查已收到消息中的 `MQPSCompCode`、`MQPSReason` 和 `MQPSReasonText` 属性。
- 指定 `msgSegment` 或 `msgLastSegment` 时，如果应用程序要读取消息（通过为非 Adaptive Server 应用程序指定 `MQGMO_COMPLETE_MSG` 或为 Adaptive Server 应用程序指定 `completeMsg=yes`），则必须在一个工作单元中发送组成逻辑消息的所有消息，因此需要分在同一个事务中的所有消息必须一起发送。

权限

您必须具有 `messaging_role` 才能运行 `msgsend`。

msgsubscribe

说明 仅限 JMS – 提供 SQL 接口，以预订当前 Adaptive Server 会话的主题。

语法 `msg_subscribe::= msgsubscribe
(subscription_name)
subscription_name::=basic_character_expression`

参数 `subscription_name`
要预订的内容的名称。一个 `basic_character_expression`。

示例 通知 JMS 消息传送提供程序开始保存发布到注册为 “subscription_1” 的主题的消息：

```
select msgsubscribe ('subscription_1')
```

用法

- 使用 `msgsubscribe` 或 `msgunsubscribe` 指定预订之前，必须使用 `sp_msgadmin` 注册预订。以下示例会注册持久性预订 “subscription_1”：

```
sp_msgadmin 'register', 'subscription', 'subscription_1',  
  'my_jms_provider?topic=topic.sample,user=user1,password=pwd',  
  'Supplier=12345', null, 'durable1', 'client1'
```

- 调用 `msgsubscribe` 后，将为 Adaptive Server 会话保存围绕指定主题发布且符合选择器标准的所有消息，直至调用 `msgconsume` 来读取消息。如果在准备使用消息前不想保存达到的消息，请勿调用 `msgsubscribe`。如果在未调用 `msgsubscribe` 的情况下调用 `msgconsume`，将在调用 `msgconsume` 时启动预订。
- `msgsubscribe` 可为客户端启动预订，以接收由结束点和过滤器（由 `subscription_name` 指定）定义的消息。如果成功，则返回 0；如果失败，则返回 1。
- 以下示例显示在应用程序逻辑准备好读取消息前使用 `msgsubscribe`，以强制 JMS 客户端保存消息。应用程序将预订：

```
select msgsubscribe ('subscription_1')
```

客户端多次使用消息，并使用与消息传送无关的其它应用程序逻辑。随后，它准备读取消息，并接收自调用 `msgsubscribe` 以来到达的所有消息：

```
select msgconsume('subscription_1')  
select msgconsume('subscription_1')
```

客户端应用程序完成此预订，然后取消预订：

```
select msgunsubscribe('subscription_1')
```

msgunsubscribe

说明 仅限 JMS — 提供 SQL 接口，以便为当前 Adaptive Server 会话取消对主题的预订。

语法

```
msg_unsubscribe::=msgunsubscribe
    (subscription_name [with {remove | retain}])
    subscription_name::=basic_character_expression
```

参数

subscription_name
要预订的内容的名称。一个 *basic_character_expression*。

with {remove | retain}
在 JMS 消息提供程序中删除或保留持久性预订。

示例 通知 JMS 消息传送提供程序停止保存发布到注册为 “subscription_1” 的主题的消息：

```
select msgunsubscribe('subscription_1')
```

用法

- 使用 `msgsubscribe` 或 `msgunsubscribe` 指定预订之前，必须使用 `sp_msgadmin` 注册预订。以下示例会注册持久性预订 “subscription_1”：

```
sp_msgadmin 'register', 'subscription', 'subscription_1',
    'my_jms_provider?topic=topic.sample,user=user1,password=pwd',
    'Supplier=12345', null, 'durable1', 'client1'
```

- `msgunsubscribe` 会停止当前 Adaptive Server 会话对结束点和过滤器（由 *subscription_name* 指定）的所有当前预订。如果成功，则返回 0；如果失败，则返回 1。
- 如果指定 `with retain`，则终止与 JMS 消息传送提供程序之间的连接，以便其它预订使用该预订中指定的同一预订者 *client_id* 进行连接。持久性预订者会在 Adaptive Server 和 JMS 消息提供程序中保持已定义状态。如果指定 `with remove`，则会从 JMS 消息提供程序中删除持久性预订的定义。缺省值为 `with retain`。

当用户从 Adaptive Server 中注销时，将取消 Adaptive Server 会话中的所有预订。这与使用 `with retain` 选项运行 `msgunsubscribe` 的效果相同。

在使用 `with remove` 取消持久性预订时，将从 JMS 消息提供程序中删除预订者定义，这会导致预订者定义中保存的消息全部丢失：

```
<login>
select msgsubscribe('subscription_1')
select msgconsume('subscription_1')
...
select msgconsume('subscription_1')
select msgunsubscribe('subscription_1' WITH REMOVE)
<logout>
```

```
----Messages published to the topic registered as subscription_1 are no
----longer held by the JMS provider
```

```
<login>
select msgsubscribe('subscription_1')
select msgconsume('subscription_1')
...
select msgconsume('subscription_1')
select msgunsubscribe('subscription_1' WITH REMOVE)
```

在另一种情况下，SQL 会话将释放预订，以便其它会话能够继续使用消息。以下示例显示会话 1 释放预订，以便会话 2 开始使用其中的消息。

表 3-22: SQL 会话

会话 1	会话 2
<pre>select msgunsubscribe ('subscription_1' WITH RETAIN) selectmsgconsume ('subscription_1') ... selectmsgconsume ('subscription_1') select msgunsubscribe ('subscription_1' WITH RETAIN)</pre>	<pre>select msgsubscribe('subscription_1') select msgconsume('subscription_1') ... select msgconsume('subscription_1') select msgunsubscribe('subscription_1' WITH RETAIN)</pre>

- 以下示例显示在应用程序逻辑准备好读取消息前使用 `msgsubscribe`，以强制 JMS 客户端保存消息。应用程序将预订：

```
select msgsubscribe ('subscription_1')
```

客户端多次使用消息，并使用与消息传送无关的其它应用程序逻辑。随后，它准备读取消息，并接收自调用 `msgsubscribe` 以来到达的所有消息：

```
select msgconsume('subscription_1')
select msgconsume('subscription_1')
```

客户端应用程序完成此预订，然后取消预订：

```
select msgunsubscribe('subscription_1')
```


endpoint

说明

(MQ) 指定 *endpoint* 的一般语法和处理方式（针对 WebSphere MQ）。各个选项在接受 *endpoint* 参数的函数和存储过程中均有介绍。

注释 JMS 结束点对 Adaptive Server 是透明的，系统不会对其正确性或有效性进行检查，而是直接将它们发送给 JMS 提供程序。

语法

```

service_provider_uri ::= provider_name?qmgr=qmgr_name,destination
                        provider_name ::= local_name | full_name
                        local_name ::= identifier
                        full_name ::= service_provider_class:service_provider_url
                        service_provider_class ::= ibm_mq
                        service_provider_url ::= [channel]/tcp/hostname(port)
                        channel ::= channel_name[(channel_security)]
                        channel_name ::= identifier
                        channel_security ::= ssl:SSLCIPH=channel_ciph
                        channel_ciph ::= identifier
                        hostname ::= identifier
                        port ::= integer

qmgr_name ::= identifier
destination ::= [remote_qmgr,]queue=queue_name
               remote_qmgr ::= remote_qmgr=remote_qmgr_name
               remote_qmgr_name ::= identifier
               queue_name ::= identifier

```

参数

local_name

已注册的发布者或预订者的名称。

qmgr_name

MQ 队列管理器的名称。MQ 将队列管理器名称的长度限制为 48 个字符（字节）。

ibm_mq

定义服务提供程序类。不区分大小写。

channel_name

MQ 服务器连接通道的名称，在 Adaptive Server 15.0.2 ESD #1 和更高版本中为可选参数。MQ 将通道名称的长度限制为 20 个字符（字节）。如果未定义 *channel_name*，Active Messaging 将使用服务器连接通道“SYSTEM.DEF.SRVCONN”连接队列管理器。

channel_security

通道的安全属性。如果未指定 *channel_security*，Adaptive Server 将在不使用任何安全协议的情况下与 WebSphere MQ 通信。

channel_security 的有效值为 *ssl*。

channel_ciph

与 *channel_security* 配合使用，可指定服务器连接通道的 SSLCIPH 属性值，并且必须是 WebSphere MQ 客户端的有效 **CipherSpec** 值。

channel_ciph 的有效值包括：

表 3-23: channel_ciph 的有效 CipherSpec 名称

CipherSpec 名称	散列算法	加密算法	加密位数
NULL_MD5 ¹	MD5	无	0
NULL_SHA ¹	SHA	无	0
RC4_MD5_EXPORT ¹	MD5	RC4	40
RC4_MD5_US ²	MD5	RC4	128
RC4_SHA_US ²	SHA	RC4	128
RC2_MD5_EXPORT ¹	MD5	RC2	40
DES_SHA_EXPORT ¹	SHA	DES	56
RC4_56_SHA_EXPORT1024 ^{3,4,5}	SHA	RC4	56
DES_SHA_EXPORT1024 ^{3,4,5,6}	SHA	DES	56
TRIPLE_DES_SHA_US ⁴	SHA	3DES	168
TLS_RSA_WITH_AES_128_CBC_SHA ⁷	SHA	AES	128
TSL_RSA_WITH_AES_256_CBC_SHA ⁷	SHA	AES	256
AES_SHA_US ⁸	SHA	AES	128

1 在 OS/400 上，仅在安装 AC2 或 AC3 后可用。

2 在 OS/400 上，仅在安装 AC3 后可用。

3 不适用于 z/OS。

4 不适用于 OS/400。

5 指定 1024 位的握手密钥大小。

6 不适用于 Windows。

7 仅适用于针对 Intel 的 AIX、HP-UX 和 Linux 平台。

8 仅适用于 OS/400、AC3。

tcp

传输协议，不区分大小写。指定 *tcp* 以通过 SSL 与 MQ 通信。

hostname

运行 MQ 监听器的计算机的主机名。

port

MQ 监听器当前监听的端口号。

注释 *hostname(port)* 的组合长度不能超过 264 字节。

queue_name

MQ 队列的名称。MQ 将队列名称的长度限制为 48 个字符（字节）。

remote_qmgr_name

包含目标队列定义的远程 MQ 队列管理器的名称。MQ 将队列管理器名称的长度限制为 48 个字符（字节）。如果使用：

- `msgsend` — 如果省略此选项，则使用本地队列管理器定位队列对象，从而获得集群队列负载均衡的好处。
- `msgreceive` — Adaptive Server 会忽略此选项。

与 JMS 支持不同，使用 `endpoint` 不能指定用户名和口令。MQ 会检查相关 OS 登录名的权限。请参见第 33 页的“MQ 安全性”。

示例

示例 1 向队列管理器发送一条消息，通过启用 SSL 的 CH1 通道进行通信，密码成套程序为 `NULL_MD5`：

```
select msgsend('e',
  'ibm_mq:CH1(ssl:sslcip=NULL_MD5)/tcp/linuxxm11:1105?qmgr=MASTER_QM1,
  queue=Q2')
```

示例 2 向本地队列发送消息“hello world 1”，该队列在安装 MQ 后已在队列管理器中变为可用：

```
select msgsend('hello world 1',
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,
  queue=SYSTEM.DEFAULT.LOCAL.QUEUE')
```

示例 3 向某个队列发送消息“hello world 2”：

```
select msgsend('hello world 2',
  'ibm_mq:channel2/tcp/host2(5678)?qmgr=QM2,
  queue=SYSTEM.DEFAULT.QUEUE')
```

示例 4 向某个队列发送消息“hello world 3”：

```
select msgsend('hello world 3',
  'ibm_mq:channel2/tcp/host2(5678)?qmgr=QM2,
  remote_qmgr=QM3,queue=QM3.Q')
```

option_string

说明	指定 <i>option_string</i> 的一般语法和处理方式。各个选项在引用它们的函数中均有介绍。
语法	<pre>option_string ::= basic_character_expression option_string_value ::= option_and_value [[,] option_and_value] option_and_value ::= option_name = option_value option_name ::= simple_identifier option_value ::= simple_identifier quoted_string integer_literal float_literal byte_literal true false null</pre>
参数	<p><i>option_string</i> 描述要指定的选项的字符串。</p> <p><i>simple_identifier</i> 标识 <i>option</i> 的值的字符串。</p> <p><i>quoted_string</i> 使用关于嵌入式引号的常规 SQL 约定形成的字符串。</p> <p><i>integer_literal</i> 按常规 SQL 约定指定的文字。</p> <p><i>float_literal</i> 按常规 SQL 约定指定的文字。</p> <p><i>true</i> 布尔型文字。</p> <p><i>false</i> 布尔型文字。</p> <p><i>null</i> 空文字。</p> <p><i>byte_literal</i> 格式为 0xHH，其中每个 H 都是一个十六进制数字。</p>
用法	有关 <i>option_string</i> 的用法，请参见第 93 页的 msgsend 。

sizespec

说明 (仅限 MQ) 接受 *size* 并接受将以下语法作为大小规范的消息选项和属性值。

语法 `sizespec ::= integer_number [sizespec_units]`
`sizespec_units ::= { M | K }`

参数 *integer_number*
大小。

K 或 k
KB。

M 或 m
MB。

sizespec_units
单位为 MB (M)、KB (K) 或字节的大小规范。
如果未提供 *sizespec_units*，则缺省值为字节。

示例 **示例 1** 显示 100 MB 的大小规范：

```
-- Specify buffer length to be 100 megabytes
select msgrecv('ibm_mq:channel1/tcp/host1(5678)?'
+ 'qmgr=QM1,queue=SYSTEM.DEFAULT.LOCAL.QUEUE'
option 'bufferLength=100M')
```

示例 2 显示 300 K 的大小规范：

```
-- Specify buffer length to be 300 kilobytes
select msgrecv(
'ibm_mq:channel2/tcp/host2(5678)?qmgr=QM2,remote_qmgr=QM3,queue=QM3.Q'
option 'bufferLength=300K')
```

示例 3 (MQ) 显示 1 MB 的大小规范：

```
-- bufferLength specified as 1 megabyte
select msgrecv(
'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=DEFAULT.QUEUE'
option 'bufferLength=1M')
```

示例 4 (MQ) 显示 10 K 的大小规范：

```
-- bufferLength specified as 10K
select msgrecv(
'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=DEFAULT.QUEUE'
option 'bufferLength=10K')
```

timespec

说明 接受使用 `timespec` 函数的时间间隔并接受将以下语法作为 MQ 和 JMS 的时间规范的消息选项和属性值。

语法 `'timeout=timespec'`
`timespec ::= integer_number [timespec_units]`
`timespec_units ::= { dd | hh | mi | ss | ms }`

参数

- `dd`
天数。
- `hh`
时数。
- `mi`
分钟数。
- `ss`
秒数。
- `ms`
毫秒数。

`timespec_units`
毫秒数。如果未提供 `timespec_units`，则缺省值为毫秒数。

示例 **示例 1** 显示 100 天的时间规范：

```
-- timeout specified as 100 days
select msgrecv('ibm_mq:channel2/tcp/host2(5678)?'
+ 'qmgr=QM2,remote_qmgr=QM3,queue=QM3.Q'
option 'timeout=100dd')
```

示例 2 显示 300 分钟的时间规范：

```
-- timeout specified as 300 minutes
select msgrecv('ibm_mq:channel1/tcp/host1(5678)?'
+ 'qmgr=QM1,queue=SYSTEM.DEFAULT.LOCAL.QUEUE'
option 'timeout=300mi')
```

示例 3 显示 1,024 毫秒的时间规范：

```
-- timeout specified as 1,024 milliseconds
select msgrecv(
'ibm_mq:channel2/tcp/host2(5678)?'
+ 'qmgr=QM2,queue=SYSTEM.DEFAULT.LOCAL.QUEUE'
option 'timeout=1024ms')
```

示例 4 (MQ) 显示 30 秒的时间规范:

```
-- timeout specified as 30 seconds
select msgrecv(
  'ibm_mq:channel1/tcp/host1(5678)?qmgr=QM1,queue=DEFAULT.QUEUE'
  option 'timespec=30ss')
```

示例 5 (JMS) 显示 30 分钟的时间规范:

```
-- timeout specified as 30 minutes
select msgrecv(
  'tibco)_jms:tcp://localhost:7222?queue=queue.sample'
  option 'timeout=30mi')
```

另请参见

[msgconsume](#), [msgpublish](#), [msgrecv](#), [msgsend](#)

本章介绍了演示随 Active Messaging 选件分发的消息传送功能的示例代码。

主题	页码
Sybase 目录	133
使用 SQL 代码示例	134
使用 Java/JDBC 代码示例	134

Sybase 目录

SYBASE 目录包含以下三个子目录：

- *functionstring* — 一些脚本，用于生成 Replication Server 函数功能串，以便将缺省的 SQL 模板转换成对消息传送系统的调用。
- *sql* — 一些 SQL 脚本，其中包含了使用 Active Messaging 的示例。
- *jdbc* — 使用 Active Messaging 的一些 JDBC 示例。

可以在 `$SYBASE/$SYBASE_ASE/samples/messaging` 目录中找到这些代码示例。

每个子目录都包含一个 *README* 文件，该文件说明了每个代码示例的用途，提供了其运行步骤，并给出所有必要的安装操作说明。

Windows 与其它平台中的操作系统文件名并非完全相同。例如，Windows 平台中的 *queue_listener.bat* 在 UNIX/Linux 平台中可能仅仅为 *queue_listener*。

使用 SQL 代码示例

`$$SYBASE/$$SYBASE_ASE/samples/messaging/sql` 中的代码示例演示了如何编写或修改 SQL（存储过程、触发器等），以便将自定义的消息发布到消息传送系统。

这些示例还演示了如何利用 SQL 代码来使用来自消息总线的消息，在此过程中将 Adaptive Server 同时用作消息传送参与者和使用消息总线的应用程序。

使用 Java/JDBC 代码示例

`$$SYBASE/$$SYBASE_ASE/samples/messaging/jdbc` 中的代码示例说明了如何编写或修改 Java 代码，以便将自定义的消息发布到消息传送系统。

这些示例还演示了使用来自消息总线的消息的 Java 代码，在此过程中将 Adaptive Server 同时用作消息传送参与者和使用消息总线的应用程序。

词汇表

此处定义的和 JMS 和 MQ 相关的术语 — 在本文档中贯穿使用。

CipherSpec	鉴定完成后对 SSL 消息应用的 WebSphere MQ 加密算法和散列函数组合。
JMS	Java Message Service。
MQ	即 WebSphere MQ，是 IBM 提供的面向消息的中间件。
MQ 发布 / 预订	WebSphere MQ 的发布和预订功能。
MQI	WebSphere MQ 消息队列接口的编程 API。
MQM	管理队列的 WebSphere MQ 消息队列管理器进程。
RF 标头	MQ 发布 / 预订使用的 WebSphere MQ 规则和格式设置标头。发送至 MQ 发布 / 预订代理或流队列的所有消息都必须具有 RF 标头。RF 标头用于将控制信息传达给 MQ 发布 / 预订代理。在 MQ 发布 / 预订消息中，消息有效负载包含一个 RF 标头，标头后跟着应用程序数据。
RFH	即 WebSphere MQ 的规则和格式设置标头；指消息标头中提供该消息的规则和格式设置信息的部分。
SSL	用于通过未采取安全保护措施的网络以安全方式传输数据的一种行业标准协议。
本地队列管理器	应用程序连接到的 WebSphere MQ 队列管理器。
部分存储库	在 WebSphere MQ 中，指集群中查询完整存储库中的信息并构建自己的子集的队列管理器。
代理	在发布 / 预订模型中执行预订解析的一种 WebSphere MQ 进程。
对称算法	要求双方使用相同密钥的密码术算法。
队列	在 JMS 中，指用于点对点消息传送的域。 在 WebSphere MQ 中，指存储所发送的消息的对象。
发布	在 WebSphere MQ 中，指发布者发送的信息。

发布者	在 WebSphere MQ 中，指发布 / 预订模型中的发送方。
非对称算法	所用的加密密钥与解密密钥不相同的密码术算法。其中一个密钥必须保密，而另一个密钥则可以公开。
非永久性预订	仅在客户端处于连接状态时才保留消息的一种 JMS 预订。
服务提供程序	Tibco JMS 消息的提供程序。例如，Tibco JMS 就是一个服务提供程序，在本文档中称作消息传送提供程序。
公用密钥	非对称算法中可以公开的密钥。
共享密钥	由多种对称算法使用的相同密钥。
集群	由在逻辑上以某种方式相关的队列管理器构成的网络。
集群队列	由一个集群队列管理器承载并提供给集群中其它队列管理器使用的 WebSphere MQ 队列。
集群队列管理器	作为集群成员的 WebSphere MQ 队列管理器。
加密	将明文消息转换成密文的过程。
解密	将密文消息转换回其明文形式的过程。
流	在 WebSphere MQ 中，指由相关的 MQ 主题构成的组。
密码成套程序	SSL 连接使用的一组加密算法。
密码术	转换可读文本（称作明文）和不可读形式文本（称作密文）的过程。
密钥存储库	数字证书及其关联的私有密钥的存储区。
数字证书	用于防范仿冒。数字证书将公用密钥与其所有者绑定起来，而不考虑该所有者是个人、队列管理器还是某种其它实体。
私有密钥	非对称算法中必须保密的密钥。
通道	作为逻辑通信链路的一种 WebSphere MQ 对象。
完整存储库	保存着集群中每个队列管理器的全部相关信息的 WebSphere MQ 队列管理器。
消息传送客户端	产生或使用消息的 JMS 应用程序。
永久性预订	在客户端连接断开时会保留消息的一种 JMS 预订。
有效负载	WebSphere MQ 消息的正文。
预订	用于发布或使用一对多消息的 Tibco JMS 域。

预订者	在 WebSphere MQ 中，指发布 / 预订拓扑中的接收方。
远程队列管理器	在 WebSphere MQ 中，指与应用程序连接到的队列管理器不同的管理器。
主题	<p>在 Tibco JMS 中，与队列类似，不过是用于一对多消息传送。</p> <p>在 WebSphere MQ 中，指发布的主题。WebSphere MQ 发布 / 预订主题与 JMS 主题是不同的。在 JMS 中，主题是发布 / 预订结束点，而在 WebSphere MQ 中，发布 / 预订主题与 JMS 主题是不同的。</p>

索引

符号

- `$$SYBASE` 目录中的 `functionstring` 子目录 133
- `$$SYBASE` 目录中的 `jdbc` 子目录 133
- `$$SYBASE` 目录中的 `sql` 子目录 133
- @@. 请参见 全局变量。

英文

- ASE 消息类型 89
- ASE_MSBODY_SCHEMA 消息属性 46
- ASE_MSGBODY 消息属性 46
- ASE_ORIGIN 消息属性 46
- ASE_RTMS_CHARSET 消息属性 46
- ASE_RTMS_VERSION 消息属性 46
- ASE_SPID 消息属性 46
- ASE_TIMESTAMP 消息属性 46
- ASE_VERSION 消息属性 46
- ASE_VERSION_FORMATS 消息属性 46
- byte 消息类型 89
- endpoint 语法段 125–127
 - 参数 125
 - 描述的 125
 - 示例 127
 - 语法 125
- Java/JDBC, 使用代码示例 134
- JMS
 - msgrecv option_string 值 89
 - msgrecv property_option_clause 值 112
 - msgsend option_string 值 101
 - 定义的 135
 - 队列, 读取的消息 6
 - 队列说明 3
 - 消息属性 4
 - 消息选择器 4
 - 消息总线 2
- JMS 主题, 发布和使用消息 6
- message
 - 消息类型 89
- message header 关键字 47
- message property 关键字 47
- message selector 关键字 47
- MQ
 - MQ 客户端中的共享库 33
 - msgrecv option 和 option_string 值 84
 - msgrecv 函数, 用法 90
 - msgsend option_string 值 101
 - msgsend property_option_clause 值 104
 - RF 标头 4
 - rfhCommand 设置为 deletePublications 时的 msgsend 属性 113
 - 安全性与 33
 - 查询 35
 - 代理命令队列 3
 - 定义的 135
 - 发布和预订过程示例 13
 - 发布和预订示例 17
 - 发布和预订说明 13
 - 发布和预订消息传送模型 3
 - 发布者和预订者的标识 17
 - 概述 9
 - 授权 34
 - 所描述的本地队列对象 9
 - 所描述的别名队列对象 9
 - 所描述的集群队列对象 10
 - 所描述的模型队列对象 9
 - 所描述的消息标头 10
 - 所描述的消息正文 10
 - 所描述的远程队列对象 9
 - 消息传送模型 3
 - 消息队列接口 (MQI) 9
 - 消息分组 11
 - 消息类型 11

- 消息属性 4
- 在 Adaptive Server 主机上安装客户端 33
- 主题的语法 15
- MQ 队列管理器, 连接 33
- MQ 发布 / 预订过程 14
- MQ 客户端的共享库及其目录 33
- MQ 客户端中的共享库 33
- MQ 消息选择器 4
- MQ 中的代理命令队列 3
- MQ 中的发布者和预订者标识 17
- MQ 中主题的语法 15
- MQI (消息队列接口) 9
 - 定义的 135
 - 说明 9
- msgconsume** 函数 6, 68–70
 - 不支持的消息类型 70
 - 参数 68
 - 调用, 结果 70
 - 描述的 68
 - 示例 69
 - 用法 69
 - 语法 68
 - 支持的消息类型 70
- @@msgcorrelation** 消息传送全局变量 38
- msgheader** XML 文档 44–46
- @@msgheader** 消息传送全局变量 38
- @@msgid** 消息传送全局变量 40
- msgpropcount** 函数 7, 71
- msgproperties** XML 文档 44–46
 - 描述的 45
- @@msgproperties** 消息传送全局变量 41
- msgproplist** 函数 7, 72
- msgpropname** 函数 7, 73
- msgproptype** 函数 7, 74–75
- msgpropvalue** 函数 7, 76
- msgpublish** *option_string* 值 78
- msgpublish** *properties_clause* 值 79
- msgpublish** 函数 6, 77–79
- msgrecv** 函数 80–92
 - 参数 80
 - 权限 92
 - 用法 89
 - 语法 80
 - 针对 MQ 的用法 90
- @@msgreplyqmgr** 消息传送全局变量 41
- @@msgreplytoinfo** 消息传送全局变量 41
- @@msgschema** 消息传送全局变量 42
- msgsend** 函数 93–121
 - JMS *option_string* 值 101
 - JMS *property_option_clause* 值 112
 - MQ *option_string* 值 101
 - MQ *property_option_clause* 值 104
 - rfhCommand** 设置为 **deletePublications** 时的 **msgsend** 属性 113
 - 参数 93–96
 - 描述的 93
 - 权限 121
 - 示例 96–99
 - 事务中的行为 31
 - 用法 100–121
 - 语法 93
- @@msgstatus** 消息传送全局变量 43
- @@msgstatusinfo** 消息传送全局变量 43
- msgsubscribe** 函数 6, 122
- @@msgtimestamp** 消息传送全局变量 43
- msgunsubscribe** 函数 6, 123
- option_string** 语法段 128
- requeue**, 用于不正确的消息类型 89
- RF 标头 4
 - 定义的 135
- RFH, 定义的 135
- RFH. 请参见 RF 标头。
- rtrim** 函数 44
- set transactional messaging** 命令 32
- sizespec** 语法段 129
- sp_configure 'enable real time messaging'** 存储过程 49–50
- sp_engine** 存储过程 53–56
 - 参数 53
 - 描述的 53
 - 权限 56
 - 示例 54
 - 用法 55
 - 语法 53

sp_msgadmin 存储过程 6, 57–67

- 参数 62
- 描述的 57
- 权限 67
- 示例 65
- 用法 66
- 语法 57

sp_msgadmin 和 MQSeries 47
SQL

- 处理消息属性的函数 6
- 描述的函数 48
- 使用代码示例 134
- 事务中的命令 31

SSL, 定义的 135

text 消息类型 89

TIBCO EMS

- 消息总线 2

timespec 语法段 130

- 参数 130
- 描述的 130
- 示例 130
- 语法 130

transactional messaging full 关键字 47

transactional messaging none 关键字 47

transactional messaging simple 关键字 47

Transact-SQL, 发送消息 5

URL

- Apache log4j 记录服务 58
- IBM Publication Center 35
- Sun 的 Java 59

with remove 关键字 47

with retain 关键字 47

XML 文档

- msgheader* 44–46
- msgproperties* 44–46
- 示例 45
- 说明 44
- 用法 45

A

安全性与 MQ 33

B

本地队列对象 (MQ) 9

表

@@msgheader 全局变量字段和说明 38

channel_ciph 的有效 CipherSpec 名称 126

JMS **@@msgheader** 字段和说明 40

JMS 的 **msgrecv option_string** 值 89

MQ 客户端的共享库及其目录 33

MQ 主体 / 组及其授权 34

msgconsume option 和 *option_string* 参数值 68

msgpublish option_string 值 78

msgpublish properties_clause 值 79

msgrecv 的 MQ *option* 和 *option_string* 值 84

rfhCommand 设置为 **deletePublications** 时的 MQ

msgsend 属性 113

rfhCommand 设置为 **deregisterPublisher** 时的

msgsend 属性 113

rfhCommand 设置为 **deregisterSubscriber** 时的

msgsend 属性 114

rfhCommand 设置为 **publish** 时的 **send** 属性 115

rfhCommand 设置为 **registerPublisher** 时的 MQ

msgsend 属性 117

rfhCommand 设置为 **registerSubscriber** 时的 MQ

msgsend 属性 118

rfhCommand 设置为 **requestUpdate** 时的 MQ

msgsend 属性 119

RTDS 中的双重关键字和三重关键字 47

valid sp_msgadmin 担 **howi option option_string** 类型和值 63

适用于 JMS 的 **msgrecv property_option_clause** 值 112

适用于 JMS 的 **msgsend option_string** 值 101

适用于 MQ 的 **msgsend option_string** 值 101

适用于 MQ 的 **msgsend property_option_clause** 值 104

特定于 Adaptive Server 的消息属性 46

别名队列对象 (MQ) 9

C

参数

- endpoint** 语法段 125
- msgconsume** 函数 68
- msgpropcount** 函数 71
- msgrecv** 函数 80
- msgsend** 函数 93–96
- msgsubscribe** 函数 122
- msgunsubscribe** 函数 123
- option_string** 语法段 128
- sizespec** 语法段 129
- sp_configure 'enable real time messaging'** 存储过程 49
- sp_engine** 存储过程 53
- sp_msgadmin** 存储过程 62
- timespec** 语法段 130
- 创建队列与主题 47
- 存储过程
 - sp_configure 'enable real time messaging'** 49–50
 - sp_engine** 53–56
 - sp_msgadmin** 57–67
 - 列表 47
 - 为使用 **sp_msgadmin 'show'** 准备 WebSphere MQ 35

D

- 代理, 定义的 135
- 代码示例
 - 使用 Java/JDBC 134
 - 使用 SQL 134
- 点对点队列
 - JMS 3
 - MQ 3
- 点对点消息传送模型 2
 - JMS 3
 - MQSeries 3
- 定义的 CipherSpec 135
- 定义的 MQ 发布 / 预订 135
- 定义的 MQM 135
- 定义的本地队列管理器 135
- 定义的部分存储库 135
- 定义的非对称算法 136

- 定义的公用密钥 136
- 定义的集群 136
- 定义的集群队列 136
- 定义的集群队列管理器 136
- 定义的加密 136
- 定义的解密 136
- 定义的密码成套程序 136
- 定义的密码术 136
- 定义的密钥存储库 136
- 定义的数字证书 136
- 定义的私有密钥 136
- 定义的通道 136
- 定义的完整存储库 136
- 定义的消息传送客户端 136
- 定义的有效负载 136
- 定义的远程队列管理器 137
- 队列
 - 定义的 135
 - 发送和接收消息 5
 - 用于一对一消息传送 135
- 队列管理器, 连接 33
- 队列管理器说明 9
- 队列与主题, 创建, 删除, 访问 47
- 对称算法, 定义的 135

F

- 发布, 定义的 135
- 发布和预订
 - JMS 消息传送模型 3
 - MQ 消息传送模型 3
 - 描述的 13
 - 示例 17
 - 消息传送模型 2
- 发布者, 定义的 136
- 发送消息 5
- 非永久性预订 3, 136
- 服务提供程序 136

G

- 共享密钥, 定义的 136
- 关键字 47
 - message header** 47
 - message property** 47
 - message selector** 47
 - transactional messaging full** 47
 - transactional messaging none** 47
 - transactional messaging simple** 47
 - with remove** 47
 - with retain** 47
- 规则和格式设置标头。请参见 RF 标头。
- 国际化 30–31

H

- 函数
 - msgconsume** 68–70
 - msgpropcount** 71
 - msgproplist** 72
 - msgpropname** 73
 - msgproptype** 74–75
 - msgpropvalue** 76
 - msgpublish** 77–79
 - msgrecv** 80–92
 - msgsend** 93–121
 - msgsubscribe** 122
 - msgunsubscribe** 123
- 函数列表 48
- 列表 48
- 描述的 48
- 用于删除尾随空白的 **rtrim** 44

J

- 集群队列对象 (MQ) 10
- 接收消息 5

L

- 流, 定义的 136

M

- 模型, 消息传送 3
- 模型队列对象 (MQ) 9
- 目录
 - functionstring* 133
 - jdbc* 133
 - sql* 133

N

- 内置函数。请参见 函数, 包括各个函数名。

Q

- 权限
 - msgrecv** 函数 92
 - msgsend** 函数 121
 - sp_engine** 存储过程 56
 - sp_msgadmin** 存储过程 67
- 全局变量
 - @@msgcorrelation* 38
 - @@msgcreplymgr* 41
 - @@msgheader* 38
 - @@msgid* 40
 - @@msgmsgschema* 42
 - @@msgproperties* 41
 - @@msgreplytoinfo* 41
 - @@msgstatus* 43
 - @@msgstatusinfo* 43
 - @@msgtimestamp* 43
- 设置 38
- 示例 43, 44
- 用法 44

S

- 使用 JMS 主题中的消息 6
- 示例 133
 - endpoint** 语法段 127
 - MQ 发布和预订过程 13
 - msgconsume** 函数 69
 - msgheader* XML 文档 45
 - msgsend** 函数 96–99
 - msgsubscribe** 函数 122
 - msgunsubscribe** 函数 123
 - sp_configure 'enable real time messaging'** 存储过程 49
 - sp_engine** 存储过程 54
 - sp_msgadmin** 存储过程 65
 - Sybase 目录 133
 - timespec** 语法段 130
 - XML 文档 45
 - 概述 133
 - 全局变量 43, 44
 - 消息传送接口 7
- 事务
 - SQL 命令 31
 - 数据库, 对消息的影响 31
 - 提交 31
- 事务型行为, 使用 **set transactional messaging** 进行控制 32
- 数据类型, binary 值 90
- 数据类型的 binary 值 90
- 说明
 - endpoint** 语法段 125
 - MQ 9
 - MQ 本地队列对象 9
 - MQ 别名队列对象 9
 - MQ 集群队列对象 10
 - MQ 模型队列对象 9
 - MQ 消息标头 10
 - MQ 消息正文 10
 - MQ 远程队列对象 9
 - msgconsume** 函数 68
 - msgheader* XML 文档 44
 - msgproperties* XML 文档 45
 - msgsend** 函数 93

- msgsubscribe** 函数 122
- msgunsubscribe** 函数 123
- option_string** 语法段 128
- sizespec** 语法段 129
- sp_configure 'enable real time messaging'** 存储过程 49
- sp_engine** 存储过程 53
- sp_msgadmin** 存储过程 57
- timespec** 语法段 130
- XML 文档 44
- 代理 135
- 队列 135
- 队列管理器 9
- 非永久性预订 3
- 面向消息的中间件 1
- 通道 136
- 消息传送的一些概念 1
- 永久性预订 3
- 所描述的消息队列接口 (MQI) 9

T

- 特定于 Adaptive Server 的消息属性 46
- 提供程序, 消息传送 2

W

- 尾随空白, 使用 **rtrim** 删除 44
- 文本消息传送 26–27
- 文件名, 在不同的平台上是不同的 133

X

- 向 JMS 主题发布消息 6
- 消息
 - JMS 中的发送方 3
 - JMS 中的接收方 3
 - JMS 中的属性 4
 - JMS 中的选择器 4
 - MQ 中的分组 11

- MQ 中的类型 11
- MQ 中的属性 4
- MQ 中的选择器 4
- 标头 4
- 从 JMS 队列中读取 6
- 格式 4
- 接口, 预览 7
- 使用 Transact SQL 应用程序发送 5
- 属性, 使用 6
- 向队列发送和从队列中接收 5
- 向主题发布消息和使用其中的消息 6
- 正文 4
- 总线, TIBCO 2
- 消息标头 4
- 消息传送
 - 概念 1
 - 模型 3
- 消息传送的一些概念 1
- 消息传送模型
 - JMS 3
 - MQ 3
 - MQ 发布和预订 3
 - MQSeries 定义的 3
 - 点对点 2
 - 发布和预订 2
- 消息传送全局变量
 - @@msgcorrelation 38
 - @@msgheader 38
 - @@msgid 40
 - @@msgmsgschema 42
 - @@msgproperties 41
 - @@msgreplymgr 41
 - @@msgreplytoinfo 41
 - @@msgstatus 43
 - @@msgstatusinfo 43
 - @@msgtimestamp 43
 - char 数据类型 44
- 消息传送提供程序 2
 - 创建、删除和访问队列与主题 47
- 消息传送系统, 异步 2
- 消息格式 4
- 消息类型
 - 二进制 5
 - 文本 5
 - 在 `msgconsume` 中受支持 70

- 消息属性 6
 - 另请参见 `msgsend` 函数 6
 - ASE_MSBODY_SCHEMA 46
 - ASE_MSGBODY 46
 - ASE_ORIGIN 46
 - ASE_RTMS_CHARSET 46
 - ASE_RTMS_VERSION 46
 - ASE_SPID 46
 - ASE_TIMESTAMP 46
 - ASE_VERSION 46
 - ASE_VERSION_FORMATS 46
 - JMS, 中的 4
 - MQ, 中的 4
 - 特定于 Adaptive Server 46
- 消息选择器 4
- 选项字符串 48

Y

- 异步消息传送 2
- 永久性预订 3
 - 定义的 136
- 用法
 - `msgconsume` 函数 69
 - `msgrecv` 函数 89
 - `msgsend` 函数 100–121
 - `msgsend` 语法 93
 - `msgsubscribe` 函数 122
 - `msgunsubscribe` 函数 123
 - `option_string` 语法段 128
 - `sp_configure 'enable real time messaging'` 存储过程 50
 - `sp_engine` 存储过程 55
 - `sp_msgadmin` 存储过程 66
 - XML 文档 45
 - 全局变量 44
- 与消息相关的全局变量 38
- 语法
 - `endpoint` 语法段 125
 - `msgconsume` 函数 68
 - `msgpropcount` 函数 71
 - `msgrecv` 函数 80
 - `msgsubscribe` 函数 122

索引

- msgunsubscribe** 函数 123
- option_string** 语法段 128
- sizespec** 语法段 129
- sp_configure 'enable real time messaging'** 存储过程
49
- sp_engine** 存储过程 53
- sp_msgadmin** 存储过程 57
- timespec** 语法段 130
- 语法段
 - endpoint** 125–127
 - option_string** 128
 - sizespec** 129
 - timespec** 130
- 列表 48
- 预订
 - 定义的 136
 - 非永久性 3
 - 永久性 3
- 预订者, 定义的 137
- 预览, 示例 7
- 预览消息传送接口 7
- 远程队列对象 (MQ) 9

Z

- 在 Adaptive Server 主机上安装 MQ 客户端 33
- 执行所述的消息传送操作 2
- 主题
 - 定义的 137
 - 发布和使用消息 6
- 字节顺序 90