

# **TIBCO Enterprise Message Service™**

## **ユーザーガイド**

ソフトウェアリリース 7.0  
2012 年 8 月

## 重要情報

TIBCO ソフトウェアの中には、他の TIBCO ソフトウェアを埋め込みまたはバンドルしているものがあります。このような埋め込みまたはバンドルされた TIBCO ソフトウェアの使用は、ライセンス認可された TIBCO ソフトウェアの所定の機能を有効に（または制限付きのアドオン機能を提供）することのみに限定されます。埋め込みまたはバンドルされたソフトウェアは他のいかなる TIBCO ソフトウェアからの使用やアクセス、その他の目的での使用やアクセスはできません。

TIBCO ソフトウェアおよび本書の使用は、個別に執行されたソフトウェアライセンス契約か、または個別の契約がない場合にはソフトウェアのダウンロード時またはインストール時に表示されるエンドユーザーライセンス契約の条件に従うものとします。ライセンス契約は『ライセンスファイル』にも記載されています。あるいは、そのようなソフトウェアライセンス契約またはエンドユーザーライセンス契約がない場合は、ソフトウェアの『ライセンスファイル』に記載されているライセンスに従うものとします。本書の使用は上記ライセンス契約の条件に従うものとし、ユーザーは本書を使用することにより同契約の条件に同意しこれを受け入れるものとします。

本書に含まれる情報は、米国および国際的著作権法および協定が適用されます。TIBCO Software Inc. の書面による許可なく本書を複製することを禁じます。

TIBCO、The Power of Now、TIB、Information Bus、TIBCO Enterprise Message Service、TIBCO Rendezvous、TIBCO Enterprise、TIBCO SmartSockets、TIBCO ActiveMatrix BusinessWorks、および TIBCO Hawk は、アメリカ合衆国やその他の国における TIBCO Software Inc. の登録商標または商標です。

EJB、Java EE、J2EE、およびすべての Java ベースの商標およびロゴは、アメリカ合衆国やその他の国における Sun Microsystems, Inc. の登録商標または商標です。

本書に記載されているその他の製品名および社名はすべて各所有者に帰属する商標または登録商標で、本書では製品または企業を識別するためにのみ使用されています。

本ソフトウェアは、複数のオペレーティングシステムで利用できます。ただし、そのすべてのオペレーティングシステム用プラットフォームにおいて所定のバージョンが同時期にリリースされているとは限りません。特定のオペレーティングシステムのプラットフォームで本バージョンのソフトウェアが使用可能かどうかについては、Readme ファイルを参照してください。

本書は「現在提供されている状態」においてのみ開示されるものであり、権利の侵害、製品の市販性または利用の適切性を含め、明示的・暗示的に関係なく、いかなる保証も保持しません。

本書には、技術や印刷の誤りが含まれる場合があります。記載された情報は定期的に変更され、本書の改訂版に反映されます。TIBCO Software Inc. は、本書に記載された製品・プログラムを予告なしに改良・変更することがあります。

本書の内容は、このソフトウェアに付随しているほかのマニュアル（リリースノートや Readme ファイルを含む）によって、直接または間接的に、変更または改訂されることがあります。

Copyright © 1997-2012 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. 社外秘

# 目次

図 .....	xix
表 .....	xxi
序文 .....	xxv
このガイドの前のリリースからの変更点 .....	xxvi
関連マニュアル .....	xxvii
TIBCO Enterprise Message Service マニュアル .....	xxvii
その他の TIBCO 製品マニュアル .....	xxvii
サードパーティ製マニュアル .....	xxviii
表記規則 .....	xxix
TIBCO リソースとの接続 .....	xxxii
TIBCOCommunity への参加方法 .....	xxxii
TIBCO マニュアルへのアクセス方法 .....	xxxii
TIBCO サポートへの連絡方法 .....	xxxii
第 1 章 概要 .....	1
JMS の概要 .....	2
JMS メッセージモデル .....	3
ポイントツーポイント .....	3
パブリッシュとサブスクライブ .....	4
マルチキャスト .....	5
EMS のデスティネーション機能 .....	7
クライアント API .....	9
サンプルコード .....	9
TIBCO Rendezvous Java アプリケーション .....	9
管理 .....	10
サーバーの管理 .....	10
ユーザーとグループの管理 .....	11
TIBCO Hawk の使用方法 .....	11
セキュリティ .....	12
フォールトトレランス .....	12
ルーティング .....	13
サードパーティ製品との統合 .....	13

トランザクションのサポート .....	13
<b>第 2 章   メッセージ .....</b>	<b>15</b>
EMS の機能を拡張した JMS メッセージ .....	16
JMS メッセージの構造 .....	17
JMS メッセージのヘッダフィールド .....	17
EMS メッセージのプロパティ .....	19
JMS メッセージ本文 .....	21
メッセージの最大サイズ .....	22
メッセージの優先度 .....	23
メッセージ配信モード .....	24
PERSISTENT .....	24
NON_PERSISTENT .....	24
RELIABLE_DELIVERY .....	25
EMS による永続的メッセージの管理方法 .....	26
キューに送信される永続的メッセージ .....	26
トピックにパブリッシュされる永続的メッセージ .....	26
永続的メッセージおよび同期ファイルストレージ .....	27
複数ストアへのメッセージの保存 .....	29
ストアのタイプ .....	29
デフォルトのストアファイル .....	30
複数ストアの設定 .....	31
mstore の間隔について .....	32
メッセージの文字エンコーディング .....	35
サポートされる文字エンコーディング .....	36
メッセージの送信 .....	36
メッセージの圧縮 .....	37
メッセージの圧縮について .....	37
メッセージの圧縮の設定 .....	37
メッセージの確認応答 .....	38
NO_ACKNOWLEDGE .....	39
EXPLICIT_CLIENT_ACKNOWLEDGE .....	39
EXPLICIT_CLIENT_DUPS_OK_ACKNOWLEDGE .....	40
メッセージセレクト .....	41
データ型の変換 .....	44
同期および非同期のメッセージコンシューム .....	45
<b>第 3 章   デスティネーション .....</b>	<b>47</b>
デスティネーションの概要 .....	48
デスティネーション名 .....	49
静的デスティネーション .....	50

動的デスティネーション	50
テンポラリデスティネーション	50
デスティネーションブリッジ	51
デスティネーション名の構文	52
例	53
デスティネーションのプロパティ	55
channel	56
exclusive	56
expiration	57
export	58
flowControl	58
global	59
import	59
maxbytes	60
maxmsgs	61
maxRedelivery	61
overflowPolicy	62
prefetch	64
redeliveryDelay	66
secure	67
sender_name	68
sender_name_enforced	68
store	69
trace	70
デスティネーションの作成と変更	71
セキュアなデスティネーションの作成	72
ワイルドカード	73
ワイルドカード * と >	73
重複するワイルドカードと独立プロパティ	73
トピックでのワイルドカードの使用	74
キューでのワイルドカードの使用	74
ワイルドカードとマルチキャスト	74
ワイルドカードと動的に作成されたデスティネーション	75
継承	76
プロパティの継承	76
アクセス権限の継承	77
デスティネーションブリッジ	78
ブリッジの作成	80
アクセス制御とブリッジ	81
トランザクション	82
フロー制御	83
フロー制御の有効化	83
フロー制御の実行	83

マルチキャストとフロー制御 .....	84
ルートとフロー制御 .....	84
デスティネーションのブリッジとフロー制御 .....	85
フロー制御、スレッド、デッドロック .....	85
<b>第 4 章 基本操作 .....</b>	<b>87</b>
サンプルクライアントについて .....	88
サンプルクライアントのコンパイル .....	89
EMS 管理ツールを使用したユーザーの作成 .....	90
EMS サーバーと EMS 管理ツールの起動 .....	90
ユーザーの作成 .....	91
ポイントツーポイントメッセージングの例 .....	92
キューの作成 .....	92
セNDERクライアントとレシーバクライアントの起動 .....	92
パブリッシュとサブスクライブメッセージングの例 .....	93
トピックの作成 .....	93
サブスクライバクライアントの起動 .....	93
パブリッシュクライアントの起動とメッセージの送信 .....	94
セキュアトピックの作成 .....	95
持続サブスクライバの作成 .....	97
マルチキャストメッセージングの例 .....	98
EMS サーバーの停止 .....	98
EMS サーバーのマルチキャストの有効化 .....	98
マルチキャストチャネルの作成 .....	98
EMS サーバーの起動 .....	99
トピックのマルチキャストの有効化 .....	99
マルチキャスト デーモン の起動 .....	99
サブスクライバクライアントの起動 .....	99
パブリッシュクライアントの起動とメッセージの送信 .....	100
<b>第 5 章 EMS サーバーの実行 .....</b>	<b>101</b>
EMS サーバーの起動と停止 .....	102
デフォルト設定での EMS サーバーの起動 .....	102
JSON 設定を使った EMS サーバーの起動 .....	102
オプションを使用した EMS サーバーの起動 .....	103
EMS サーバーの停止 .....	105
Windows サービスとしての EMS サーバーの起動 .....	106
emsntsrg .....	106
エラーの回復ポリシー .....	109
セキュリティに関する注意事項 .....	110
セキュアな環境 .....	110
デスティネーションのセキュリティ .....	110

authorization パラメータ .....	110
Admin パスワード .....	111
接続セキュリティ .....	111
通信セキュリティ .....	112
認証データのソース .....	112
タイムスタンプ .....	112
パスワード .....	113
監査トレースログ .....	113
EMS による共有ストアファイルへのアクセスの管理方法 .....	114
パフォーマンス調整 .....	115
スループットを向上するためのスレッド関係の設定 .....	115
コア割り当ての決定 .....	115
ネットワーク I/O 接続 .....	116
その他の考慮事項 .....	116
<b>第 6 章 EMS 管理ツールの使用 .....</b>	<b>117</b>
EMS 管理ツールの起動 .....	118
tibemsadmin の最初の起動 .....	120
命名規則 .....	121
名前の長さの制限 .....	121
コマンドのリスト .....	122
add member .....	122
addprop factory .....	122
addprop queue .....	122
addprop route .....	123
addprop topic .....	123
autocommit .....	123
commit .....	123
compact .....	124
connect .....	124
create bridge .....	125
create durable .....	125
create factory .....	125
create group .....	125
create jndiname .....	125
create queue .....	126
create route .....	126
create rvcmlistener .....	126
create topic .....	127
create user .....	127
delete all .....	127
delete bridge .....	127
delete connection .....	128

delete durable . . . . .	128
delete factory . . . . .	128
delete group . . . . .	128
delete jndiname . . . . .	128
delete message . . . . .	128
delete queue . . . . .	129
delete route . . . . .	129
delete rvcmlistener . . . . .	129
delete topic . . . . .	129
delete user . . . . .	129
disconnect . . . . .	130
echo . . . . .	130
exit . . . . .	130
grant queue . . . . .	130
grant topic . . . . .	131
grant admin . . . . .	132
help . . . . .	132
info . . . . .	132
jaci clear . . . . .	132
jaci resetstats . . . . .	132
jaci showstats . . . . .	132
purge all queues . . . . .	133
purge all topics . . . . .	133
purge durable . . . . .	133
purge queue . . . . .	133
purge topic . . . . .	133
remove member . . . . .	133
removeprop factory . . . . .	134
removeprop queue . . . . .	134
removeprop route . . . . .	134
removeprop topic . . . . .	134
resume route . . . . .	134
revoke admin . . . . .	134
revoke queue . . . . .	135
revoke topic . . . . .	135
rotatelog . . . . .	136
set password . . . . .	136
set server . . . . .	137
setprop factory . . . . .	142
setprop queue . . . . .	142
setprop route . . . . .	143
setprop topic . . . . .	143
show bridge . . . . .	143
show bridges . . . . .	144
show channel . . . . .	144



show channels . . . . .	145
show config. . . . .	146
show consumer. . . . .	146
show consumers. . . . .	146
show connections. . . . .	149
show db . . . . .	152
show durable . . . . .	152
show durables. . . . .	153
show factory . . . . .	154
show factories. . . . .	154
show jndiname . . . . .	154
show jndinames . . . . .	155
show group . . . . .	155
show groups . . . . .	155
show members . . . . .	155
show message . . . . .	156
show messages . . . . .	156
show parents . . . . .	156
show queue . . . . .	157
show queues. . . . .	158
show route . . . . .	159
show routes . . . . .	160
show rvcmltransportledger . . . . .	160
show rvcmllisteners. . . . .	161
show server . . . . .	161
show stat. . . . .	161
show store . . . . .	162
show stores. . . . .	164
show topic. . . . .	164
show topics. . . . .	166
show transaction. . . . .	168
show transactions. . . . .	169
show transport . . . . .	170
show transports . . . . .	170
show user . . . . .	171
show users . . . . .	171
showacl admin . . . . .	171
showacl group. . . . .	171
showacl queue . . . . .	171
showacl topic . . . . .	172
showacl user. . . . .	172
shutdown . . . . .	172
suspend route. . . . .	172
time. . . . .	173
timeout . . . . .	173

transaction commit . . . . .	173
transaction rollback . . . . .	173
updatecrl . . . . .	173
whoami . . . . .	174
<b>第 7 章 設定ファイルの使用 . . . . .</b>	<b>175</b>
設定ファイルの場所 . . . . .	176
設定の仕組み . . . . .	176
tibemsd.conf . . . . .	177
グローバルなシステムのパラメータ . . . . .	189
ストレージファイルのパラメータ . . . . .	196
接続とメモリのパラメータ . . . . .	197
ネットワーク接続障害の検出パラメータ . . . . .	201
フォールトトレランスのパラメータ . . . . .	203
メッセージ追跡のパラメータ . . . . .	207
マルチキャストパラメータ . . . . .	207
TIBCO Rendezvous パラメータ . . . . .	209
TIBCO SmartSockets パラメータ . . . . .	209
ログファイルパラメータのトレース . . . . .	210
統計情報収集のパラメータ . . . . .	212
SSL サーバーのパラメータ . . . . .	214
LDAP パラメータ . . . . .	218
拡張可能なセキュリティパラメータ . . . . .	224
JVM パラメータ . . . . .	226
その他の設定ファイルの使用 . . . . .	228
acl.conf . . . . .	229
bridges.conf . . . . .	230
channels.conf . . . . .	231
durables.conf . . . . .	234
factories.conf . . . . .	235
groups.conf . . . . .	239
jaas.conf . . . . .	240
queues.conf . . . . .	240
routes.conf . . . . .	241
stores.conf . . . . .	243
tibrvcm.conf . . . . .	247
topics.conf . . . . .	247
transports.conf . . . . .	248
users.conf . . . . .	252
<b>第 8 章 認証とアクセス権限 . . . . .</b>	<b>255</b>
EMS のアクセス制御 . . . . .	256
管理者のアクセス権限 . . . . .	257

定義済みの管理者ユーザーとグループ .....	257
管理アクセス権限の付与と取り消し .....	258
管理者のアクセス権限の設定 .....	259
グローバルな管理者アクセス権限 .....	259
デスティネーションレベルのアクセス権限 .....	262
保護アクセス権限 .....	263
アクセス制御の有効化 .....	265
サーバー制御 .....	265
デスティネーション制御 .....	266
ユーザーとグループ .....	267
ユーザー .....	268
グループ .....	269
外部ディレクトリの設定 .....	269
ユーザーのアクセス権限 .....	273
ユーザーパラメータの設定の例 .....	274
ユーザーアクセス権限の継承 .....	274
ユーザーのアクセス権限の取り消し .....	275
アクセス権限の確認 .....	276
アクセス権限確認の例 .....	277
<b>第 9 章 拡張可能なセキュリティ .....</b>	<b>279</b>
拡張可能なセキュリティの概要 .....	280
拡張可能な認証 .....	282
拡張可能な認証の有効化 .....	282
認証モジュールの作成 .....	283
拡張可能なアクセス権限 .....	285
キャッシュされたアクセス権限 .....	285
アクセス権限を付与する方法 .....	286
アクセス権限にワイルドカードを使用した場合の影響 .....	288
拡張可能なアクセス権限の有効化 .....	289
アクセス権限モジュールの作成 .....	290
EMS サーバー内の JVM .....	292
JVM の有効化 .....	292
<b>第 10 章 データベースストアの使用 .....</b>	<b>293</b>
データベースストアの概要 .....	294
データベースストアの設定 .....	295
tibemsd.conf の設定 .....	296
stores.conf の設定 .....	297
Oracle RAC データベースの設定 .....	301
EMS Schema Export Tool .....	302

<b>第 11 章 EMS クライアントアプリケーションの開発</b>	<b>307</b>
JMS 仕様	308
JMS 1.1 仕様	308
JMS 1.0.2b 仕様	309
JMS 1.1 および JMS 1.0.2b インタフェースの概要	310
サンプルクライアント	311
プログラマー用チェックリスト	312
Java プログラマーのチェックリスト	312
C プログラマーのチェックリスト	313
C# プログラマーのチェックリスト	319
接続ファクトリ	323
接続ファクトリの照合	323
接続ファクトリの動的作成	323
接続の試行回数、タイムアウト、および遅延時間のパラメータ設定	324
EMS サーバーへの接続	326
接続の開始、停止、および切断	327
セッションの作成	328
例外リスナーの設定	329
トピックとキューの動的作成	331
メッセージプロデューサの作成	333
メッセージプロデューサの設定	334
メッセージコンシューマの作成	336
非同期でメッセージをコンシュームするメッセージリスナーの作成	337
メッセージの操作	341
メッセージの作成	341
メッセージプロパティの設定と取得	342
メッセージの送信	343
メッセージの受信	344
<b>第 12 章 JNDI を実装した EMS の使用</b>	<b>347</b>
EMS での管理対象オブジェクトの作成と変更	348
セキュア接続のための接続ファクトリの作成	348
フォールトトレランス接続のための接続ファクトリの作成	349
EMS に保存されている管理対象オブジェクトの照合	350
完全 URL 名によるオブジェクトの検索	352
セキュア照合の実行	353
フォールトトレランス照合の実行	354
<b>第 13 章 マルチキャストの使用方法</b>	<b>357</b>
マルチキャストの概要	358

マルチキャストの使用目的 .....	359
要件 .....	361
マルチキャストの設定 .....	362
マルチキャストの動的な設定 .....	362
マルチキャスト デーモン の設定 .....	363
マルチキャスト対応トピックへのアクセス制御 .....	366
マルチキャストの実行 .....	367
マルチキャスト デーモン の起動 .....	367
マルチキャストコンシューマの作成 .....	367
監視と統計情報 .....	368
監視 .....	368
統計情報 .....	368
<b>第 14 章   マルチキャストの実装とトラブルシューティング .....</b>	<b>369</b>
実装に関する注意事項 .....	370
接続 .....	370
マルチキャストトラフィックの制限 .....	372
帯域幅の管理 .....	372
マルチキャスト実装の概略 .....	376
ステップ 1 : マルチキャストネットワークアーキテクチャの設計 .....	376
ステップ 2 : EMS のインストールおよびセットアップ .....	378
ステップ 3 : ネットワーク機能およびアプリケーション機能の決定 .....	381
EMS のマルチキャストのトラブルシューティング .....	385
トラブルシューティングのヒント .....	385
アプリケーションおよびマルチキャスト デーモン のエラーおよび警告 .....	387
サーバーエラー .....	388
<b>第 15 章   TIBCO Rendezvous との相互運用 .....</b>	<b>389</b>
概要 .....	390
メッセージの変換 .....	390
設定 .....	390
Rendezvous のためのトランスポートの設定 .....	392
トランスポート定義 .....	393
トピック .....	398
サブスクライバが存在する場合のみのインポート .....	398
ワイルドカード .....	398
保証メッセージ .....	399
キュー .....	400
設定 .....	400
インポート : 開始および停止 .....	400
ワイルドカード .....	400

インポートに関連する問題.....	402
フィールド ID .....	402
JMSDestination .....	402
JMSReplyTo.....	402
JMSExpiration .....	402
JMSTimestamp .....	403
保証配信 .....	403
エクスポートに関連する問題.....	404
JMSReplyTo.....	404
保証メッセージ.....	404
保証配信 .....	404
メッセージの変換.....	405
JMS ヘッダフィールド.....	405
JMS プロパティフィールド.....	406
メッセージ本文.....	407
データ型 .....	409
Pure Java Rendezvous プログラム .....	411
<b>第 16 章    TIBCO SmartSockets との相互運用 .....</b>	<b>413</b>
概要.....	414
メッセージの変換 .....	414
設定 .....	414
サーバーの起動 .....	415
SmartSockets のためのトランスポートの設定 .....	415
トランスポート定義 .....	416
デスティネーション名：構文とセマンティクス .....	420
トピック .....	421
サブスクライバが存在する場合のみのインポート .....	421
ワイルドカード .....	422
キュー.....	422
設定 .....	422
インポート：開始および停止 .....	422
ワイルドカード .....	423
インポートに関連する問題.....	424
インポートしたデスティネーション名は固有の名前である必要がある .....	424
JMSReplyTo.....	424
保証配信 .....	424
エクスポートに関連する問題 .....	425
JMSReplyTo.....	425
ワイルドカードサブスクリプション .....	425
保証配信 .....	425
メッセージの変換.....	426

JMS ヘッダフィールド	426
JMS プロパティフィールド	426
SmartSockets メッセージのプロパティ	427
メッセージ本文	428
データ型	430
デスティネーション名	431
<b>第 17 章 サーバーアクティビティの監視</b>	<b>433</b>
ログファイルとトレース	434
ログファイルの設定	434
サーバー上でのトレース	435
メッセージのトレース	440
デスティネーションのメッセージトレースの有効化	440
メッセージのメッセージトレースの有効化	441
サーバーイベントの監視	442
システム監視トピック	442
メッセージの監視	442
監視トピックの表示	445
監視トピックのパフォーマンスへの影響	446
サーバーの統計情報の操作	448
サーバーの全般統計情報	448
統計情報収集の有効化	449
統計情報の表示	451
<b>第 18 章 SSL プロトコルの使用</b>	<b>453</b>
TIBCO Enterprise Message Service での SSL のサポート	454
デジタル証明書	455
デジタル証明書形式	456
秘密鍵形式	456
証明書と鍵のファイル名	457
サーバーでの SSL の設定	459
SSL パラメータ	459
コマンドラインオプション	459
EMS クライアントでの SSL の設定	460
クライアントデジタル証明書	460
SSL の設定	461
暗号スイートの指定方法	464
暗号スイートの構文	464
サポートされる暗号スイート	466
認証にのみ SSL を使用する	470
FIPS 準拠の有効化	471

EMS サーバーの有効化 .....	471
EMS クライアントの有効化 .....	472
<b>第 19 章    フォールトトレランス .....</b>	<b>473</b>
フォールトトレランスの概要 .....	474
共有ステート .....	474
非共有ステートフェイルオーバー .....	475
設定ファイル .....	475
共有ステートフェイルオーバーのプロセス .....	476
検出 .....	476
応答 .....	476
役割の交代 .....	477
クライアントの移管 .....	477
メッセージの再配信 .....	478
ハートビートパラメータ .....	479
非共有ステートフェイルオーバーのプロセス .....	480
デュアルステートフェイルオーバー .....	481
共有ステート .....	483
共有ステートの実装 .....	483
共有ステートに保存されたメッセージ .....	485
ストレージファイル .....	485
ストレージパラメータ .....	486
フォールトトレランスサーバーの設定 .....	487
共有ステート .....	487
非共有ステート .....	489
Central Administration でのフォールトトレランスの設定 .....	490
フォールトトレランスサーバー用のクライアントの設定 .....	492
三つ以上の URL の指定 .....	492
再接続障害パラメータの設定 .....	493
非共有ステート接続用のクライアントの設定 .....	495
非共有ステートライブラリのインクルード .....	495
非共有ステート接続ファクトリの作成 .....	495
サーバーの URL の指定 .....	496
<b>第 20 章    ルートの使用 .....</b>	<b>497</b>
ルーティングの概要 .....	498
ルート .....	499
基本操作 .....	499
グローバルデスティネーション .....	500
一意のルーティングパス .....	500
ゾーン .....	502



基本操作 .....	502
1 ホップゾーンの冗長パスの削除 .....	502
複数のゾーンのオーバーラップ .....	503
アクティブルートとパッシブルート .....	505
ルートとゾーンの設定 .....	506
フォールトトレランスサーバーへのルーティング .....	507
ルーティングと SSL .....	507
トピックメッセージのルーティング .....	511
登録されている対象サブスクリプションの伝播 .....	511
トピックメッセージのルーティングに使用するセクタ .....	513
ルーティング先キュー .....	516
ルーティングと認可 .....	519
 <b>付録 A メッセージの監視 .....</b>	<b>521</b>
監視トピックの説明 .....	522
トピックメッセージプロパティの説明 .....	525
 <b>付録 B エラーメッセージとステータスメッセージ .....</b>	<b>535</b>
エラーメッセージとステータスメッセージ .....	536
 <b>索引 .....</b>	<b>629</b>





図 1	メッセージの配信	3
図 2	ポイントツーポイントメッセージ	4
図 3	メッセージのパブリッシュとサブスクライブ	5
図 4	マルチキャストメッセージ	6
図 5	永続的メッセージ配信	24
図 6	非永続的メッセージ配信	25
図 7	信頼メッセージ配信	25
図 8	キューに送信される永続的メッセージ	26
図 9	トピックにパブリッシュされる永続的メッセージ	27
図 10	メッセージ配信と確認応答	38
図 11	ひとつのトピックからひとつのキューへのブリッジ	79
図 12	ひとつのトピックから複数のデスティネーションへのブリッジ	79
図 13	ひとつのキューから複数のデスティネーションへのブリッジ	80
図 14	ふたつのスレッドに発生したフロー制御によるデッドロック	86
図 15	ユーザー、グループ、およびアクセス権限	267
図 16	ユーザーの認証およびアクセス権限の確認方式	281
図 17	アクセス権限のデシジョンツリー	287
図 18	JMS 1.1 プログラミングモデル	308
図 19	JMS 1.0.2b プログラミングモデル	309
図 20	マルチキャストメッセージのコンシューマの作成	358
図 21	マルチキャストの利点	360
図 22	サンプルのマルチキャスト実装アーキテクチャ	377
図 23	EMS サーバー内の Rendezvous トランスポート	390
図 24	EMS サーバー内の SmartSockets トランスポート	414
図 25	共有状態を使用したプライマリサーバーとバックアップサーバー	474
図 26	非共有状態を使用した現在のサーバーと 2 番目のサーバー	475
図 27	プライマリサーバーでの障害の発生	476
図 28	回復したサーバーのバックアップサーバーへの移行	477

図 29	非共有ステートフェイルオーバー .....	480
図 30	デュアルステートフェイルオーバーのプロセス.....	481
図 31	ルート：双方向に通信し合うデスティネーション.....	499
図 32	ルート：グローバルデスティネーション .....	500
図 33	ルート：一意のパス .....	501
図 34	ゾーン：マルチホップ .....	502
図 35	ゾーン：1ホップ .....	503
図 36	ゾーン：オーバーラップ.....	504
図 37	ルーティング：サブスクライバの伝播 .....	511
図 38	ルーティング：トピックセレクトアの例 .....	513
図 39	ルーティング：キュー .....	516
図 40	ルーティング：認可 .....	519

# 表

表 1	全般的な表記規則	xxix
表 2	構文で使用する表記規則	xxxi
表 3	管理機能の概要	10
表 4	JMS メッセージのヘッダ	17
表 5	メッセージプロパティの概要	19
表 6	JMS メッセージタイプ	22
表 7	データ型の変換	44
表 8	デスティネーションの概要	48
表 9	デスティネーション名で特別な意味を持つ文字	53
表 10	有効なデスティネーション名の例	53
表 11	無効なデスティネーション名の例	54
表 12	デスティネーションのプロパティ	55
表 13	Prefetch	64
表 14	tibemsd オプション	104
表 15	tibemsdadmin のオプション	118
表 16	Set server のパラメータ	137
表 17	出力フィールドの説明	144
表 18	出力フィールドの説明	147
表 19	show connections コマンドの type パラメータ	149
表 20	出力フィールドの説明	150
表 21	show durable テーブルの情報	152
表 22	show durables テーブルの情報	154
表 23	show queue テーブルの情報	157
表 24	show queues テーブルの情報	158
表 25	show routes テーブルの情報	160
表 26	show store テーブルの情報	162
表 27	show topic テーブルの情報	164
表 28	show topics テーブルの情報	166

表 29	show transactions テーブルの情報	168
表 30	show transactions テーブルの情報	170
表 31	tibemsd.conf パラメータ	177
表 32	設定ファイル	228
表 33	ACL パラメータ	229
表 34	ブリッジのパラメータ	230
表 35	チャンネルのパラメータ	231
表 36	持続サブスクライバのパラメータ	234
表 37	接続ファクトリのパラメータ	235
表 38	グループのパラメータ	239
表 39	ルートのパラメータ	241
表 40	ストアファイルのパラメータ	243
表 41	RVCM リスナーのパラメータ	247
表 42	トランスポートのパラメータ	248
表 43	ユーザーのパラメータ	252
表 44	グローバルな管理者アクセス権限	259
表 45	デスティネーションレベルの管理アクセス権限	262
表 46	よく利用される LDAP サーバーのデフォルト設定	271
表 47	キューのアクセス権限	273
表 48	トピックのアクセス権限	274
表 49	データベースストアファイルのパラメータ	298
表 50	EMS Schema Export Tool のオプション	303
表 51	JMS API オブジェクトの概要	310
表 52	32 ビット UNIX のリンカフラグ	315
表 53	64 ビット UNIX のリンカフラグ	315
表 54	Microsoft Windows のダイナミックライブラリファイル	316
表 55	Microsoft Windows の静的ライブラリファイル	317
表 56	OpenVMS の共有可能イメージライブラリファイル	318
表 57	OpenVMS の静的ライブラリファイル	318
表 58	EMS アセンブリ DLL	319
表 59	EMS ポリシーファイル	320
表 60	.NET 機能のサポート	321

表 61	tibemsmcd オプション .....	364
表 62	Rendezvous : トランスポートのパラメータ .....	393
表 63	Rendezvous : JMS ヘッダフィールドと RV データ型との対応関係 .....	405
表 64	Rendezvous の対応するメッセージのプロパティ .....	406
表 65	Rendezvous : 対応するメッセージタイプ (インポート) .....	407
表 66	Rendezvous : 対応するメッセージタイプ (エクスポート) .....	408
表 67	Rendezvous : データ型の対応関係 .....	409
表 68	SmartSockets : トランスポートのパラメータ .....	416
表 69	SmartSockets : 対応するメッセージプロパティ (インポートとエクスポート) .....	427
表 70	SmartSockets : 対応するメッセージタイプ (エクスポート) .....	429
表 71	SmartSockets : データ型の対応関係 .....	430
表 72	サーバーのトレースオプション .....	436
表 73	メッセージ監視修飾子 .....	443
表 74	ファイル形式 .....	457
表 75	ConnectionFactory SSL パラメータ .....	462
表 76	Java クライアントの暗号スイートの修飾子 .....	464
表 77	暗号スイートの OpenSSL の修飾子 .....	465
表 78	サポートされる暗号スイート .....	466
表 79	フォールトトレランスの共有ストレージ条件 .....	483
表 80	ルートの SSL のパラメータ .....	508
表 81	監視トピック .....	522
表 82	メッセージのプロパティ .....	525
表 83	イベントアクションのプロパティ値 .....	530
表 84	イベント発生理由のプロパティ値 .....	532





# 序文

TIBCO は TIBCO Enterprise Message Service™ の最新リリースを発表しました。このリリースは、Information Bus® の機能を活用して、真にイベント駆動型の IT 環境を実現する歴代の TIBCO 製品の中の最新バージョンです。TIBCO Enterprise Message Service と他の TIBCO 製品に TIB® テクノロジーがどのように装備されているかについての詳細は、[www.tibco.com](http://www.tibco.com) を参照してください。

TIBCO Enterprise Message Service ソフトウェアを使用することで、アプリケーションプログラムは Java メッセージサービス (JMS: Java Message Service) プロトコルに従ってメッセージを送受信できるようになります。また、メッセージ製品の TIBCO Rendezvous と TIBCO SmartSockets も統合できます。

## トピック

---

- 「このガイドの前のリリースからの変更点」、xxvi ページ
- 「関連マニュアル」、xxvii ページ
- 「表記規則」、xxix ページ
- 「TIBCO リソースとの接続」、xxxii ページ

## このガイドの前のリリースからの変更点

---

このセクションでは、このガイドの前のリリースからの主な変更点を列挙します。

### 機能拡張

次の機能拡張が、以下に示すセクションで説明されています。

- **マルチコアマシンでの EMS サーバーパフォーマンスの向上**：このリリースでは、ふたつの新規パラメータが導入されました。それらを使用して、マルチコアマシンでのパフォーマンスを詳細に調整できるようになります。
  - ネットワーク I/O スレッドを特定のプロセッサにバインドするには、[processor\\_ids](#) パラメータを `tibemsd.conf` に設定します。
  - ストレージスレッドを特定のプロセッサにバインドするには、[processor\\_id](#) を `stores.conf` ファイルに設定します。詳細は、[パフォーマンス調整](#) ページ 115 を参照してください。
- **未確認メッセージを持つコンシューマへの対処**：[disconnect\\_non\\_acking\\_consumers](#) パラメータを使用すると、`maxbytes` または `maxmsgs` の制限を超える、保留中のメッセージを持つコンシューマをサーバーが特定できるようになります。制限に到達して、コンシューマがそのメッセージを確認応答しなかった場合、サーバーは、コンシューマに送信されたメッセージを破棄し、コンシューマの接続を切断します。これにより、メッセージに確認応答せずにメッセージを消費するアプリケーションからサーバーが保護されます。詳細は、[disconnect\\_non\\_acking\\_consumers](#) ページ 190 を参照してください。

### 機能の変更点

次の機能の動作が変更されました。

- **JSON 設定ファイル**：TIBCO Enterprise Message Service のこのリリースでは、EMS サーバーに対する新しい設定オプションが導入されました。EMS サーバー設定は、単一の JSON ベースの設定ファイル `tibemsd.json` に保存できます。このファイルには、サーバーの設定全体が含まれています。サブファイルを作成する必要はありません。また、単一の JSON 設定ファイルにフォールトトレランスサーバーのペアの設定が保持されます。この設定オプションは、Central Administration の機能を有効にするために使用されます。

JSON 設定ファイルを使用して EMS サーバーを開始する方法については、[JSON 設定を使った EMS サーバーの起動](#) ページ 102 を参照してください。Central Administration についての詳細は、TIBCO Enterprise Message Service Central Administration のガイドを参照してください。

## 関連マニュアル

---

このセクションでは、参照できるマニュアルを紹介します。

### TIBCO Enterprise Message Service マニュアル

TIBCO Enterprise Message Service マニュアルセットには、以下のマニュアルが含まれています。

- 『TIBCO Enterprise Message Service ユーザーガイド』(本書): 製品、機能、設定について全般的に理解するには、このマニュアルをお読みください。
- 『TIBCO Enterprise Message Service Central Administration』: Central Administration インタフェースについての詳細は、このマニュアルを参照してください。
- 『TIBCO Enterprise Message Service のインストール』: この製品をインストールする前にこのマニュアルの該当セクションをお読みください。
- 『TIBCO Enterprise Message Service C & COBOL API リファレンス』: C API リファレンスを HTML 形式と PDF 形式で参照できます。
- 『TIBCO Enterprise Message Service Java API リファレンス』: Java API リファレンスには、HTML 文書インタフェースを介してのみアクセスできます。
- 『TIBCO Enterprise Message Service .NET API リファレンス』: .NET API リファレンスには、HTML 文書インタフェースを介してのみアクセスできます。
- 『TIBCO Enterprise Message Service リリースノート』: 新機能と変更された機能のリストについては、リリースノートを参照してください。また、このリリースにおいて判明している問題と解決済みの問題のリストも掲載しています。この文書は、PDF 形式でのみ参照できます。

### その他の TIBCO 製品マニュアル

以下の TIBCO 製品のマニュアルをお役立てください。

- TIBCO Rendezvous<sup>®</sup>
- TIBCO SmartSockets<sup>®</sup>
- TIBCO Hawk<sup>®</sup>
- TIBCO EMS<sup>®</sup> Client for z/OS (CICS)
- TIBCO EMS<sup>®</sup> Client for z/OS (MVS)
- TIBCO EMS<sup>®</sup> Client for i5/OS

## サードパーティ製マニュアル

- Java™ Message Service の仕様  
(<http://java.sun.com/products/jms/index.html> を参照)。
- 『Java™ Message Service』 著者 : Richard Monson-Haefel、David A. Chappell。  
出版社 : O'Reilly and Associates (カリフォルニア州セバストポリ)、2001 年  
発行。
- 『Java™ Authentication and Authorization Service (JAAS) LoginModule  
Developer's Guide』 および 『Reference Guide』  
(<http://java.sun.com/products/jaas/> を参照)。

## 表記規則

本書では、以下の表記規則を使用します。

表 1 全般的な表記規則

種類	説明
<i>TIBCO_HOME</i> <i>ENV_HOME</i> <i>EMS_HOME</i>	<p>多数の TIBCO 製品が同じホームディレクトリ内にインストールされる必要があります。このディレクトリは、文書内で <i>TIBCO_HOME</i> として参照されます。<i>TIBCO_HOME</i> の値は、オペレーティングシステムによって異なります。たとえば、Windows システムの場合、デフォルト値は <code>C:\¥tibco</code> です。</p> <p>他の TIBCO 製品はインストール環境にインストールされます。互換性のない製品や、同じ製品の複数のインスタンスは、別のインストール環境にインストールされます。そのような製品がインストールされるディレクトリは、文書内で <i>ENV_HOME</i> として参照されます。<i>ENV_HOME</i> の値は、オペレーティングシステムによって異なります。たとえば、Windows システムの場合、デフォルト値は <code>C:\¥tibco</code> です。</p> <p>TIBCO Enterprise Message Service は <i>TIBCO_HOME</i> 内の任意のディレクトリにインストールされます。このディレクトリは、文書内で <i>EMS_HOME</i> として参照されます。<i>EMS_HOME</i> の値は、オペレーティングシステムによって異なります。たとえば、Windows システムの場合、デフォルト値は <code>C:\¥tibco¥ems¥7.0.</code> です。</p>
code font	<p>コードフォントは、コマンド、コード例、ファイル名、パス名、コマンドウィンドウに表示される出力を表します。例：</p> <p>TIBCO foo プロセスの開始には、MyCommand を使用します。</p>
bold code font	<p>太字のコードフォントは、以下の場合に使用します。</p> <ul style="list-style-type: none"> <li>手順で、ユーザー・タイプを示すときに使用します。例：ユーザー名 <b>admin</b> を入力します。</li> <li>長いコード例で、その特定部分を示すに使用します。</li> <li>コマンド構文では、デフォルト値を示すときに使用します。たとえば以下は、パラメータが指定されていない場合に MyCommand が有効になることを示します。</li> </ul> <p>MyCommand [<b>enable</b>   disable]</p>

表 1 全般的な表記規則

種類	説明
<i>italic font</i>	<p>斜体フォントは、以下の場合に使用します。</p> <ul style="list-style-type: none"><li>文書のタイトルを示すときに使用します。例：詳細は、『TIBCO BusinessWorks 概要』を参照してください。</li><li>新規用語を示します。例：ポータルページには、いくつかの「ポートレット」を含めることができます。ポートレットは、ポータルで動作する小型アプリケーションです。</li><li>コマンドまたはコード構文で置き換える必要がある変数を示します。例： MyCommand <i>pathname</i></li></ul>
キーの組み合わせ	<p>プラス記号で区切られたキー名は、同時に押すキーを示します。例：Ctrl+C</p> <p>コンマと半角スペースで区切られたキー名は、順番に押すキーを示します。例：Esc, Ctrl+Q</p>

表 2 構文で使用する表記規則

種類	説明
[ ]	<p>コマンドまたはコマンド構文でのオプションアイテム。</p> <p>例：</p> <pre>MyCommand [optional_parameter] required_parameter</pre>
	<p>複数のアイテムからひとつを選択するときにそれを区切るための OR 演算子。</p> <p>たとえば、以下のパラメータのひとつだけを選択できます。</p> <pre>MyCommand para1   param2   param3</pre>
{ }	<p>アイテムの論理グループ。各論理グループ内で、使用方法は異なります。</p> <p>たとえば、以下のコマンドではふたつのパラメータを必要とし、param1 と param2 か param3 と param4 です。</p> <pre>MyCommand {param1 param2}   {param3 param4}</pre> <p>次の例では、コマンドはふたつのパラメータを必要とします。最初のパラメータは param1 か param2、ふたつ目のパラメータは param3 か param4 です。</p> <pre>MyCommand {param1   param2} {param3   param4}</pre> <p>次の例では、コマンドでふたつのパラメータか三つのパラメータを使用できます。1 番目のパラメータは param1 である必要があります。オプションで、2 番目のパラメータとして param2 を使用できます。最後の 3 番目のパラメータは param3 か param4 のいずれかになります。</p> <pre>MyCommand param1 [param2] {param3   param4}</pre>

## TIBCO リソースとの接続

---

### TIBCOCommunity への参加方法

TIBCOCommunity は、TIBCO の顧客、パートナー、および常駐エキスパート用のウェブサイトです。ここで、TIBCO コミュニティの集約的なエクスペリエンスを共有し、アクセスできます。TIBCOCommunity では、フォーラム、ブログ、およびさまざまなリソースへのアクセスが提供されています。登録するには、<http://www.tibcommunity.com> にアクセスしてください。

### TIBCO マニュアルへのアクセス方法

ここで、TIBCO のマニュアルにアクセスできます。

<http://docs.tibco.com>

### TIBCO サポートへの連絡方法

このマニュアルまたは記載されているソフトウェアに対するご意見や質問は、下記アドレスの TIBCO サポートまでご連絡ください。

- TIBCO サポートの概要と、TIBCO サポートのご使用方法については、次のサイトを参照してください。

<http://www.tibco.com/services/support>

- すでに有効期限内の保守契約またはサポート契約を結ばれている場合は、次のサイトを参照してください。

<https://support.tibco.com>

このサイトにアクセスするには、ユーザー名およびパスワードが必要です。ユーザー名をお持ちでない場合はお問い合わせください。なお、上記のサイトのいずれも現在のところ英文のみとなっております。



# 第 1 章 概要

この章では、JMS (Java Message Service) および TIBCO Enterprise Message Service の基本概念についての概要を説明します。

## トピック

---

- 「[JMS の概要](#)」、2 ページ
- 「[JMS メッセージモデル](#)」、3 ページ
- 「[EMS のデスティネーション機能](#)」、7 ページ
- 「[クライアント API](#)」、9 ページ
- 「[管理](#)」、10 ページ
- 「[セキュリティ](#)」、12 ページ
- 「[フォールトトレランス](#)」、12 ページ
- 「[ルーティング](#)」、13 ページ
- 「[サードパーティ製品との統合](#)」、13 ページ

## JMS の概要

---

JMS 1.1 は、アプリケーション間のメッセージングに関する Java フレームワーク仕様です。この仕様は、エンタープライズアプリケーション間で使用可能な統一したメッセージングインタフェースを提供するために、TIBCO をはじめとする各社の協力の下、Sun Microsystems 社によって開発されました。

メッセージサービスを使用すると、企業内のアプリケーションを統合することができます。たとえば企業内で、あるアプリケーションを顧客管理用に、別のアプリケーションを製品のインベントリ用に、他のアプリケーションを原材料の追跡用にというように、用途別に複数のアプリケーションを使用しているとします。どのアプリケーションも企業の運営にとって重要なものですが、より重要なことは、アプリケーション間でデータを交換し、ビジネスプロセスの流れをスムーズにすることです。MOM (Message-oriented-middleware: メッセージ指向ミドルウェア) は、これらのアプリケーション間の共通通信プロトコルを作成し、企業内のコンピュータ環境にある新しいアプリケーションと既存のアプリケーションを簡単に統合することができます。

JMS フレームワーク (実装ではなくインタフェース仕様) は、MOM の開発用の基盤を提供するように設計されています。TIBCO Enterprise Message Service は JMS を実装し、これにより TIBCO Rendezvous や TIBCO SmartSockets などの他のメッセージサービスとの接続を可能にしています。この章では、JMS および TIBCO Enterprise Message Service での JMS 実装の基本概念を説明します。JMS の要件と機能についての詳細は、以下の資料を参照してください。

- JMS (Java Message Service) の仕様  
(<http://java.sun.com/products/jms/index.html> を参照)。
- 『Java Message Service』著者: Richard Monson-Haefel, David A. Chappell。  
出版社: O'Reilly and Associates (カリフォルニア州セバストポリ)、2001 年発行。

### JMS への準拠

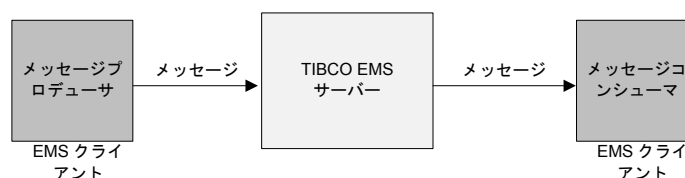
TIBCO Enterprise Message Service 7.0 は、Sun Microsystem 社が開発した Java Message Service 1.1 (JMS 1.1) 用の Technology Compatibility Kit (TCK) のテストに合格しています。つまり、EMS 7.0 は、JMS 1.1 仕様に準拠しているということです。

## JMS メッセージモデル

JMS は、メッセージの作成と配信に基づいています。メッセージとは、あるアプリケーションが別のアプリケーションへと送信する、構造化されたデータです。メッセージを作成するアプリケーションを「プロデューサ」、メッセージを受信するアプリケーション（レシーバ）を「コンシューマ」と呼びます。TIBCO EMS サーバーは、メッセージを中継する役割を果たし、正しいデスティネーション（宛先）へメッセージが配信されるよう管理します。また、フォールトトレランス、メッセージのルーティング、および他のメッセージングシステム（TIBCO Rendezvous<sup>®</sup>、TIBCO SmartSockets<sup>®</sup> など）との通信などの機能を企業規模で提供します。

図 1 は、あるアプリケーションで作成されたメッセージがサーバー経由で送信され、別のアプリケーションで受信される過程を示しています。

図 1 メッセージの配信



JMS は、以下のメッセージングモデルをサポートしています。

- **ポイントツーポイント**（キュー）
- **パブリッシュとサブスクライブ**（トピック）
- **マルチキャスト**（トピック）

### ポイントツーポイント

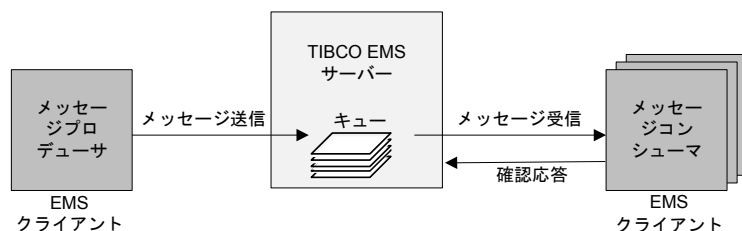
ポイントツーポイントメッセージングでは、メッセージごとに、ひとつのプロデューサとひとつのコンシューマが存在します。このメッセージングモデルでは、受信されるまでメッセージを格納するために「キュー」を使用します。メッセージプロデューサはキューにメッセージを送信します。一方、メッセージコンシューマはキューからメッセージを取り出し、メッセージを受信したことを示す確認応答を返します。

複数のプロデューサが同じキューにメッセージを送信し、複数のコンシューマが同じキューからメッセージを取り出すことが可能です。キューは必要に応じて、排他キューとして設定できます。排他キューでは、そのキューに対して指定されている最初のコンシューマだけがキューメッセージを取得できます。排他キューは、特定のキューに対するメッセージをひとつのアプリケーションだけが受信す

るようにしたい場合に有用です。一方、非排他キューでは、いくつものレシーバがそのキューからメッセージを取得できます。非排他キューは、受信メッセージの負荷を複数のレシーバ間で分散させたい場合に有用です。キューが排他であるかどうかにかかわらず、キューに入れられた各メッセージをコンシュームできるのは、ひとつのコンシューマだけです。

図2は、非排他キューを使用したポイントツーポイントのメッセージングを示しています。各メッセージコンシューマは、キューからメッセージを受信し、メッセージを受信したことを確認応答します。メッセージはキューから削除されるため、他のコンシューマがそのメッセージを受信することはありません。

図2 ポイントツーポイントメッセージ



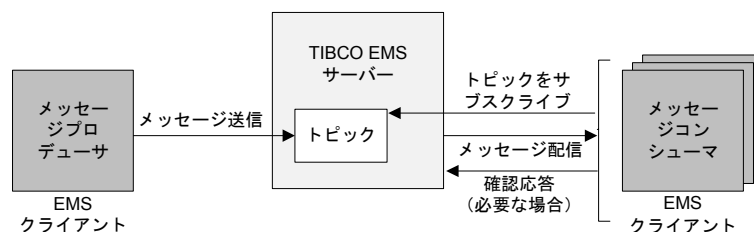
## パブリッシュとサブスクライブ

パブリッシュ - サブスクライブ型のメッセージングシステムでは、プロデューサは「トピック」宛てにメッセージを送信します。このモデルでは、プロデューサは「パブリッシャ」と呼ばれ、コンシューマは「サブスクライバ」と呼ばれます。

複数のパブリッシャが同じトピックに対してパブリッシュでき、ひとつのパブリッシャからのメッセージを複数のサブスクライバが受信できます。サブスクライバはトピックをサブスクライブし、あるトピックにパブリッシュされたすべてのメッセージは、そのトピックのすべてのサブスクライバが受信します。このタイプのメッセージプロトコルは、「ブロードキャスト」メッセージングとも呼ばれます。ラジオ信号やテレビ信号の送受信に似たような方法で、ネットワーク経由で送信されたメッセージが、そのメッセージを必要とするすべてのサブスクライバによって受信されるからです。

図3は、パブリッシュとサブスクライブのメッセージングを示しています。各メッセージコンシューマはトピックをサブスクライブします。そのトピックにメッセージがパブリッシュされると、サブスクライブしているすべてのコンシューマがそのメッセージを受け取ります。

図3 メッセージのパブリッシュとサブスクライブ



## トピックの持続サブスクライバ

デフォルトでは、メッセージを受け取るのはアクティブなサブスクライバのみです。トピックにメッセージが届いたときにアクティブになっていないサブスクライバは、それらのメッセージを受け取りません。

EMS API を使用して持続サブスクライバを作成すると、メッセージコンシューマが現在起動していない場合でも、メッセージを受信することができます。持続サブスクリプション宛てのメッセージは、トピックに持続サブスクライバが存在する限り、またはメッセージの有効期限が切れるか、トピックの保存量の制限値に達するまで、サーバーに保存されます。持続サブスクライバは、メッセージが送信されたときにアクティブでなかった場合でも、持続サブスクリプションからメッセージを受信することができます。

アプリケーションが再起動して、持続サブスクライバが同じ ID で再作成されると、サーバーに保存されていたそのトピックのすべてのメッセージが、その持続サブスクライバに配信されます。

持続サブスクライバの作成方法についての詳細は、336 ページの「[メッセージコンシューマの作成](#)」を参照してください。

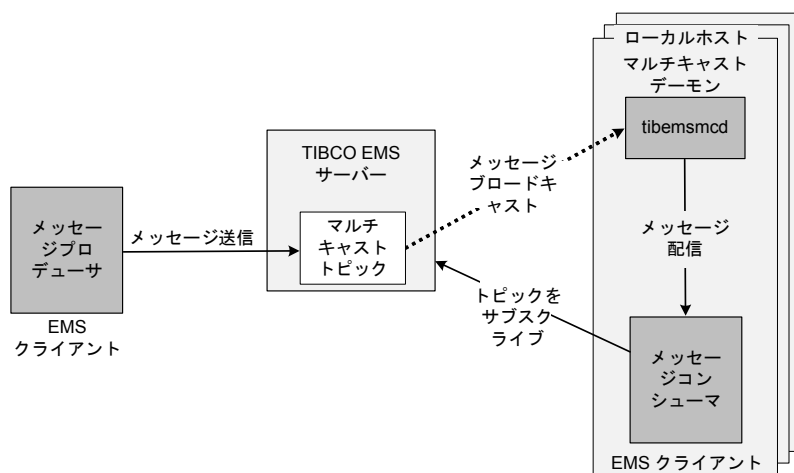
## マルチキャスト

マルチキャストメッセージングとは、ひとつのメッセージプロデューサから、サブスクライブしている複数のコンシューマ宛てにメッセージを同時送信することです。パブリッシュ - サブスクライブ型のメッセージングモデルと同様、メッセージプロデューサはトピック宛てにメッセージを送信します。TCP を介して個別のサブスクライバ宛てに 1 通ずつメッセージを送信するのではなく、Pragmatic General Multicast (PGM) を介して EMS サーバーからメッセージをブロードキャストします。EMS クライアントがサブスクライバとなっているマシンで実行するデーモンはマルチキャストメッセージを受信すると、そのメッセージをメッセージコンシューマに配信します。

マルチキャストはメッセージのブロードキャストに使用される帯域幅だけでなく、使用される EMS サーバーのリソースも削減できるため、スケーラビリティに優れています。ただし、マルチキャストはすべてのサブスクライバに対するメッセージの配信を保証しません。

6 ページの「[図 4](#)」は、マルチキャストメッセージングモデルを示しています。各メッセージコンシューマはマルチキャスト対応のトピックをサブスクライブします。該当するトピックにメッセージが送信されると、EMS サーバーからメッセージがブロードキャストされます。受信側マルチキャストデーモンがメッセージを受信し、サブスクライブしているクライアントに配信します。

図 4 マルチキャストメッセージ



マルチキャストについての詳細は、357 ページの[第 13 章](#)、「[マルチキャストの使用](#)[方法](#)」を参照してください。

## EMS のデスティネーション機能

TIBCO Enterprise Message Service では、デスティネーションを設定することで、各メッセージングモデルの機能を拡張することができます。

EMS のデスティネーション機能には、以下の機能があります。

- キューまたはトピックレベルでアクセス制御に [secure](#) モードを設定し、デスティネーションごとにアクセス権限が必要かどうかを設定できます。266 ページの「[デスティネーション制御](#)」を参照してください。
- トピックまたはキューのメッセージを保存するために EMS サーバーが使用するメモリ量のしきい値を設定できます。また、しきい値を超えた場合のサーバーの応答を細かく設定できます。58 ページの「[flowControl](#)」および 62 ページの「[overflowPolicy](#)」を参照してください。
- デスティネーションに送信されたメッセージを他のサーバーにルーティングできます。497 ページの「[ルートの使用](#)」を参照してください。
- 同じタイプまたは異なるタイプのデスティネーション間にブリッジを作成し、混合メッセージングモデルのアプリケーションを作成することもできます。同じメッセージをトピックとキューの両方に送信する必要があるアプリケーションの場合、この機能を使用すると便利です。デスティネーション間にブリッジを作成する方法およびブリッジを適用できる状況についての詳細は、78 ページの「[デスティネーションブリッジ](#)」を参照してください。
- デスティネーションへのメッセージのフローを制御できます。メッセージコンシューマがメッセージを受け取る速度よりもメッセージプロデューサがメッセージを送る速度が速い場合、フロー制御によって送受信の量を調整可能です。メッセージのフロー制御についての詳細は、83 ページの「[フロー制御](#)」を参照してください。
- 他のメッセージサービスとメッセージを交換できます。キューは、TIBCO Rendezvous と TIBCO SmartSockets のメッセージを受信できます。トピックは、Rendezvous と TIBCO SmartSockets のメッセージを送受信できます。389 ページの「[TIBCO Rendezvous との相互運用](#)」および 413 ページの「[TIBCO SmartSockets との相互運用](#)」を参照してください。
- キューを排他キューまたは非排他キューに設定できます。排他キューからは、ひとつのレシーバしかメッセージを受信できません。一方、非排他キューからは、複数のレシーバがメッセージを受信できます。56 ページの「[exclusive](#)」を参照してください。
- キューに再配信ポリシーを指定できます。メッセージを再配信する必要がある場合、キューのプロパティを設定して、メッセージが再配信される最大回数を指定できます。61 ページの「[maxRedelivery](#)」を参照してください。

- デスティネーションを経由するすべてのメッセージをトレースし、ログを作成できます。70 ページの「[trace](#)」を参照してください。
- メッセージプロデューサのユーザー名をメッセージに含めることができます。68 ページの「[sender\\_name](#)」および 68 ページの「[sender\\_name\\_enforced](#)」を参照してください。
- 管理者の操作で、デスティネーション名にワイルドカードを使用できます。ワイルドカードのデスティネーション名は親であるため、ワイルドカードのデスティネーション名と一致する名前はすべて親のプロパティを継承します。73 ページの「[ワイルドカード](#)」を参照してください。
- **store** プロパティを使用して、デスティネーションに送信するメッセージをストアファイルに書き込めます。デスティネーションストアを **store=\$sys.fail-safe** と設定してメッセージをファイルに同期書き込みすれば、どのような状況でもメッセージが消失しないようにできます。詳細は、69 ページの「[store](#)」を参照してください。
- システムのパフォーマンスを向上させるために、メッセージがバックグラウンドでバッチ受信されるようにコンシューマを設定できます。または、キューから一度にひとつのメッセージしか受信しないようにレシーバ（コンシューマ）を設定する方法もあります。詳細は、64 ページの「[prefetch](#)」を参照してください。



## クライアント API

---

Java アプリケーションは、`javax.jms` パッケージを使用して JMS メッセージの送受信を行います。これは、JMS 仕様によって指定されている標準セットのインタフェースです。このインタフェースを使用して、EMS サーバーへの接続の作成、送信するメッセージのタイプの指定、メッセージを送受信するデスティネーション（トピックまたはキュー）の作成などを行います。`javax.jms` パッケージについての詳細は、オンラインマニュアルの『TIBCO Enterprise Message Service Java API リファレンス』の説明を参照してください。EMS には、JMS 標準が実装されているため、[java.sun.com/products/jms/index.html](http://java.sun.com/products/jms/index.html) のサイトで、JMS 仕様とともに、これらのインタフェースに関する資料を参照することもできます。

TIBCO Enterprise Message Service には、他の開発環境のためのパラレル API が用意されています。詳細は、以下のマニュアルを参照してください。

- TIBCO Enterprise Message Service C & COBOL API リファレンス
- 『TIBCO Enterprise Message Service .NET API リファレンス』（オンラインマニュアル）

## サンプルコード

EMS には、サンプルプログラムがいくつか含まれています。これらのサンプルプログラムには、EMS のさまざまな機能が記述されています。各機能についてこのマニュアルの説明を読む際に、これらのサンプルプログラムを参照することができます。サンプルファイルは、EMS のインストールディレクトリの `samples` サブディレクトリにあります。

サンプルコードの実行についての詳細は、87 ページの第 4 章、「基本操作」を参照してください。

## TIBCO Rendezvous Java アプリケーション

EMS には、ピュア Java の TIBCO Rendezvous アプリケーションを EMS サーバーに直接接続できる Java クラスが含まれています。詳細は、411 ページの「[Pure Java Rendezvous プログラム](#)」を参照してください。

# 管理

EMS は、サーバーの動作を管理し、サーバーの管理対象となるオブジェクト (ConnectionFactory や Destination など) を作成するメカニズムを提供します。

管理機能は、コマンドライン管理ツールを使用するか、管理 API (Java または .NET) を使用するアプリケーションを作成する方法のいずれかで実行することができます。コマンドライン管理ツールについての詳細は、117 ページの第 6 章、「EMS 管理ツールの使用」を参照してください。管理 API についての詳細は、オンラインマニュアルを参照してください。

管理インタフェースを使用して、管理対象オブジェクト (ConnectionFactory、Topic、Queue など) を作成し、管理できます。EMS クライアントは、JNDI (Java Naming and Directory Interface) を使用して、これらの管理対象オブジェクトへの参照を取得することができます。静的な管理対象オブジェクトを作成することにより、クライアント内にオブジェクトを実装させることなく、クライアントにこれらのオブジェクトを使用させることができます。

## サーバーの管理

EMS の管理機能を使用して、サーバーを監視および管理できます。以下の表に、各管理機能の概要とこのマニュアルの中でその機能を説明しているセクションを示します。

表 3 管理機能の概要 (1 / 2 シート)

機能	詳しい説明のあるセクション
サーバーの特性を指定できる設定ファイル。	175 ページの第 7 章、「設定ファイルの使用」
サーバー管理に使用するコマンドラインインタフェースを提供する管理ツール。	117 ページの第 6 章、「EMS 管理ツールの使用」
サーバーおよびデスティネーションへのアクセスを制限できる認証およびアクセス権限。管理者アクセス権限を使用して、管理上のアクティビティを実行できるユーザーを指定することも可能。	255 ページの第 8 章、「認証とアクセス権限」
サーバーのさまざまなアクティビティに関する情報を提供するログファイルの設定。	433 ページの第 17 章、「サーバーアクティビティの監視」

表 3 管理機能の概要 ( 2 / 2 シート)

機能	詳しい説明のあるセクション
さまざまなシステムイベントの発生時に、サーバーはメッセージをパブリッシュする。これにより、これらのシステム監視トピックをサブスクライブする堅牢な監視アプリケーションを作成可能。	433 ページの <a href="#">第 17 章、「サーバーアクティビティの監視」</a>
必要に応じた詳細度で、サーバーからさまざまな統計情報を取得できる。	433 ページの <a href="#">第 17 章、「サーバーアクティビティの監視」</a>

## ユーザーとグループの管理

EMS では、サーバーのユーザーおよびグループをローカルに作成し、管理することができます。EMS サーバーは、ユーザー認証やグループ情報の保存に、LDAP サーバーなどの外部システムを使用できます。ユーザーおよびグループの管理に使用する外部システムを操作するための EMS の設定についての詳細は、255 ページの [第 8 章、「認証とアクセス権限」](#) を参照してください。

## TIBCO Hawk の使用方法

EMS サーバーの監視および管理に TIBCO Hawk<sup>®</sup> を使用できます。詳細は、TIBCO Hawk のマニュアルを参照してください。

## セキュリティ

---

サーバーとクライアント間、およびサーバー同士の通信のセキュリティには、EMS 内で明示的に SSL (Secure Sockets Layer) を設定する必要があります。

SSL は、暗号化されたデータをインターネットや内部ネットワークで転送するためのプロトコルです。SSL プロトコルでは、SSL 接続で転送されるデータは公開鍵と秘密鍵で暗号化されます。ほとんどの Web ブラウザは SSL をサポートしており、多くの Web サイトや Java アプリケーションは SSL プロトコルを使用して、クレジットカード番号などのユーザーの個人情報を取得しています。

EMS では、以下のコンポーネント間で SSL がサポートされます。

- EMS クライアントと EMS サーバー間
- 管理ツールと EMS サーバー間
- 管理 API と EMS サーバー間
- ルーティング先サーバー間
- フォールトトレランスサーバー間

EMS でサポートされる SSL についての詳細は、453 ページの[第 18 章、「SSL プロトコルの使用」](#)を参照してください。

## フォールトトレランス

---

EMS サーバーをプライマリサーバーおよびバックアップサーバーとして設定し、フォールトトレランスの環境を構築することができます。プライマリサーバーとバックアップサーバーはペアで動作します。プライマリサーバーはクライアント接続を受け入れてメッセージ処理作業を行い、セカンダリサーバーは障害発生時にバックアップサーバーとして動作します。アクティブなサーバーに障害が発生すると、バックアップサーバーはオペレーションを引き継ぎ、プライマリアクティブサーバーになります。

EMS のフォールトトレランス機能についての詳細は、473 ページの[第 19 章、「フォールトトレランス」](#)を参照してください。

## ルーティング

---

EMS では、サーバー間でメッセージがルーティングされるように、サーバーを設定できます。閉路がない場合（つまり、メッセージは、一度訪れたことのあるサーバーにはルーティングされない）、複数のホップに渡ってトピックメッセージをルーティングできます。キューメッセージは、キューを所有しているサーバーから任意の他のサーバーに 1 ホップだけで到達できます。

EMS では、ほとんどの場合、メッセージを保存して転送することで、ルーティングが行われていない場合でもオペレーションが行われるようになっています。

EMS のルーティング機能についての詳細は、497 ページの [第 20 章、「ルートの使用」](#) を参照してください。

## サードパーティ製品との統合

---

EMS は、サードパーティ製の名前付け / ディレクトリサービス製品またはサードパーティ製のアプリケーションサーバーと統合できます。

## トランザクションのサポート

---

TIBCO Enterprise Message Service は、Java Transaction API (JTA) に準拠したトランザクションマネージャと統合できます。EMS には、JTA への準拠に必要なすべてのインタフェースが実装されています。EMS C API は、X/Open XA 仕様に準拠しています。EMS の .NET API は、Microsoft Distributed Transaction Coordinator (MS DTC) をサポートしています。.NET クライアントで MSDTC を使用して作成したトランザクションは、C および Java クライアントでは XA トランザクションとみなされます。



## 第2章

# メッセージ

この章では、EMS メッセージの概要を説明します。

### トピック

---

- 「[EMS の機能を拡張した JMS メッセージ](#)」、16 ページ
- 「[JMS メッセージの構造](#)」、17 ページ
- 「[メッセージの優先度](#)」、23 ページ
- 「[メッセージ配信モード](#)」、24 ページ
- 「[EMS による永続的メッセージの管理方法](#)」、26 ページ
- 「[複数ストアへのメッセージの保存](#)」、29 ページ
- 「[メッセージの文字エンコーディング](#)」、35 ページ
- 「[メッセージの圧縮](#)」、37 ページ
- 「[メッセージの確認応答](#)」、38 ページ
- 「[同期および非同期のメッセージコンシューム](#)」、45 ページ

## EMS の機能を拡張した JMS メッセージ

---

JMS 仕様では、メッセージのヘッダと本文の標準フォーマットが定義されています。プロパティはプロバイダが独自に定義でき、JMS の機能に対してプロバイダ固有の実装や拡張機能を追加することができます。EMS 固有のメッセージプロパティについては、19 ページの「[EMS メッセージのプロパティ](#)」のリストを参照してください。

EMS のメッセージプロパティに加え、EMS は厳選された拡張機能を JMS に提供します。EMS の拡張機能は以下のとおりです。

- JMS 仕様では、PERSISTENT と NON\_PERSISTENT のふたつのメッセージ配信モードが規定されています。EMS には、さらに、他の配信モードに関するオーバーヘッドを低減する RELIABLE\_DELIVERY モードが追加されています。[25 ページの「RELIABLE\\_DELIVERY」](#)を参照してください。
- コンシューマのセッションでは、必要に応じて、メッセージの確認応答をしなくても済むように NO\_ACKNOWLEDGE モードを指定することができます。EMS では、個々のメッセージに対する確認応答を制限するために、さらに EXPLICIT\_CLIENT\_ACKNOWLEDGE と EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE のふたつのモードが用意されています。[38 ページの「メッセージの確認応答」](#)を参照してください。
- EMS では、[MapMessage](#) および [StreamMessage](#) の本文タイプが拡張されています。これらの機能を拡張することにより、EMS では JMS の MapMessage および StreamMessage では利用できない機能を持つ TIBCO Rendezvous と ActiveEnterprise のフォーマットでメッセージを交換することができます。

TIBCO Enterprise Message Service では、これらのふたつの拡張機能を、MapMessage および StreamMessage の本文タイプに追加します。

- MapMessage または StreamMessage に、別の MapMessage または StreamMessage のインスタンスをサブメッセージとして追加することで、フラットなメッセージではなく、ネストされた一連のメッセージを作成できます。
- プリミティブ型だけでなく配列型の値も使用できます。

これらの機能拡張によって、MapMessage および StreamMessage の本文タイプの柔軟性が著しく増します。ただし、これは拡張機能であるため、JMS 仕様には準拠していません。拡張機能で作成されたメッセージには、ベンダーによる機能拡張であることを示すプロパティタグ [JMS\\_TIBCO\\_MSG\\_EXT](#) が付けられます。

Rendezvous メッセージとの互換性についての詳細は、[407 ページの「メッセージ本文」](#)を参照してください。



## JMS メッセージの構造

JMS メッセージは、標準化構造を持っています。この構造は、以下の部分で構成されています。

- ヘッダ（必須）
- プロパティ（オプション）
- 本文（オプション）

### JMS メッセージのヘッダフィールド

ヘッダには、メッセージのルーティングと配信に使用される 10 個の定義済みフィールドが含まれます。表 4 で、JMS メッセージの各ヘッダフィールドについて説明します。

表 4 JMS メッセージのヘッダ

ヘッダフィールド	設定方法	説明
JMSDestination	send または publish メソッド	メッセージのデスティネーション（送信先）。
JMSDeliveryMode	send または publish メソッド	永続的または非永続的メッセージ。デフォルトは永続的メッセージ。  EMS には、配信モードを拡張した RELIABLE_DELIVERY モードが用意されています（25 ページの「RELIABLE_DELIVERY」を参照）。

表4 JMS メッセージのヘッダ

ヘッダフィールド	設定方法	説明
JMSExpiration	send または publish メソッド	<p>メッセージの有効期限。0 に設定すると、メッセージは期限切れになりません。有効期限は、ミリ秒単位で指定します。</p> <p>サーバーの <b>expiration</b> プロパティをデスティネーションで設定すると、メッセージプロデューサが設定した JMSExpiration の値が無効にされます。</p> <p>EMS バージョン 4.4 以降では、クライアントのクロックは接続の作成時にサーバーと自動同期されます。ただし、長期間にわたる接続では、ネットワークタイムプロトコル (NTP: Network Time Protocol) がサーバーとクライアントの間の継続的な同期を維持する最も信頼性の高い方法です。また、EMS サーバーまたはクライアントアプリケーションが EMS バージョン 4.4 より前のバージョンに基づいている場合、クライアントアプリケーションのメッセージの有効期限にゼロ以外の値が設定されているときは、メッセージを送受信するすべてのホスト間でクロックが同期していることを確認する必要があります。クロックの許容誤差を、最も短いメッセージの有効期限と比べて非常に小さい値に抑えて同期させます。</p>
JMSPriority	send または publish メソッド	<p>0 ～ 9 の数値順位で、標準または優先としてメッセージの優先度を定義。値が大きいほど優先順位が高くなります。</p> <p>詳細は、23 ページの「<a href="#">メッセージの優先度</a>」を参照してください。</p>
JMSMessageID	send または publish メソッド	プロバイダによって送信された各メッセージを個々に識別する値。
JMSTimestamp	send または publish メソッド	送信するようにメッセージがプロバイダに渡された時刻のタイムスタンプ。実際のメッセージはこのタイムスタンプの時刻よりも後に送信されることがあります。
JMSCorrelationID	メッセージクライアント	メッセージ間のリンク（要求メッセージと応答メッセージとのリンクなど）に使用できる ID。このフィールドの入力はオプションです。JMS 関連 ID の推奨最大サイズは 4 KB です。これより大きい値の場合は、メッセージが拒否される可能性があります。

表 4 JMS メッセージのヘッダ

ヘッダフィールド	設定方法	説明
JMSReplyTo	メッセージクライアント	メッセージ応答のデスティネーション（送信先）。このフィールドの入力はオプションです。
JMSType	メッセージクライアント	メッセージタイプの識別子。
JMSRedelivered	JMS プロバイダ	このフィールドが設定されている場合、メッセージが以前にクライアントに配信されたが、その時点では確認応答されなかった可能性があります。

## EMS メッセージのプロパティ

プロパティエリアで、JMS システムのアプリケーション、ベンダー、管理者のオプションのプロパティを追加できます。プロパティエリアの入力はオプションです。空白のままにしておくこともできます。JMS 仕様には、JMS メッセージのプロパティが規定されています。このセクションでは、EMS の固有のメッセージプロパティを説明します。

TIBCO 固有のプロパティ名は、JMS\_TIBCO の文字列で始まります。クライアントプログラムは、EMS の機能にアクセスするために TIBCO 固有のプロパティを使用しますが、クライアントプログラム間でアプリケーション固有の情報を通信するためには使用しません。

EMS のプロパティの概要について、表 5 で説明します。詳細は、この章の以降のセクションを参照してください。

表 5 メッセージプロパティの概要（1 / 2 シート）

プロパティ	説明	詳細情報
JMS_TIBCO_CM_PUBLISHER	TIBCO Rendezvous からインポートしたメッセージに対応する RVCM センダーの名前。	<a href="#">406</a>
JMS_TIBCO_CM_SEQUENCE	TIBCO Rendezvous からインポートした RVCM メッセージのシーケンス番号。	<a href="#">406</a>
JMS_TIBCO_COMPRESS	ストレージの効率を上げるためのメッセージ圧縮が可能。	<a href="#">37</a>

表5 メッセージプロパティの概要 (2 / 2 シート)

プロパティ	説明	詳細情報
JMS_TIBCO_DISABLE_SENDER	可能な場合、メッセージの送信元のユーザー名をメッセージに含めないように指定する。	21
JMS_TIBCO_IMPORTED	メッセージが Rendezvous または SmartSockets からインポートされたときにサーバーによって設定される。	406 426
JMS_TIBCO_MSG_EXT	MapMessage および StreamMessage の本文タイプの機能を拡張し、サブメッセージまたは配列を本文に含める。	16 406 426
JMS_TIBCO_MSG_TRACE	メッセージのトレースをプロデューサからコンシューマまで行うように指定します。	440
JMS_TIBCO_PRESERVE_UNDELIVERED	メッセージを削除する必要がある場合、未配信メッセージキューに入れるメッセージを指定します。	20
JMS_TIBCO_SENDER	メッセージの送信元のユーザー名が含まれます。	21
JMS_TIBCO_SS_SENDER	EMS サーバーがメッセージを TIBCO SmartSockets からインポートしたときに、このプロパティを SmartSockets sender ヘッダフィールド (SmartSockets 構文内) に設定します。	426

## 未配信メッセージキュー

メッセージの有効期限が切れたか、キューの [maxRedelivery](#) プロパティで指定されている値を超えた場合、サーバーはメッセージの

[JMS\\_TIBCO\\_PRESERVE\\_UNDELIVERED](#) プロパティをチェックします。

[JMS\\_TIBCO\\_PRESERVE\\_UNDELIVERED](#) が `true` に設定されている場合、サーバーは

メッセージを未配信メッセージキュー `$sys.undelivered` に移します。この未配信メッセージキューはシステムによって作成されるキューです。常に存在し、削除することはできません。`JMS_TIBCO_PRESERVE_UNDELIVERED` プロパティが `false` に設定されていると、メッセージがサーバーによって削除されます。

未配信メッセージキューを使用するには、メッセージを送信またはパブリッシュするアプリケーションが、メッセージを送信またはパブリッシュする前に、ブール値プロパティ `JMS_TIBCO_PRESERVE_UNDELIVERED` を `true` に設定する必要があります。

未配信プロパティはメッセージ単位でしか設定することができません。未配信メッセージキューをトピックごとまたはキューごとに設定する方法はありません。

キューレシーバを作成して、メッセージが未配信メッセージキューに入れられるたびに、受信と処理を行うようにすることができます。未配信メッセージキュー内のメッセージを受信せずに削除する場合は、管理ツールで `purge queue` コマンド (122 ページの「[コマンドのリスト](#)」を参照) を使用して、`$sys.undelivered` キューをページします。TIBCO Enterprise Message Service に含まれている管理用 API を使用して、メッセージを削除することもできます。

`$sys.undelivered` は [global](#) のデスティネーションプロパティ設定を無視します。未配信メッセージキューのメッセージは他のサーバーにルーティングされません。

## メッセージ送信元を含める

接続の確立時に、EMS は、メッセージにメッセージプロデューサのユーザー名を含めることができます。デスティネーションの `sender_name` と `sender_name_enforced` プロパティには、メッセージプロデューサのユーザー名を送信メッセージに含めるかどうかを指定します。

ユーザー名がメッセージに含まれている場合、[JMS\\_TIBCO\\_SENDER](#) という文字列のメッセージプロパティを取得することで、メッセージコンシューマはそのユーザー名を取得できます。

デスティネーションの `sender_name` プロパティを有効にし、`sender_name_enforced` プロパティを無効にすると、メッセージプロデューサはユーザー名を送信メッセージに含まれないように指定できます。メッセージプロデューサは、特定のメッセージに [JMS\\_TIBCO\\_DISABLE\\_SENDER](#) ブール値のプロパティを指定できます。これにより、メッセージプロデューサのユーザー名はメッセージに含まれません。ただし、`sender_name_enforced` プロパティが有効に設定されていると、`JMS_TIBCO_DISABLE_SENDER` プロパティは無視され、ユーザー名は常にメッセージに含まれます。

## JMS メッセージ本文

JMS メッセージには、いくつかあるタイプのメッセージ本文のうちのひとつがあるか、メッセージ本文がまったくないかのいずれかです。

メッセージのタイプについては、表 6 で説明します。

表 6 JMS メッセージタイプ

メッセージタイプ	メッセージ本文の内容
Message	このメッセージタイプには、本文はありません。これは単純なイベント通知に使用します。
TextMessage	<code>java.lang.String</code> 。
MapMessage	<p>名前 / 値のペアのセット。名前は <code>java.lang.String</code> オブジェクトです。値は <code>Java</code> のプリミティブ値型か、またはそのラッパーです。列挙された順番に、エントリにアクセスできます。または名前を指定して直接アクセスできます。エントリの順番は定義されていません。</p> <p>EMS が <code>Rendezvous</code> または <code>ActiveEnterprise</code> とメッセージを交換する場合、一連のネストされた <code>MapMessage</code> を生成することができます (16 ページの「<a href="#">EMS の機能を拡張した JMS メッセージ</a>」を参照)。</p>
BytesMessage	連続バイトのストリーム。バイトには型がありません。すなわち、プリミティブデータ型に割り当てられていません。
StreamMessage	<p><code>Java</code> プログラミング言語のプリミティブ値のストリーム。値の各セットはプリミティブデータ型に属しており、順番に読み取られる必要があります。</p> <p>EMS が <code>Rendezvous</code> または <code>ActiveEnterprise</code> とメッセージを交換する場合、一連のネストされた <code>StreamMessage</code> を生成することができます (16 ページの「<a href="#">EMS の機能を拡張した JMS メッセージ</a>」を参照)。</p>
ObjectMessage	<code>Java</code> プログラミング言語で構築されたシリアル化可能なオブジェクト。

メッセージの最大サイズ

EMS では、最大 512MB のメッセージがサポートされます。ただし、メッセージがこの最大サイズに近づくにつれ、現在のハードウェアおよびオペレーティングシステムのプラットフォームのパフォーマンスの制限を圧迫してしまうため、アプリケーションプログラムで使用するメッセージは小さいサイズにすることを勧めます。

## メッセージの優先度

---

JMS 仕様には、送信元が 0 ～ 9 の値の範囲でメッセージの優先度を設定できる [JMSPriority](#) メッセージヘッダフィールドがあります。EMS では、メッセージの優先度をオプションでサポートしています（他のベンダーでは実装されていない場合があります）。

コンシューマクライアントに配信する準備ができたメッセージが複数ある場合、EMS サーバーは優先度の高いメッセージから先に配信します。

ただし、優先順位が適用されるのは、配信可能なコンシューマ宛てのメッセージのバックログがサーバーにある場合のみです。コンシューマに一度に配信されるメッセージがひとつしかサーバーにない場合、優先順位の機能に目立つ効果はありません。

メッセージプロデューサのデフォルトのメッセージ優先度を設定することができます（334 ページの「[メッセージプロデューサの設定](#)」を参照）。デフォルトの優先度は、メッセージを送信する際にクライアントによって無効にされることがあります（343 ページの「[メッセージの送信](#)」を参照）。

**関連トピック**    JMS 仕様（3.4.10）

## メッセージ配信モード

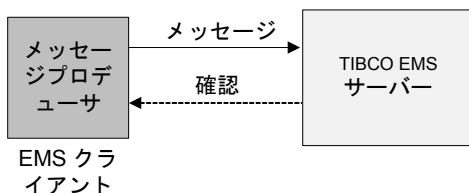
`JMSDeliveryMode` のメッセージヘッダフィールドで、メッセージの配信モードを定義します。JMS は、トピックとキューの両方で `PERSISTENT` および `NON_PERSISTENT` の配信モードをサポートしています。EMS では、これらの配信モードの機能を拡張し、さらに `RELIABLE_DELIVERY` モードを用意しています。

メッセージプロデューサのデフォルトの配信モードを設定することができます (334 ページの「メッセージプロデューサの設定」を参照)。デフォルトの配信モードは、メッセージを送信する際にクライアントによって無効にされることがあります (343 ページの「メッセージの送信」を参照)。

### PERSISTENT

図 5 に示されているように、プロデューサが `PERSISTENT` メッセージを送信する際、サーバーから確認の応答があるまで待機する必要があります。メッセージは、サーバーによってディスクに保管されます。この配信モードでは、どのような場合でも、サーバー上のデスティネーションへ確実にメッセージが配信されます。ただし、この配信モードでは、各メッセージやグループメッセージのトランザクションのコミットの際に双方向のネットワークトラフィックが発生するのが難点です。

図 5 永続的メッセージ配信



### NON\_PERSISTENT

`NON_PERSISTENT` メッセージを送信すると、ディスク上にメッセージを保管しておくオーバーヘッドがなくなるため、システムのパフォーマンスが向上します。

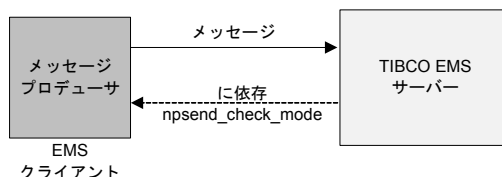
サーバーでの `authorization` を無効にすると、サーバーはメッセージプロデューサに確認メッセージを送信しません。

サーバーでの `authorization` を有効にすると、デフォルトの条件として、`PERSISTENT` モードの場合と同様に、プロデューサはサーバーから確認の応答があるまで待機する必要があります。



`authorization` が有効かどうかに関わらず、`tibemsd.conf` ファイル内の `npsend_check_mode` パラメータを設定して、サーバーがプロデューサに `NON_PERSISTENT` メッセージの確認を送信する条件を設定できます。詳細は、192 ページの「`npsend_check_mode`」の説明を参照してください。

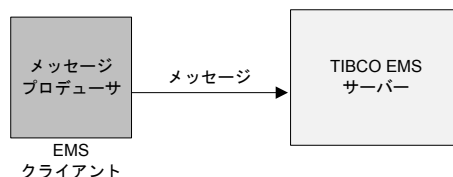
図 6 非永続的メッセージ配信



## RELIABLE\_DELIVERY

EMS は、JMS の配信モードを拡張し、信頼性の高いメッセージ配信を実現しています。RELIABLE\_DELIVERY メッセージを送信すると、サーバーでの確認処理が省かれるため、`authorization` の設定に関係なくシステムのパフォーマンスが向上します。

図 7 信頼メッセージ配信



RELIABLE\_DELIVERY モードを使用すると、サーバーはプロデューサに受領確認やアクセス拒否などのメッセージを送信しないため、プロデューサは待機する必要がありません。信頼モードは、メッセージのトラフィック量を削減し、メッセージの転送量を増やします。このモードは、たとえば、株価相場などの時間依存型データを含むメッセージを送信する場合に利用できます。

信頼配信モードを使用すると、クライアントアプリケーションはサーバーから応答を受け取りません。したがって、サーバーへの接続が切断されない限り、すべてのパブリッシュ呼び出しは常に成功します（例外をスローすることはありません）。

場合によっては、信頼モードでパブリッシュされたメッセージのデスティネーションが有効でなかったり、アクセスが許可されなかったために、そのメッセージがサーバーによって不適格とされ、処理されないことがあります。この場合、メッセージはメッセージコンシューマには配信されません。ただし、サーバーへの接続が切断されない限り、メッセージを受け取ったコンシューマが存在しない場合でも、メッセージをパブリッシュしたアプリケーションは例外を受け取ることはありません。

## EMS による永続的メッセージの管理方法

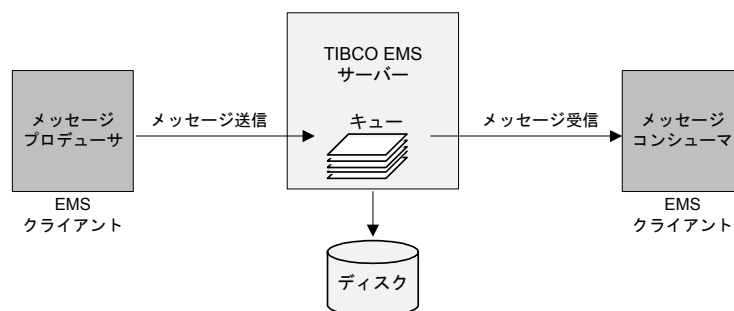
24 ページの「[メッセージ配信モード](#)」の説明のとおり、JMS では `PERSISTENT` および `NON_PERSISTENT` のふたつのメッセージ配信モードが定義され、EMS では [RELIABLE\\_DELIVERY](#) モードが定義されています。

`NON_PERSISTENT` メッセージと `RELIABLE_DELIVERY` メッセージは、永続的なストレージには書き込まれません。`PERSISTENT` メッセージは、EMS サーバーによって受信されると、永続的なストレージに書き込まれます。

### キューに送信される永続的メッセージ

キューに送信された永続的メッセージは常にディスクに書き込まれます。永続的メッセージがサブスクライバに送信される前にサーバーに障害が発生すると、サーバーは再起動され、サブスクライバがサーバーに再接続したときに、永続的メッセージがサブスクライバに送信されます。

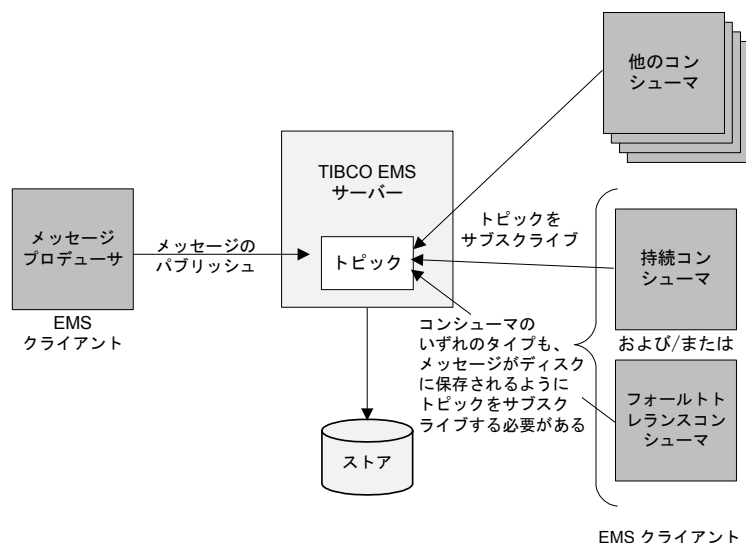
図 8 キューに送信される永続的メッセージ



### トピックにパブリッシュされる永続的メッセージ

トピックにパブリッシュされた永続的メッセージがディスクに書き込まれるのは、そのトピックに、持続サブスクライバまたは EMS サーバーにフォールトトレランス接続をしているサブスクライバが少なくともひとつある場合です。持続サブスクライバまたはフォールトトレランス接続をしているサブスクライバがないということは、サーバーの障害時に、メッセージの再送信を必要とするサブスクライバが存在しないということです。この場合、サーバーは不必要に永続的メッセージを保存しません。これにより、メッセージを保存するディスクへのデータ書き込み / 読み取りが低減されるので、システムのパフォーマンスが向上します。

図9 トピックにパブリッシュされる永続的メッセージ



この動作が JMS 仕様に準拠しているのは、パブリッシュされたメッセージがトピックの持続サブスクライバによって保存されるためです。また、フォールトトレランス接続を利用するトピックのサブスクライバは、フェイルオーバーの後にセカンダリサーバーからメッセージを受け取る必要があります。ただし、フォールトトレランス接続を使用しない非持続サブスクライバがサーバーの障害後に再接続を行うと、新しく作成されたサブスクライバと見なされ、その作成時刻より以前に作成されたメッセージを受け取る資格がありません（サブスクライバが再接続する前にパブリッシュされたメッセージは再送信されない）。

## 永続的メッセージおよび同期ファイルストレージ

ファイルストレージ使用時は、デフォルトでは EMS サーバーによって受信された永続的メッセージは非同期的にディスクに書き込まれます。つまり、プロデューサが永続的メッセージを送信する際、サーバーはディスク書き込み処理の完了を待たずに、プロデューサに制御を返します。ディスク書き込み処理が完了する前にサーバーに障害が発生した場合、メッセージの保存に障害が起きたことを検出し、適切なアクションを実行する方法がプロデューサにはありません。

**stores.conf** ファイルで特定のファイルストレージの **mode** パラメータを **sync** に設定し、トピックまたはキューの永続的メッセージが同期的にディスクに書き込まれるように指定できます。**mode = sync** とすると、サーバーでディスク書き込み処理が完了するまで、永続的プロデューサはブロックされます。

各 EMS サーバーは、永続的メッセージをストアファイルに書き込みます。ふたつのサーバーが同じストアファイルを使用しないように、どちらのサーバーも、処理中のストアファイルへのアクセスを制限します。EMS による共有ストアファイルの管理方法についての詳細は、114 ページの「[EMS による共有ストアファイルへのアクセスの管理方法](#)」を参照してください。

## 複数ストアへのメッセージの保存

24 ページの「[メッセージ配信モード](#)」の説明のとおり、EMS サーバーはサブスクリバからの受信確認を待っている間に **PERSISTENT** メッセージをディスクに書き込みます。メッセージは永続的にストアに保持されます。EMS サーバーは、ファイルベースのストア、**mstore**、およびデータベースストアなどの異なるタイプのストアにメッセージを書き込むことができます。

デフォルトでは、EMS サーバーは永続的メッセージをファイルベースのストアに書き込みます。デフォルトのストアファイルは三つあります（30 ページの「[デフォルトのストアファイル](#)」を参照）。デフォルトのストアファイルとその場所は、システム設定で変更できます。また、永続的メッセージをひとつ以上のストアファイルに保存したり、デスティネーション別にフィルタリングしたりできます。ストアは **stores.conf** 設定ファイルで定義されます。デスティネーションとの関連付けには **store** デスティネーションプロパティが使用されます。

ストアには、ストアファイルをサーバーでどのように管理するかを制御できるプロパティがあります。例：

- ファイルベースのストアを使用する場合
  - ストアファイル用にディスク領域を事前に割り当てる
  - 定期的にファイルを切り捨て、ディスク領域を解放する
  - メッセージを同期と非同期のどちらで書き込むかを指定する
- データベースにメッセージを保存する

複数ストア機能を使うと、メッセージングアプリケーションのメッセージをアプリケーション別に異なる場所に保存するよう設定できます。または、関連デスティネーションで個別ファイルを作成できます。たとえば、「マーケティング」のメッセージを保存するストアと、「営業」のメッセージを保存するストアをひとつずつ作成できます。ストアはサーバーに設定されるため、クライアントには透過的です。

管理者は、EMS 管理ツールの **show stores** コマンドや **show store** コマンドを使ってシステムに設定されているストアとその設定内容を確認できます。

## ストアのタイプ

TIBCO Enterprise Message Service では、以下に説明する各種のストアタイプを設定できます。

### ファイルベースのストア

EMS サーバーは、永続的メッセージをファイルベースのストアに保存します。デフォルトのストアファイルを使用するか、独自のファイルベースのストアを作成できます。EMS サーバーでメッセージをストアファイルに書き込むようにするには、デスティネーションをストアに関連付けます。

ファイルベースのストアはデフォルトで有効になっており、以下で説明する三つのデフォルトストアが自動定義されます。デフォルトストアを使用するのに必要な操作は特にありません。

31 ページの「[複数ストアの設定](#)」のセクションでは、ストア設定の変更方法およびカスタムストアの作成方法を説明します。

## mstore

mstore は、フェイルオーバー後の回復をすみやかに行うように設計されています。mstore を使用すると、EMS サーバーの起動は高速ですが、mstore のキャッシュが完全に読み込まれるまでの動作は遅くなる場合があります。これは、EMS サーバーがバックグラウンドで継続的にストアを監視するためです。サーバーは mstore を徐々に読み込みながら、パージされたメッセージや期限切れメッセージなどの古いデータを破棄します。

その結果、期限切れのメッセージやパージされたメッセージは mstore からすぐに削除されません。コンシューマに配信されないにもかかわらず、ファイルベースのストアやデータベースストアよりも長期間、ストアに残ります。これらのメッセージが破棄されるのは、ストアの定期スキャン時です。スキャン時の動作はストア設定に含まれるパラメータの設定によって決まります。詳細は、32 ページの「[mstore の間隔について](#)」を参照してください。

この動作が原因で、サーバーに合計保留メッセージ数を問い合わせると、不正確な値が返される場合があります。ただし、個別のデスティネーションに対するクエリでは、正確な数が返されます。

31 ページの「[複数ストアの設定](#)」のセクションでは、mstore の設定プロセスを説明します。

## データベースストア

EMS サーバーは、ひとつ以上のデータベースインスタンスにメッセージを保存できます。データベースストアは、サポート対象のデータベースを使用するように設定する必要があります。この機能の詳細は、[第10章](#)、「[データベースストアの使用](#)」を参照してください。

## デフォルトのストアファイル

EMS サーバーでは、次のデフォルトのストアファイルを定義し、永続的メッセージおよびメタデータをこれらのストアファイルに書き込みます。

- **\$sys.nonfailsafe** — デフォルトでは、store プロパティが指定されていない永続的メッセージは \$sys.nonfailsafe に書き込まれます。サーバーは非同期 I/O 呼び出しを使用してこのストアにメッセージを書き込みます。
- **\$sys.failsafe** — メッセージを同期的に書き込むには、デスティネーションをこのストアに関連付けます。サーバーは同期 I/O 呼び出しを使用してこのストアにメッセージを書き込みます。

- **\$sys.meta** — サーバーでは、持続サブスクライバ、フォールトトレランス接続の状態情報、およびその他のメタデータをこのストアに書き込みます。

EMS サーバーは、これらのファイルベースのストアを自動的に作成します。これらのストアを有効化または実装するために必要な手順はありません。ただし、システム設定を変更して、デフォルトのストアファイル設定をカスタマイズしたり、他のファイルの場所を参照するか、**mstore** またはデータベースに書き込むようにデフォルトのストア設定を無効にしたりすることもできます。



**\$sys.meta** ストアは **mstore** タイプを使用するように再設定することはできません。

## 複数ストアの設定

このセクションでは、ファイルベースのストアおよび **mstore** の設定に必要な基本手順について説明します。データベースストアの設定の詳細は、[第 10 章、「データベースストアの使用」](#)を参照してください。

複数ストアを作成および設定するための設定は、EMS サーバーで管理されており、クライアントに対して透過的です。複数ストア機能を設定するには、次の手順を実行してください。

1. **stores.conf** ファイルにストアをセットアップおよび設定します。

ストアは **stores.conf** ファイルに作成され、設定されます。各ストアには一意の名前が必要です。ストアはパラメータを使用して設定されます。

- ファイルベースのストアには **type** および **file** というふたつの必須パラメータがあります。これらのパラメータは、ストアがファイルベースのストアであることを決定し、その場所とファイル名を設定します。オプションのパラメータで、メッセージをファイルに書き込む方法、ファイルの最小サイズ、および EMS サーバーがファイルの切り捨てを試行するかどうかなど、他の設定を決定できます。
- **mstore** にも **type** および **file** というふたつの必須パラメータがあります。オプションのパラメータで、期限切れメッセージやパージされたメッセージを削除するスキャン間隔を設定します。間隔設定についての詳細は、後述の 32 ページの「**mstore の間隔について**」を参照してください。

2. 設定されたストアにデスティネーションを関連付けます。

メッセージは、そのデスティネーションに従って異なるストアに送信されます。デスティネーションは、**topics.conf** ファイルおよび **queues.conf** ファイルの **store** パラメータを使用して特定のストアに関連付けられます。

### ファイルベースの ストア

ファイルベースのストアを使用する場合、EMS 管理ツールの **setprop topic** または **setprop queue** コマンドを使用してストアの関連付けを動的に変更することもできます。

**mstore**

**mstore** を使用する場合、**mstore** の関連付けを設定した後にこの関連付けを動的に変更することはできません。ストアタイプ **mstore** からデスティネーションの **store** プロパティを変更するには、次の手順を実行します。

- a. EMS サーバーを停止します。
- b. デスティネーションのメッセージに関連付けられている **mstore** を空にします。
- c. **topics.conf** ファイルまたは **queues.conf** ファイルのデスティネーションの **store** プロパティを手動で編集して、ストアの関連付けを変更します。
- d. EMS サーバーを再起動します。



デスティネーションに対して **mstore** が有効になったら、**setprop** または **addprop** を使用して **store** プロパティ値を動的に変更することはできません。**store** プロパティを変更するには、サーバーを停止し、**mstore** を空にし、手動で変更を行って、再起動してください。



**mstore** は、複数のファイルにデータを格納します。**mstore** はディスク容量が不足する場合には動作することはできません。容量が不足する状況の発生を回避するため、**mstore** に格納される可能性のあるデータの最大容量の保持に必要なディスク領域の、少なくとも2倍の領域を **mstore** で利用できるようにすることをお勧めします。

同一のストアに複数のデスティネーションを明示的にまたはワイルドカードを使用してマッピングできます。ストアが設定されていない場合でも、サーバーはストアに関連付けられていない永続的メッセージをデフォルトのストアに送信します。詳細は、[デフォルトのストアファイル](#)を参照してください。

**store** パラメータの詳細は、[69 ページの「store」](#)を参照してください。

## **mstore の間隔について**

**mstore** は EMS サーバーが迅速にスタートアップできるように設計されています。この機能を使用できるようにするために、EMS サーバーがバックグラウンドで継続的にストアを監視する必要があります。サーバーは **mstore** を徐々に読み込みながら、ページされたメッセージや期限切れメッセージなどの古いデータを破棄します。

バックグラウンドアクティビティによりサーバーのパフォーマンスが低下しないように維持するには、インクリメント単位で調査を実行します。これらのインクリメント単位の長さおよび各インクリメントで処理されるデータ量は、ふたつのパラメータ設定によって制御されます。**mstore** ごとにこれらの [stores.conf](#) パラメータを設定できます。



デフォルトのパラメータ設定は、ほとんどの本番環境で最高のパフォーマンスが得られるように最適化されています（デフォルト値については、246 ページの「[mstore のパラメータ](#)」を参照）。ただし、mstore のデータ量が大幅に増加した場合は、バックグラウンドアクティビティに関連する読み取り率が EMS サーバーのメッセージ転送率に影響するようになる可能性があります。EMS サーバーのパフォーマンスが mstore のサイズによって悪影響を受ける場合は、mstore パラメータ値を調整することで、より長い期間に mstore のバックグラウンドアクティビティを分散し、関連する読み取り率を低下させることができます。

- [scan\\_target\\_interval](#) — ストアの各メッセージを調査するまでに許容される最大時間。

たとえば、scan\_target\_interval が 24 時間の場合、mstore の各セクションは少なくとも毎日 1 回は調査されます。ページされたメッセージや期限切れメッセージは、このバックグラウンドプロセスによって調査されるまで mstore から削除されません。つまり、purge コマンドの実行後、キューからメッセージが削除される（基盤のストレージ領域を再利用可能にする）までに最大 24 時間かかる場合があります。

- [scan\\_iter\\_interval](#) — バックグラウンドアクティビティのインクリメント間の時間間隔。

たとえば、scan\_iter\_interval が 10 秒の場合、EMS サーバーは 10 秒ごとに mstore の新しいセクションの調査を開始します。各インクリメント間隔で読み取られるデータ量は、ストアの合計サイズおよび scan\_target\_interval の長さによって異なります。サーバーは各間隔で十分なデータを調査し、目標の間隔内にすべてのストアを検査する必要があります。

## 例

たとえば、scan\_iter\_interval が 10 秒で、scan\_target\_interval が 1 日 (86,400 秒) で、mstore ストアに 9 GB のデータが格納されているとします。EMS サーバーは 10 秒ごとに約 1 MB のデータを調査します。この場合、平均読み取り率は約 100 KB/秒になります。これは、最新のストレージメディアでパフォーマンスの低下を発生させる可能性の低い数値とされています。

それでも EMS サーバーのパフォーマンスが低下する場合は、より長い期間にバックグラウンドアクティビティを分散するように scan\_target\_interval 値を引き上げる必要があります。show store コマンドを使用して、[Discard scan interval] に対する [Discard scan bytes] の割合を確認し、問題の設定を監視することができます。最良の結果を得るためには、各 mstore でこの割合をディスク処理容量の 20% 未満に保持する必要があります。この割合をストレージメディアの全体的なデータ転送容量との関連で検討し、バックグラウンドアクティビティがシステムのリソースを過剰に占有しないようにします。

## 統計情報への影響

`mstore` で実行されるバックグラウンド監視およびクリーンアップも主要なサーバーの統計情報の一部に影響します。最初のスキャンが完了するまで、サーバーからまだ削除されていないパージされたメッセージや期限切れメッセージの影響で一部のメッセージの統計情報が実際より少なく報告される場合があります。一部あるいはすべての `mstore` で最初のバックグラウンドスキャンが完了するまで、サーバーは正確なメッセージ数を把握していない可能性があります。

たとえば、EMS サーバーが最初に起動されると、管理ツールの `info` コマンドによって報告される [Pending Messages] および [Pending Message Size] の数値を確認できます。これは、このコマンドが発行される前にスキャンされたメッセージに関する数値のみを報告するためです。同様に、`show store` コマンドによって報告される [Message Count] および [Message Size] は、ストアに実際存在する値より少ない数値を報告する可能性があります。

最初のスキャンが完了したら、これらの数値は正確であると見なすことができます。`mstore` のスキャンステータスをチェックするには、`show store` コマンドを使用します。返される統計情報に、[First scan finished] フィールドが含まれます。このフィールドでは、前回の EMS サーバー起動時以降のスキャンステータスが報告されます。このフィールドの値が `true` の場合は、サーバーの統計情報が正確であると見なすことができます。

これらの統計情報の正しい値をより早く取得する必要がある場合は、`scan_target_interval` を引き下げる必要があります。

## メッセージの文字エンコーディング

文字エンコーディングとは、文字を表す数値セットのことです。たとえば、Latin-1 として知られる ISO 8859-1 の文字エンコーディングには、ほとんどの西ヨーロッパ言語で使用される文字と記号が含まれています。アプリケーションが送受信するメッセージに使用されている文字が英語（ASCII 文字セット）だけの場合、それ以外の文字エンコーディングを処理できるようにプログラムを変更する必要はありません。EMS サーバーとアプリケーション API によって、メッセージ内の ASCII 文字は自動的に処理されます。

ASCII 文字以外の文字（日本語など）が使用されているメッセージをアプリケーションで処理する場合は、文字セットに注意が必要です。また、クライアントはデフォルトでメッセージを UTF-8 でエンコーディングします。文字エンコーディングの中には 1 バイトで文字を表すエンコーディングがありますが、UTF-8 ではその同じ文字を表すのに 2 バイト使用することがあります。たとえば、Latin-1 はシングルスバイト文字エンコーディングです。メッセージ内のすべての文字列が、Latin-1 エンコーディングで表される文字のみを使用している場合、メッセージ内の文字列のエンコーディングに Latin-1 を指定するとシステムのパフォーマンスが向上することがあります。

EMS クライアントでは、一般的に使用されるさまざまな文字エンコーディングをメッセージ内の文字列に指定することができます。メッセージの文字エンコーディングは、メッセージ内の以下の箇所のいずれかに表示される文字列に適用されます。

- プロパティ名とプロパティ値
- `MapMessage` フィールド名とフィールド値
- メッセージの本文内のデータ

EMS のクライアント API (Java、.NET、および C) には、文字列を処理し、メッセージ内のすべての文字列に適用される文字エンコーディングを指定するメカニズムが備えられています。以下のセクションでは、EMS クライアントで使用される文字列の文字エンコーディングの影響関係について説明します。



ほとんどすべての文字セットには表示できない文字が含まれています。EMS ソフトウェアでは、表示できない文字をプログラムが使用することを制限していません。ただし、表示できない文字を含むメッセージ（ヘッダまたはデータのいずれの場合も）を EMS で表示しようとする、と、予期しない結果になることがあります。たとえば、メッセージトレース機能を有効にすると、EMS はメッセージをトレースログファイルに表示します。

## サポートされる文字エンコーディング

各メッセージには、メッセージ内の文字列のエンコーディングに使用されている文字エンコーディングの名前が含まれます。この文字エンコーディングの名前は、Java 仕様に含まれている文字エンコーディングの標準名のひとつです。標準の文字エンコーディング名のリストは、[java.sun.com](http://java.sun.com) Web サイトで参照できます。

Java と .NET クライアントは、文字エンコーディングの名前を設定または取得する際に、これらの標準の文字エンコーディング名を使用します。C クライアントには、これらの標準名に対応するマクロのリストがあります。これらのインタフェースでサポートされる文字エンコーディングのリストについては、C 言語の API リファレンスを参照してください。

## メッセージの送信

クライアントがメッセージを送信すると、メッセージにはそのメッセージ内の文字列に使用されている文字エンコーディングの名前が含まれます。Java クライアントでは、Unicode を使用して文字を表示します。エンコーディングが指定されずに Java クライアントによって作成されたメッセージには、メッセージ内のエンコーディングとして UTF-8 が使用されます。UTF-8 では、文字表示に最大 4 バイトを使用します。Java クライアントは、可能な場合、シングルバイトの文字エンコーディングを明示的に使用することで、システムのパフォーマンスを向上させることができます。

Java クライアントでは、使用するエンコーディングを `setEncoding` メソッドを使ってグローバルに設定するか、`setMessageEncoding` メソッドを使ってメッセージごとに設定することができます。これらのメソッドについての詳細は、『TIBCO Enterprise Message Service Java API リファレンス』を参照してください。

通常、C クライアントは、クライアントが実行されているマシンの文字エンコーディングを使用して文字を処理します。

## メッセージの圧縮

TIBCO Enterprise Message Service では、クライアントはメッセージの本文を圧縮してからメッセージをサーバーに送信することができます。EMS は、クライアントタイプ (Java、C、C#) 間でのメッセージの圧縮 / 解凍をサポートしています。たとえば、Java のプロデューサが圧縮したメッセージを C のコンシューマが解凍できます。



メッセージの圧縮は、Visual C++ 8 / .NET 2.0 のインストールパッケージを使用する .NET クライアントでもサポートしています。.Visual C++ 7 / .NET 1.1 パッケージの .NET クライアントでは、圧縮はサポートしていません。

### メッセージの圧縮について

メッセージの圧縮は、サーバー (永続的キューメッセージ、または持続サブスクライバがいるトピック) にメッセージを保存する場合に最適です。圧縮すれば、メッセージがサーバー上のストレージ内で使用するメモリ量が確実に減ります。圧縮されて保存されたメッセージは、圧縮された状態でサーバーによって処理されます。圧縮により、メッセージが消費するディスク領域は小さくなり、EMS サーバーによる処理も高速になります。

圧縮オプションでは、メッセージの本文だけが圧縮されます。ヘッダやプロパティが圧縮されることはありません。圧縮の使用は、メッセージ本文が大きく、メッセージがサーバー上に保存される場合に最適です。

メッセージがサーバー上に保存されない場合は、圧縮は有効ではありません。通常、圧縮には時間がかかるため、圧縮されたメッセージの送信 (またはパブリッシュ) と受信にかかる時間は、一般的に未圧縮の同じメッセージの送受信にかかる時間よりも長くなります。このため、サーバーに保存されない小さいメッセージを圧縮するメリットはほとんどありません。

### メッセージの圧縮の設定

メッセージの圧縮は個々のメッセージに対して指定します。つまり、メッセージの圧縮はメッセージ単位で行えます。TIBCO Enterprise Message Service では、メッセージの圧縮をトピック単位またはキュー単位で設定する方法はありません。

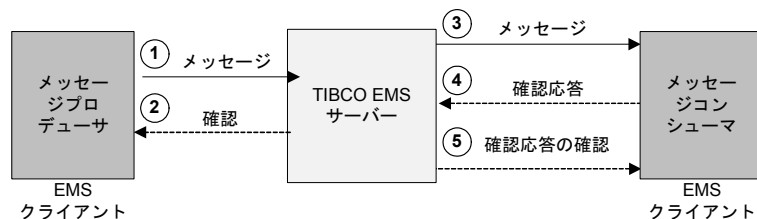
メッセージの圧縮を設定するには、メッセージを送信またはパブリッシュするアプリケーションが、メッセージを送信またはパブリッシュする前にメッセージのプロパティにアクセスし、ブール値プロパティ `JMS_TIBCO_COMPRESS` を `true` に設定する必要があります。

圧縮されたメッセージは透過的に処理されます。クライアントコードは、`JMS_TIBCO_COMPRESS` プロパティだけを設定します。クライアントはその他のアクションを実行する必要はありません。クライアントは、受信したすべての圧縮されたメッセージを自動的に解凍します。

## メッセージの確認応答

JMS のインタフェース仕様では、多くの場合（すべての場合ではない）において、メッセージの配信が確実に行われることが要求されます。図 10 は、メッセージ配信と確認応答の基本構造を示しています。

図 10 メッセージ配信と確認応答



メッセージ配信と確認応答の手順は、以下のとおりです。

1. メッセージはメッセージプロデューサから、EMS サーバーが配置されているマシンに送信されます。
2. 永続的メッセージの場合、EMS サーバーはメッセージ受信の確認メッセージをメッセージプロデューサに送信します。
3. サーバーはメッセージをコンシューマに送信します。
4. コンシューマは、メッセージ受信の確認応答をサーバーに送信します。特定の確認応答モードのセッションを設定して、コンシューマとサーバー間の確認応答の処理方法を指定することができます。これらの確認応答モードについては、以下で説明します。
5. 多くの場合、サーバーは確認応答の確認メッセージをコンシューマに送信します。

JMS 仕様では、以下の 3 つのレベルで、非トランザクションセッションの確認応答モードが定義されています。

- **CLIENT\_ACKNOWLEDGE** モードでは、セッションでそれまで配信されたすべてのメッセージに対してコンシューマが確認応答することが指定されます。このモードを使用すると、コンシューマによるメッセージ処理が遅延し、確認応答されないメッセージが大量に蓄積されることがあります。
- **AUTO\_ACKNOWLEDGE** モードでは、メッセージ処理が完了すると、コンシューマによるメッセージ受信をセッションが自動的に確認応答することが指定されます。
- **DUPS\_OK\_ACKNOWLEDGE** モードでは、コンシューマへのメッセージ配信についてセッションが確認応答を「遅延」することが指定されます。「遅延」とは、コンシューマが、都合のよいときまでサーバーへのメッセージ確認応答を遅らせるということです。この間にサーバーはメッセージを再配信することが

あります。このモードでは、セッションのオーバーヘッドが低減されます。JMS に障害があった場合、コンシューマはメッセージを重複して受け取ることがあります。

EMS には、JMS の確認応答モードを拡張した以下のモードが用意されています。

- [NO\\_ACKNOWLEDGE](#)
- [EXPLICIT\\_CLIENT\\_ACKNOWLEDGE](#)
- [EXPLICIT\\_CLIENT\\_DUPS\\_OK\\_ACKNOWLEDGE](#)

確認応答モードは、セッションが作成される際に設定されます (328 ページの「[セッションの作成](#)」を参照)。

## NO\_ACKNOWLEDGE

NO\_ACKNOWLEDGE モードは、メッセージ受信の確認応答を停止します。サーバーがクライアントにメッセージを送信すると、そのコンシューマに対するそのメッセージに関するすべての情報がサーバーから削除されます。したがって、クライアントアプリケーションは、メッセージ受信の確認応答をサーバーに送信する必要がありません。確認応答を送信しなければ、メッセージトラフィックが減少し、レシーバが使う時間が節約されるので、システムリソースが有効に活用できます。



確認応答なし受信モードで作成されるセッションでは、持続サブスクライバを作成することはできません。

また、別のサーバーからルーティングされているキューのキューレシーバに、NO\_ACKNOWLEDGE モードを指定することはできません。

## EXPLICIT\_CLIENT\_ACKNOWLEDGE

EXPLICIT\_CLIENT\_ACKNOWLEDGE モードの機能は、CLIENT\_ACKNOWLEDGE と同様ですが、セッションでそれまで受信されたすべてのメッセージに対してではなく、メッセージごとに確認応答する点が異なります。

EXPLICIT\_CLIENT\_ACKNOWLEDGE モードが使用される例としては、受信したメッセージをデータとしてデータベースに保存する場合です。データベースへのデータ挿入処理に時間がかかる場合は、複数のアプリケーションスレッドを使用して、データ挿入をすべて同時に行うことができます。挿入を完了したスレッドは、EXPLICIT\_CLIENT\_ACKNOWLEDGE モードを使用して、現在実行中のメッセージに対してのみ確認応答することができます。

## EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE

EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE モードの機能は、DUPS\_OK\_ACKNOWLEDGE と同様ですが、セッションでそれまで受信されたすべてのメッセージに対してではなく、メッセージごとに確認応答を「遅延」する点が異なります。



## メッセージセクタ

メッセージセクタはクライアントプログラムがメッセージのセットを指定できるようにする文字列で、メッセージヘッダとメッセージプロパティの値に基づきます。セクタ文字列にメッセージからヘッダとプロパティの値を代入した後に文字列が `true` と評価された場合、セクタはメッセージに一致します。コンシューマは、セクタに一致するメッセージのみをサーバーが配信するように要求できます。

セクタの構文は、SQL92 条件式構文のサブセットに基づきます。

### 識別子

識別子はメッセージヘッダとメッセージプロパティの値を参照できますが、メッセージ本文は参照できません。識別子は太文字と小文字を区別します。

**基本構文** 識別子は、先頭が文字で任意の長さの文字と数字のシーケンスです。Java と同様、文字のセットには `_` (アンダースコア) と `$` (ドル) が含まれます。

**不正** 例外として、特定の名前は識別子として使用できません。特に、`NULL`、`TRUE`、`FALSE`、`NOT`、`AND`、`OR`、`BETWEEN`、`LIKE`、`IN`、`IS`、および `ESCAPE` は、メッセージセクタ構文では特別な意味を持ちます。

**値** 識別子は、メッセージヘッダ名またはメッセージプロパティ名のいずれかを参照します。メッセージセクタの識別子の型は、ヘッダまたはプロパティの値の型に対応します。メッセージ内に存在しないヘッダまたはプロパティを識別子が参照した場合、識別子の値は `NULL` になります。

### リテラル

**文字列リテラル** 文字列リテラルは一重引用符で囲まれます。リテラル内で一重引用符を表すには、`'literal's'` のようにふたつの一重引用符を使用します。文字列リテラルでは、Unicode 文字エンコーディングを使用します。文字列リテラルは太文字と小文字を区別します。

**真数リテラル** 真数リテラルは、小数点を含まない数値 (`57`、`-957`、`+62` など) です。long の範囲の数値がサポートされています。

**概数リテラル** 概数リテラルは、小数点を含む数値 (`7.`、`-95.7`、`+6.2` など)、または科学的表記法を使用した数値 (`7E3`、`-57.9E2` など) です。double の範囲の数値がサポートされています。概数リテラルは、Java プログラミング言語の浮動小数点リテラル構文を使用します。

**ブール値リテラル**      ブール値リテラルは TRUE と FALSE で、大文字と小文字を区別します。

式の値の内部計算では、SQL に似た 3 値ブール論理を使用します。ただし、式の最終値は常に TRUE または FALSE のいずれかで、UNKNOWN になることはありません。

## 式

**式としてのセ  
レクタ**      すべてのセレクタは条件式で表されます。true と評価されるセレクタはメッセージに一致し、false または unknown と評価されるセレクタはメッセージに一致しません。

**算術式**      算術式は、数値リテラル、識別子（数値リテラルとして評価）、算術演算、より小さい算術式で構成されます。

**条件式**      条件式は、比較演算、論理演算、より小さい条件式で構成されます。

**評価の順序**      評価の順序は、優先順位レベル内の左から右です。かっこの順序よりも優先されます。

## 演算子

**大文字と小文字を  
区別しない**      演算子名は大文字と小文字を区別しません。

**論理演算子**      論理演算子（優先順位順）：NOT、AND、OR

**比較演算子**      比較演算子：=、>、>=、<、<=、<>（不等号）

これらの演算子は、比較可能な型の値のみを比較できます（真数値と概数値は比較可能な型です）。比較できない型を比較しようとする場合、false になります。比較するいずれかの値が NULL と評価された場合、結果は SQL の 3 値論理での unknown となります。

文字列の値の比較は、= と <> のみに制限されています。ふたつの文字列は、同じ文字のシーケンスを含む場合に限り等しくなります。

ブール値の比較は、= と <> のみに制限されています。

**算術演算子**      算術演算子（優先順位順）

- +、-（単項）
- \*、/（乗算および除算）
- +、-（加算および減算）

算術演算は、Java プログラミング言語の数値昇格ルールに従います。

**BETWEEN** *arithmetic-expr1* [NOT] BETWEEN *arithmetic-expr2* AND *arithmetic-expr3*

#### 演算子

BETWEEN 比較演算子にはその終点が含まれます。例：

- `age BETWEEN 5 AND 9` は `age >= 5 AND age <= 9` と同等
- `age NOT BETWEEN 5 AND 9` は `age < 5 OR age > 9` と同等

#### 文字列 メンバーシップ の設定

*identifier* [NOT] IN (*string-literal1*, *string-literal2*, ...)

*identifier* は文字列または NULL として評価される必要があります。NULL の場合、この式の値は **unknown** です。

#### パターン一致

*identifier* [NOT] LIKE *pattern-value* [ESCAPE *escape-character*]

*identifier* は文字列として評価される必要があります。

*pattern-value* は文字列リテラルで、一部の文字は特別な意味を持ちます。

- `_` (アンダースコア) は、任意の 1 文字に一致します。
- `%` (パーセント) は、任意の 0 文字以上のシーケンスに一致します。
- いずれかの特殊文字の前の *escape-character* は、特殊文字を通常文字 (その文字自体に一致) に変更します。

#### NULL ヘッダまたはプロパティ

*identifier* IS NULL

この比較演算子は、メッセージヘッダが NULL かどうか、またはメッセージプロパティが指定されていないかどうかをテストします。

*identifier* IS NOT NULL

この比較演算子は、メッセージヘッダまたはメッセージプロパティが NULL 以外かどうかをテストします。

#### 空白

空白は、空白文字、水平タブ、フォームフィード、行ターミネータのいずれか、またはこのセットの連続する文字列です。

# データ型の変換

表 7 に適正なデータ型の変換を示します。表 7 の記号 X は、その行の型としてメッセージ内に書き込まれた値をその列の型として抽出できることを示します。この表は、マップペア、ヘッダ、およびプロパティを含むすべてのメッセージの値に適用されますが、以下に示す例外があります。

表 7 データ型の変換

	bool	byte	short	char	int	long	float	double	string	byte[]
bool	X								X	
byte		X	X		X	X			X	
short			X		X	X			X	
char				X					X	
int					X	X			X	
long						X			X	
float							X	X	X	
double								X	X	
string	X	X	X		X	X	X	X	X	
byte[]										X

- メモ
- メッセージプロパティにはバイト配列の値は指定できません。
  - 文字列として書き込まれた値は、文字列をその型の数値として解析できる場合にのみ数値またはブール型として抽出できます。

## 同期および非同期のメッセージコンシューム

---

EMS の API では、同期および非同期のメッセージコンシュームが可能です。同期コンシュームの場合、メッセージコンシューマはトピックまたはキューに対して明示的に `receive` 呼び出しを行います。メッセージの同期受信を行うと、メッセージが届くまでコンシューマはブロックされた状態になります。詳細は、344 ページの「[メッセージの受信](#)」を参照してください。

コンシューマがメッセージを非同期で受信するには、メッセージを受信する「メッセージリスナー」を登録します。メッセージがデスティネーションに届くと、メッセージリスナーはメッセージをメッセージコンシューマに配信します。メッセージコンシューマは、メッセージ間の他の処理も実行することができます。詳細は、337 ページの「[非同期でメッセージをコンシュームするメッセージリスナーの作成](#)」を参照してください。



## 第3章

# デスティネーション

この章では、デスティネーション（トピックとキュー）について説明します。

### トピック

---

- [「デスティネーションの概要」](#)、48 ページ
- [「デスティネーションのプロパティ」](#)、55 ページ
- [「デスティネーションの作成と変更」](#)、71 ページ
- [「ワイルドカード」](#)、73 ページ
- [「継承」](#)、76 ページ
- [「デスティネーションブリッジ」](#)、78 ページ
- [「フロー制御」](#)、83 ページ

## デスティネーションの概要

メッセージのデスティネーションは、トピックまたはキューのいずれかになります。デスティネーションは、サーバーの設定ファイルを設定して静的に作成することができます。また、クライアントアプリケーションから動的に作成することもできます。

ルートによって接続されたサーバー間では、テンポラリトピックに送信されたメッセージを交換します。この結果、テンポラリトピックは、要求応答における応答メッセージの理想的なデスティネーションとなります。

表 8 に、静的、動的、およびテンポラリの各デスティネーションの違いの概要を示します。詳しくは、この表の後のセクションで説明します。

表 8 デスティネーションの概要 (1 / 2 シート)

様態	静的	動的	テンポラリ
用途	静的デスティネーションでは、管理者は、エンタープライズレベルでの EMS の動作を設定できます。これらの管理対象オブジェクトは管理者によって定義されます。クライアントプログラムはその定義に従って、設定を担当する開発者およびエンドユーザーにプログラムを配信します。	動的デスティネーションでは、クライアントプログラムは、短期間の使用に必要とされるデスティネーションを自由に定義できます。	テンポラリデスティネーションは、応答サブジェクトなどの限られた範囲の使用に適切です。
配信の範囲	静的デスティネーションは、同時使用が可能です。クライアントの複数プロセス（および 1 プロセス内の複数スレッド）が、デスティネーションを示す複数のローカルオブジェクトを作成し、そのデスティネーションからメッセージをコンシュームできます。	動的デスティネーションは、同時使用が可能です。クライアントの複数プロセス（および 1 プロセス内の複数スレッド）が、デスティネーションを示す複数のローカルオブジェクトを作成し、そのデスティネーションからメッセージをコンシュームできます。	テンポラリデスティネーションは、ローカルでのみ使用が可能です。テンポラリデスティネーションを作成したクライアント接続だけが、そのデスティネーションからメッセージをコンシュームできます。 ただし、ルートによって接続されたサーバー間では、テンポラリトピックに送信されたメッセージを交換します。
作成	静的デスティネーションは、管理者が EMS サーバーの管理ツールまたは API を使って作成します。	動的デスティネーションは、クライアントプログラムによって作成されます (サーバーの設定で許可されている場合)。	テンポラリデスティネーションは、クライアントプログラムによって作成されます。



表 8 デスティネーションの概要 (2 / 2 シート)

様態	静的	動的	テンポラリ
照合	静的デスティネーションの照合は、クライアントプログラムによって名前で行われます。照合に成功すると、デスティネーションのローカル・オブジェクト表示を返します。	適用なし。	適用なし。
期間	静的デスティネーションは、管理者によって明示的に削除されるまでサーバーに存在します。	<p>動的デスティネーションは、少なくともひとつのクライアントがそれを使用している限り、サーバーに存在します。以下のすべて条件が満たされると、動的デスティネーションはサーバーから自動的に(都合のよいときに)削除されます。</p> <ul style="list-style-type: none"> <li>• <b>トピックまたはキュー:</b> デスティネーションにアクセスするクライアントプログラムの接続がすべて切断している</li> <li>• <b>トピック:</b> オフラインになっているトピックの持続サブスクライバがない</li> <li>• <b>キュー:</b> キューに保存されているメッセージがない</li> </ul>	テンポラリデスティネーションは、それを作成したクライアントによって明示的に削除されるか、クライアントがサーバーとの接続を切断するまで、サーバーに存在します。

## デスティネーション名

デスティネーション名の文字列は、ドット (.) で区切られたいくつかの要素で構成されます。ドットを使って、複数の部分からなるデスティネーション名を作成して、デスティネーションを区分的にすることができます。

例として、複数のデスティネーションにメッセージをパブリッシュする会計アプリケーションが挙げられます。アプリケーションはすべてのメッセージに、ACCT のプリフィクスを付けます。名前の各要素で、アプリケーションの各コンポーネントを指定できます。ACCT.GEN\_LEDGER.CASH、ACCT.GEN\_LEDGER.RECEIVABLE、および ACCT.GEN\_LEDGER.MISC は、アプリケーションの一般元帳部分のサブジェクトになります。

サブジェクト名を複数の要素に分けておくと、アプリケーションはワイルドカードを使用して複数のサブジェクトを指定できます。詳細は、73 ページの「[ワイルドカード](#)」を参照してください。デスティネーション名にワイルドカードを使用して、「親」のデスティネーションと「子」のデスティネーションとの関係を定義することもできます。この場合、子のデスティネーションは親のプロパティを継承します。76 ページの「[プロパティの継承](#)」を参照してください。

## 静的デスティネーション

静的デスティネーションの設定情報は、EMS サーバーの設定ファイルに保存されています。設定情報は、さまざまな方法で変更できます。静的デスティネーションを管理するには、テキストエディタを使用して設定ファイルを編集します。または管理ツールや管理 API を使用することもできます。

クライアントは、JNDI や LDAP などの名前付けサービスを使用して、静的デスティネーションへの参照を取得します。クライアントによる静的デスティネーションの使用方法についての詳細は、71 ページの「[デスティネーションの作成と変更](#)」を参照してください。

## 動的デスティネーション

動的デスティネーションは、クライアントアプリケーションの要求に応じて、EMS サーバーによってその場（オンザフライ）で作成されます。動的デスティネーションは設定ファイルの要素にはなく、デスティネーションにメッセージまたはコンシューマがある限り存在します。クライアントは、動的キューおよびトピックの参照に JNDI を使用できません。

管理ツールで [show queues](#) または [show topics](#) コマンドを使用すると、名前の前にアスタリスク（\*）が付いた動的トピックまたはキューがリストに表示されます。キューまたはトピックのプロパティで名前の前にアスタリスク（\*）が表示されている場合、そのプロパティは親のキューまたはトピックから継承されており、変更できないことを意味します。

EMS サーバーで動的にトピックまたはキューを作成する方法についての詳細は、331 ページの「[トピックとキューの動的作成](#)」を参照してください。

## テンポラリデスティネーション

TIBCO Enterprise Message Service は、JMS 仕様 1.1 と API で定義されているテンポラリデスティネーションをサポートしています。

ルートによって接続されたサーバー間では、テンポラリトピックに送信されたメッセージを交換します。この結果、テンポラリトピックは、要求応答における応答メッセージの理想的なデスティネーションとなります。

テンポラリキューとトピックについての詳細は、xxviii ページの「[サードパーティ製マニュアル](#)」に記載されている JMS 仕様書を参照してください。

## デスティネーションブリッジ

同じまたは異なるタイプのデスティネーション間にサーバーベースのブリッジを作成することで、混合メッセージングモデルのアプリケーションを作成することができます。これにより、ひとつのデスティネーションに配信したすべてのメッセージが、ブリッジされているデスティネーションにも送信されます。ブリッジは、異なるタイプのデスティネーション間、同じタイプのデスティネーション間、または複数のデスティネーションに対して作成することができます。たとえば、キューとトピックの間にブリッジを作成したり、あるトピックから別のトピックへブリッジを作成したりすることができます。

デスティネーションのブリッジについての詳細は、78 ページの「[デスティネーションブリッジ](#)」を参照してください。

## デスティネーション名の構文

TIBCO Enterprise Message Service では、デスティネーション名の構文と解釈にいくつかの制限があります。システムの設計者や開発者は、デスティネーション名に対する独自の命名規則を自由に定めることができます。アプリケーション独自のデータ構造を反映するデスティネーション名が最適です。

**構造** デスティネーション名の文字列は、ドット (.) で区切られたいくつかの要素で構成されます。ドットを使って、複数の部分からなるデスティネーション名を作成して、デスティネーションを区分することができます。

空の文字列 ("" ) はデスティネーション名には使用できません。また、エスケープシーケンスを使用してドットを要素に組み込むことはできません。

タブ、空白、および表示できない文字は禁止されていませんが、デスティネーション名には使用しないことをお勧めします。ただし、ワイルドカードは使用できます。詳細は、73 ページの「ワイルドカード」を参照してください。

**長さ** デスティネーションの長さは合計 249 文字に制限されています。ただし、この長さの一部は内部使用のために予約されています。内部で使用するために予約されている長さは、デスティネーション内の要素数によって異なります。最大要素数を含むデスティネーションは、196 文字に制限されています。

各デスティネーションには最大 64 要素を含めることができます。要素の合計は 127 文字以内にする必要があります。ドットの区切り文字は要素の長さには含まれません。

デスティネーション名のパフォーマンスに関する注意事項

デスティネーション名の命名規則を定める場合、以下のパフォーマンスに関する注意事項を確認します。

- 短いデスティネーション名は、長いデスティネーション名よりもパフォーマンスが高くなります。
- 複数の短い要素を含むデスティネーションは、ひとつの長い要素を含むデスティネーションよりもパフォーマンスが高くなります。
- 要素リストの最初の方が異なるデスティネーションセットは、最後の要素のみが異なるデスティネーションセットよりもパフォーマンスが高くなります。

## デスティネーション名での特殊文字

以下の文字は、デスティネーション名で使用される場合は特別な意味を持ちます。

表 9 デスティネーション名で特別な意味を持つ文字

文字	文字名	特別な意味
_	アンダースコア	アンダースコアで始まるデスティネーション名は予約されています。アプリケーションプログラムでは、最初の要素の最初の文字がアンダースコアのデスティネーションを送信できません。ただし、_INBOX は例外です。  デスティネーション名の他の位置でアンダースコアを使用できます。
.	ドット	デスティネーション名に含まれる要素を区切ります。
>	大なり	ワイルドカード文字です。後続のひとつ以上の要素に一致します。
*	アスタリスク	ワイルドカード文字です。ひとつの要素に一致します。

ワイルドカードによる一致についての詳細は、73 ページの「[ワイルドカード\\*と>](#)」を参照してください。

## 例：

以下の例は、デスティネーション名の構文を示しています。

表 10 有効なデスティネーション名の例

NEWS.LOCAL.POLITICS.CITY_COUNCIL
NEWS.NATIONAL.ARTS.MOVIES.REVIEWS
CHAT.MRKTG.NEW_PRODUCTS
CHAT.DEVELOPMENT.BIG_PROJECT.DESIGN
News.Sports.Baseball
finance
This.long.subject_name.is.valid.even.though.quite.uninformative

表 11 無効なデスティネーション名の例

News..Natural_Disasters.Flood (NULL 要素)
WRONG. (NULL 要素)
.TRIPLE.WRONG.. (三つの NULL 要素)
News.Tennis.Stats.Roger¥.Federer (要素 Roger のバックスラッシュは要素名に含まれ、ドットはエスケープされない)

## デスティネーションのプロパティ

このセクションでは、トピックとキューのプロパティを説明します。

`topics.conf` または `queues.conf` ファイルに直接プロパティを設定するか、EMS 管理ツールの `setprop topic` または `setprop queue` コマンドを使ってプロパティを設定できます。

表 12 に、トピックまたはキューに割り当て可能なプロパティを示します。各プロパティについては、この表の後のセクションで説明します。

表 12 デスティネーションのプロパティ (1 / 2 シート)

プロパティ	説明のあるページ	トピック	キュー
<code>channel</code>	56	はい	いいえ
<code>exclusive</code>	56	いいえ	はい
<code>expiration</code>	57	はい	はい
<code>export</code>	58	はい	いいえ
<code>flowControl</code>	58	はい	はい
<code>global</code>	59	はい	はい
<code>import</code>	59	はい	はい
<code>maxbytes</code>	60	はい	はい
<code>maxmsgs</code>	61	はい	はい
<code>maxRedelivery</code>	61	いいえ	はい
<code>overflowPolicy</code>	62	はい	はい
<code>prefetch</code>	64	はい	はい
<code>redeliveryDelay</code>	66	いいえ	はい
<code>secure</code>	67	はい	はい
<code>sender_name</code>	68	はい	はい
<code>sender_name_enforced</code>	68	はい	はい
<code>store</code>	69	はい	はい

表 12 デスティネーションのプロパティ (2 / 2 シート)

プロパティ	説明のあるページ	トピック	キュー
<a href="#">trace</a>	<a href="#">70</a>	はい	はい

channel

`channel` プロパティは、このトピックに送信されるメッセージのブロードキャストに使用されるマルチキャストチャンネルを決定します。`channel` プロパティを組み込むと、関連付けられたトピックをマルチキャストできるようになります。

`channel` プロパティは、次の形式で設定します。

```
channel=name
```

`name` には、`channels.conf` ファイルで定義されている `channel` の名前を指定します。

たとえば、次のように指定すると、`topic.foo` というトピックに送信されたすべてのメッセージが `mycast` という名前のチャンネルを介してブロードキャストされます。

```
topic.foo channel=mycast
```

トピックごとにひとつのチャンネルのみを使用できます。このため、重複する複数のワイルドカードトピックを `channel` プロパティとともに使用することはできません。あるワイルドカードトピックに `channel` プロパティを指定し、それと重複するワイルドカードトピックに `channel` プロパティを指定して作成しようとしても失敗します。詳細は、73 ページの「[重複するワイルドカードと独立プロパティ](#)」を参照してください。



このパラメータを使用するには、最初に `channels.conf` ファイルにマルチキャストチャンネルを設定し、この機能を `tibemsd.conf` ファイルで有効にする必要があります。詳細は、357 ページの第 13 章、「[マルチキャストの使用方法](#)」を参照してください。

exclusive

`exclusive` プロパティは（トピックではなく）キューでのみ使用できます。また、グローバルキューとともに使用することはできません。

キューの `exclusive` パラメータが設定されている場合、サーバーはそのキュー内のすべてのメッセージをひとつのコンシューマにのみ送信します。他のコンシューマは、キューからメッセージを受信できません。ただし、これらの他のコンシューマは「スタンバイ」状態になっているため、プライマリコンシューマへの送信が失敗した場合、サーバーはスタンバイ状態になっているコンシューマのひとつをプライマリコンシューマとして選択し、そのコンシューマへのメッセージ配信を開始します。

`exclusive` プロパティは、次の形式で設定します。

```
exclusive
```



## 非排他キューとラウンドロビン型配信

デフォルトでは、キューに **exclusive** は設定されておらず、サーバーはラウンドロビン方式で、受信の準備ができたレシーバから順番にメッセージを配信していきます。新しいメッセージを受信可能なレシーバがまだある場合、サーバーは次のサイクルで、受信の準備ができたレシーバから順番にメッセージを配信していきます。メッセージを受信可能なレシーバがなくなったら、サーバーは、キューレシーバからメッセージの受信が可能になったことを通知されるまで待機します。

この仕組みにより、大量のメッセージがひとつのレシーバに集中するのを避け、キューレシーバのセット全体で受信メッセージの負荷を分散します。

## expiration

デスティネーションに **expiration** プロパティを設定すると、サーバーは無効化の期限に従って、**expiration** プロパティによって指定されている期間、メッセージを保持します。

ただし、サーバーは、プロデューサが **0** 値を使用してメッセージヘッダに設定した **JMSExpiration** 値を無効にするため、コンシュームするクライアントはメッセージを期限切れにしません。

キューとトピックの **expiration** プロパティは、次の形式で設定します。

```
expiration=time[msec|sec|min|hour|day]
```

**time** には、秒数を指定します。ゼロの値は、デスティネーションへ配信するメッセージに有効期限を設定しない、特別な場合に使用します。

**time** 値の単位として、**msec** (ミリ秒)、**sec** (秒)、**min** (分)、**hour** (時間)、または **day** (日数) を指定できます。例：

```
expiration=10min
```

10 分が指定されます。

メッセージの有効期限が切れると、メッセージは破棄されるか、メッセージの **JMS\_TIBCO\_PRESERVE\_UNDELIVERED** プロパティに **true** が設定されている場合は、限定されたコンシューマが処理できるようメッセージは未配信キューに入れます。詳細は、20 ページの「[未配信メッセージキュー](#)」を参照してください。

EMS バージョン 4.4 以降では、クライアントのクロックは接続の作成時にサーバーと自動同期されます。ただし、長期間にわたる接続では、ネットワークタイムプロトコル (NTP: Network Time Protocol) がサーバーとクライアントの間の継続的な同期を維持する最も信頼性の高い方法です。また、EMS サーバーまたはクライアントアプリケーションが EMS バージョン 4.4 以前のバージョンに基づいている場合、クライアントアプリケーションのメッセージの有効期限にゼロ以外の値が設定されているときは、メッセージを送受信するすべてのホスト間でクロックが同期していることを確認する必要があります。クロックの許容誤差を、最も短いメッセージの有効期限と比べて非常に小さい値に抑えて同期させます。

## export

`export` プロパティを設定すると、クライアントがトピックにパブリッシュしたメッセージを外部システムにエクスポートすることができます。ただし、外部システムへのトランスポートが設定されている必要があります。

`export` プロパティは、次の形式で設定します。

```
export="list"
```

`list` には、`transports.conf` ファイルの `[transport_name] ids` で指定されているトランスポート名をひとつ以上指定します。複数のトランスポート名を列挙して指定する場合は、コンマで区切ります。

例：

```
export="RV1,RV2"
```

これで、`SmartSockets` または `Rendezvous` の信頼メッセージプロトコルと保証メッセージプロトコルのトランスポートが設定されます。`export` プロパティには、同じタイプのひとつ以上のトランスポートの名前を指定できます。

トピックの `export` プロパティを設定するには、外部システム (`Rendezvous` など) を購入してインストールし、設定する必要があります。また、外部システムに通信パラメータを設定するために、`transports.conf` ファイルでトランスポート名を作成する必要があります。

外部メッセージサービスについての詳細は、以下の章を参照してください。

- 389 ページの [第 15 章、「TIBCO Rendezvous との相互運用」](#)
- 413 ページの [第 16 章、「TIBCO SmartSockets との相互運用」](#)

## flowControl

`flowControl` プロパティを指定して、デスティネーションの保留中のメッセージを保存するためにサーバーが使用できる最大サイズを設定できます。メッセージ数がこの最大値を超えると、サーバーはプロデューサがメッセージを送信する速度を、コンシューマがメッセージを受信できる速度まで落とします。メッセージコンシューマがメッセージをコンシュームする速度よりもメッセージプロデューサがメッセージを送信する速度が速い場合に、この機能は有用です。`overflowPolicy` プロパティの設定による動作とは異なり、`flowControl` プロパティでは、メッセージは破棄されず、メッセージプロデューサにエラーが返されません。

`flowControl` プロパティは、次の形式で設定します。

```
flowControl=size[KB|MB|GB]
```

`size` には、デスティネーションに保留されるメッセージの最大保存バイト数を指定します。値を指定せずに `flowControl` プロパティを指定した場合、最大バイト数（目標値）は 256KB に設定されます。

数字の後に KB（キロバイト）、MB（メガバイト）、または GB（ギガバイト）の単位を指定することもできます。例：

```
flowControl=1000KB
```

1000 キロバイトを意味します。

このプロパティの値がサーバーに反映されるようにするには、[tibemsd.conf](#) ファイルの `flow_control` パラメータを `enabled` に設定する必要があります。フロー制御についての詳細は、83 ページの「[フロー制御](#)」を参照してください。

## global

`global` プロパティが設定されているトピックまたはキューに送信されたメッセージは、そのサーバーのルーティングに参加している他のサーバーにルーティングされます。

`global` プロパティは、次の形式で設定します。

```
global
```

サーバー間のルーティングについての詳細は、497 ページの第 20 章、「[ルートの使用](#)」を参照してください。

## import

`import` プロパティを設定すると、外部システムによってパブリッシュされたメッセージを EMS デスティネーション（トピックまたはキュー）で受信することができます。ただし、外部システムへのトランスポートが設定されている必要があります。

`import` プロパティは、次の形式で設定します。

```
import="list"
```

`list` には、[transports.conf](#) ファイルの `[NAME] ids` で指定されているトランスポート名をひとつ以上指定します。複数のトランスポート名を列挙して指定する場合は、コンマで区切ります。例：

```
import="RV1,RV2"
```

これで、TIBCO SmartSockets または TIBCO Rendezvous の信頼メッセージプロトコルと保証メッセージプロトコルのトランスポートが設定されます。`import` プロパティには、同じタイプのひとつ以上のトランスポートの名前を指定できます。

トピックの `import` プロパティを設定するには、外部システム（Rendezvous など）を購入してインストールし、設定する必要があります。また、外部システムに通信パラメータを設定するために、[transports.conf](#) ファイルでトランスポート名を作成する必要があります。

外部メッセージサービスについての詳細は、以下の章を参照してください。

- 389 ページの第 15 章、「[TIBCO Rendezvous との相互運用](#)」
- 413 ページの第 16 章、「[TIBCO SmartSockets との相互運用](#)」

## maxbytes

トピックとキューの `maxbytes` プロパティは、次の形式で設定します。

```
maxbytes=value[KB|MB|GB]
```

`value` には、バイト数を指定します。例：

```
maxbytes=1000
```

1000 バイトが指定されます。

数字の後に KB（キロバイト）、MB（メガバイト）、または GB（ギガバイト）の単位を指定することもできます。例：

```
maxbytes=1000KB
```

1000 キロバイトを意味します。

キューの場合、`maxbytes` はキューに保存できる、キュー内のすべてのメッセージの合計の最大量（バイト）を定義します。この最大量を超えると、メッセージはサーバーに拒否され、メッセージプロデューサへの `send` 呼び出しがエラーを返します（[overflowPolicy](#) も参照）。たとえば、レシーバが長い時間オフラインになっていると、キューのサイズがこの最大量に達し、新しいメッセージにメモリを割り当てることができなくなることがあります。

`maxbytes` パラメータが設定されていない場合、またはゼロに設定されている場合、サーバーはキューのメモリ割り当てを制限しません。

同じキューに対して、`maxmsgs` と `maxbytes` の両方のプロパティを設定できます。いずれかのプロパティの設定値を超えた場合、コンシューマがキューのサイズをこれらの設定値未満に減らさない限り、サーバーは新しいメッセージを受け付けません。

トピックの場合、`maxbytes` は、そのトピックの各持続または非持続オンラインサブスクライバに配信するために、トピックが保存できる最大量（バイト）を制限します。つまり、この制限は、トピックの各サブスクライバに別々に適用されます。たとえば、持続サブスクライバが長い時間オフラインになっている場合、保留中のメッセージが `maxbytes` の値を超えるまで、メッセージがトピックに蓄積されます。すなわち、サブスクライバがメッセージをコンシュームすると（ストレージを解放すると）、トピックはサブスクライバ宛ての新しいメッセージを受け入れます。非持続サブスクライバの場合、`maxbytes` はサブスクライバがオンラインである間に蓄積できる保留中のメッセージの数を制限します。



特定の状況では、メッセージがパイプラインで処理されることやトランザクションメッセージの要件によって、`maxbytes` の制限をわずかに超過する可能性があります。メッセージの合計数が、設定された制限値をわずかに上回る場合があります。

デスティネーションが複数の親デスティネーションから、異なる値のプロパティを継承する場合は、一番小さな値が継承されます。



パラメータ `disconnect_non_acking_consumers` を使用して、メッセージの確認応答を行わずにメッセージを受け取るコンシューマからさらに保護することができます。

## maxmsgs

トピックとキューの **maxmsgs** プロパティは、次の形式で設定します。

**maxmsgs=***value*

*value* には、キュー内で待機可能なメッセージの最大数を指定します。メッセージの追加によってこの制限値を超えると、サーバーはストレージに保存されるメッセージを受け付けず、メッセージプロデューサの **send** 呼び出しがエラーを返します (**overflowPolicy** も参照)。

**maxmsgs** パラメータが設定されていない場合、またはゼロに設定されている場合、サーバーはキュー内のメッセージの数を制限しません。

同じキューに対して、**maxmsgs** と **maxbytes** の両方のプロパティを設定できます。いずれかのプロパティの設定値を超えた場合、コンシューマがキューのサイズをこれらの設定値未満に減らさない限り、サーバーは新しいメッセージを受け付けません。



特定の状況では、メッセージがパイプラインで処理されることやトランザクションメッセージの要件によって、**maxmsgs** の制限をわずかに超過する可能性があります。メッセージの合計数が、設定された制限値をわずかに上回る場合があります。



パラメータ **disconnect\_non\_acking\_consumers** を使用して、メッセージの確認応答を行わずにメッセージを受け取るコンシューマからさらに保護することができます。

## maxRedelivery

**maxRedelivery** プロパティを指定して、キューに送信されたメッセージの配信をサーバーが試みる回数を設定できます。**maxRedelivery** プロパティは、次の形式で設定します。

**maxRedelivery=***count*

*count* には、2 ～ 255 の整数を入力し、メッセージがレシーバに配信されるまでの最大試行回数を指定します。ゼロの値を指定すると **maxRedelivery** が無効になり、無限に試行を繰り返します。

指定した回数だけサーバーがメッセージの配信を試行すると、メッセージは破棄されるか、メッセージの **JMS\_TIBCO\_PRESERVE\_UNDELIVERED** プロパティに **true** が設定されている場合は、限定されたコンシューマが処理できるようメッセージは未配信キューに入れられます。詳細は、20 ページの「**未配信メッセージキュー**」を参照してください。

再配信されたメッセージの **JMSRedelivered** ヘッダプロパティに **true** が設定され、メッセージがキューに配信された回数が **JMSXDeliveryCount** プロパティに設定されます。サーバーが再起動した場合、その時点での **JMSXDeliveryCount** プロパティに設定されていた配信試行回数は保持されません。



クライアントによって突然終了された場合、`maxRedelivery` のカウントが間違っ  
て増加する可能性があります。突然終了すると、クライアントはサーバーと通信  
できなくなります。たとえば、接続を閉じずにクライアントが終了した場合や、  
クライアントアプリケーションがクラッシュした場合などです。クライアントア  
プリケーションが突然終了すると、EMS サーバーはクライアントに送信された  
メッセージがアプリケーションに伝達されなかったとしても、それらのメッセ  
ジをすべて配信済みとしてカウントします。

詳細は、20 ページの「[未配信メッセージキュー](#)」を参照してください。

## overflowPolicy

トピックとキューの `overflowPolicy` プロパティを指定し、`maxbytes` または  
`maxmsgs` のいずれかに設定されたメッセージの許容量を超過した場合の影響を変  
更できます。

`overflowPolicy` プロパティは、次の形式で設定します。

```
overflowPolicy=default|discardOld|rejectIncoming
```

`overflowPolicy` を設定しない場合、ポリシーは `default` になります。

`overflowPolicy` プロパティの影響は、トピックまたはキューのどちらに設定し  
たかによって異なるため、`overflowPolicy` の各プロパティ値の影響はトピック  
とキューに分けて説明します。

マルチキャストが有効なトピックに `overflowPolicy` が設定されている場合、マ  
ルチキャスト デーモンでそれが考慮されます。つまり、マルチキャストデー  
モンは、サーバー内のバックログではなく、デーモン内のバックログに基づいて  
メッセージを破棄します。

マルチキャストが有効になっていないトピックやコンシューマの場合、  
`overflowPolicy` への応答は EMS サーバーで行われます。

`.conf` ファイルでワイルドカードを使用すると、複数の親から `overflowPolicy`  
ポリシーを継承した場合の影響は次のようになります。

- 子のデスティネーションにデフォルト以外の `overflowPolicy` ポリシーが設  
定されている場合、そのポリシーが使用され、競合するポリシーは親から継  
承しません。
- `OVERFLOW_REJECT_INCOMING` ポリシーが設定されている親デスティネーション  
がある場合、そのポリシーが子のデスティネーションに継承され、その他の  
ポリシーは無効にされます。
- `OVERFLOW_REJECT_INCOMING` ポリシーが設定されている親デスティネーション  
がなく、`OVERFLOW_DISCARD_OLD` のポリシーが設定されている親デスティネ  
ーションがある場合、そのポリシーが子のデスティネーションに継承されます。
- `OVERFLOW_REJECT_INCOMING` ポリシーが設定されている親デスティネーション  
も、`OVERFLOW_DISCARD_OLD` ポリシーが設定されている親デスティネーション  
もない場合、`default` のポリシーが子のデスティネーションに継承されます。

## default

トピックの場合、**default** を指定すると、メッセージは各サブスクライバに順番に送信されます。サブスクライバで **maxbytes** または **maxmsgs** 設定に到達すると、そのサブスクライバはメッセージを受け取らなくなります。メッセージプロデューサにエラーは返されません。

キューの **default** プロパティを指定すると、**maxbytes** または **maxmsgs** の設定値を超えた場合、新しいメッセージがサーバーに拒否され、メッセージプロデューサにエラーを返します。

トピックとキューのこのデフォルトの動作は、EMS 4.3 の場合と同じです。

## discardOld

トピックの **discardOld** プロパティを指定すると、メッセージの最大保存量を超えて、サーバーに未配信メッセージが著しく溜まっているサブスクライバがいる場合、古いメッセージが破棄されてそのサブスクライバには配信されません。

**discardOld** プロパティの設定は個々のサブスクライバに影響します。たとえば、あるトピックに 3 つのサブスクライバがあり、そのうちひとつのサブスクライバのメッセージがサーバーの最大保存量を超えているとします。この場合、このサブスクライバの一番古いメッセージが破棄され、残りのふたつのサブスクライバはすべてのメッセージを受信し続けます。

トピックまたはキューのメッセージが **maxbytes** または **maxmsgs** の値を超えると、最も古いメッセージがサイレントに破棄されます。プロデューサにエラーは返されません。

## rejectIncoming

トピックの **rejectIncoming** プロパティを指定すると、メッセージの最大保存量を超えて、サーバーに未配信メッセージが著しく溜まっているサブスクライバがいる場合、新しいメッセージはすべて拒否されて、そのメッセージのプロデューサにエラーが返されます。

キューの **rejectIncoming** プロパティを指定すると、キュー内のメッセージ数が **maxbytes** または **maxmsgs** の設定値を超えた場合、新しいメッセージはすべて拒否されて、そのメッセージのプロデューサにエラーが返されます（これは、**default** の動作と同じです）。

## 例：

キューに入れられたメッセージ数が 1000 を超えたときに **myQueue** キューからメッセージが破棄されるようにするには、次のコマンドを入力します。

```
setprop queue myQueue maxmsgs=1000,overflowPolicy=discardOld
```

いずれかのトピックサブスクライバ宛ての未配信メッセージによって使用されるメモリ量が 100KB を超えたときに、**myTopic** にパブリッシュされる後続のメッセージがすべて拒否されるようにするには、次のコマンドを入力します。

```
setprop topic myTopic maxbytes=100KB,overflowPolicy=rejectIncoming
```



prefetch

クライアントのメッセージコンシューマ部分とサーバーは、`prefetch` プロパティの設定に従い、協調してメッセージの取得を調整します。`prefetch` プロパティはトピックとキューの両方について設定できます。

`prefetch` プロパティは、次の形式で設定します。

```
prefetch=value
```

`value` には、表 13 にある値のいずれかを指定します。

表 13 Prefetch

値	説明
2 以上	メッセージコンシューマは自動的にサーバーからメッセージを取得します。メッセージコンシューマは、 <code>value</code> に指定された数より多くのメッセージを取得しません。  詳細は、65 ページの「 <a href="#">自動取得の有効化</a> 」を参照してください。
1	メッセージコンシューマは自動的にサーバーからメッセージを取得します。現在のメッセージを保持しない場合にのみ、取得を開始します。
none	自動取得を行いません。つまり、メッセージコンシューマは、クライアントが <code>receive</code> を呼び出した場合にのみメッセージの取得を開始します。呼び出しは、明示的な同期呼び出しか、暗黙的な呼び出し（非同期コンシューマの場合）のいずれかです。  この値は、トピックまたはグローバルキューとともに使用することはできません。  詳細は、66 ページの「 <a href="#">自動取得の無効化</a> 」を参照してください。
0	デスティネーションは、名前的一致した親デスティネーションの <code>prefetch</code> プロパティの値を継承します。親デスティネーションがないか、 <code>prefetch</code> の値が設定されている親デスティネーションがない場合、デフォルトでキューの値は 5 で、トピックは 64 です。  デスティネーションに <code>prefetch</code> プロパティの値が設定されていない場合は、デフォルトで 0 の値が設定されます（これにより、親デスティネーションの <code>prefetch</code> プロパティの値が継承されます）。  詳細は、66 ページの「 <a href="#">継承</a> 」を参照してください。





**prefetch** および **maxRedelivery** の両方のプロパティにゼロ以外の値が設定されている場合、いずれかのメッセージが **maxRedelivery** プロパティの設定値を超えると、プリフェッチしたメッセージが失われることがあります。たとえば、**prefetch=5** および **maxRedelivery=4** が設定されているとします。最初のメッセージの再配信が 4 回試行されると、**maxRedelivery** の設定値に達し、メッセージは（予定どおり）未配信メッセージキューに送信されます。ところが、プリフェッチが 4 回試行された他のメッセージも未配信メッセージキューに送信されますが、受信アプリケーションはこれら进行处理しません。この対処方法として **prefetch=none** と設定する方法がありますが、これによって大量のインタフェースのパフォーマンスに影響を与えます。

## 背景

サーバーのデスティネーションからメッセージコンシューマにメッセージを配信する場合、「fetch」と「accept」のふたつの段階があります。

- **fetch** 段階では、さらに 2 段階に分かれてメッセージコンシューマとサーバー間でやり取りが行われます。
  - メッセージコンシューマが取得段階を開始する際、新しいメッセージを受信可能であることをサーバーに通知します。
  - サーバーはこれに応答して、ひとつ以上のメッセージをクライアントに送信し、メッセージはメッセージコンシューマに保存されます。
- **accept** 段階では、クライアントのコードを使ってメッセージコンシューマからメッセージを取得します。

**receive** 呼び出しは、このふたつの段階にどちらにも使用されます。この呼び出しは、必要に応じて取得を開始し、メッセージコンシューマからのメッセージに対して確認応答を行います。

クライアントプログラムの待機時間を少なくするために、メッセージコンシューマは、メッセージをプリフェッチすることができます。これより、サーバーからバッチでメッセージを取得し、クライアントコードでひとつずつ確認応答できるようにそれらのメッセージを保持します。

## 自動取得の有効化

メッセージの取得を自動化するには、**prefetch** プロパティに正の整数を設定します。自動取得では、メッセージを受信できるようになると、クライアントコードがメッセージを受信する準備ができるようになるまで待機するように設定されます。サーバーがメッセージを転送する間のクライアントのアイドルタイムを削減または除去することで、システムパフォーマンスは向上します。

ただし、キューコンシューマがメッセージグループをプリフェッチする場合、サーバーは他のキューコンシューマにそれらのメッセージを配信しません（最初のキューコンシューマのサーバー接続が切断されない限り）。

## 自動取得の無効化

自動取得を無効にするには、`prefetch=none` と設定します。

`prefetch=none` と設定しても、キューコンシューマはメッセージを保持することがあります。たとえば、`receive` 呼び出しが取得を開始し、サーバーがメッセージの転送を完了する前にタイムアウトになったとします。この場合、取得されたメッセージはメッセージコンシューマ内で待機状態になります。次の `receive` 呼び出しでは、新しいメッセージを取得せず、現在待機中のメッセージに対して確認応答します。さらに次の `receive` 呼び出しで、新しいメッセージの取得が開始されます。

待機中のメッセージは依然としてキューコンシューマに保持され、サーバーはそのメッセージを別のキューコンシューマに配信しません（最初のキューコンシューマのサーバー接続が切断されない限り）。メッセージの待機状態が長くないようにするには、タイムアウトを設定せずに `receive` を呼び出すか、タイムアウトを設定した場合は、繰り返し `receive` 呼び出しを行い、呼び出し間隔が短くなるようにコードをプログラムします。



グローバルキューまたはトピックに対して自動取得を無効にすることはできません。

## 継承

デスティネーションが、名前的一致した親デスティネーションの `prefetch` プロパティの値を継承する場合、条件によって次のいずれかの動作になります。

- すべての親デスティネーションの値が `none` に設定されている場合、子のデスティネーションは `none` の値を継承します。
- ゼロ以外の数値が設定されている親デスティネーションがある場合、子のデスティネーションは親デスティネーションの中で「最も高い」値を継承します。
- ゼロ以外の数値が設定されている親デスティネーションがない場合、子のデスティネーションはデフォルトの値（5）を使用します。

## redeliveryDelay

`redeliveryDelay` が設定されている場合、EMS サーバーは確認応答されないメッセージをキューに返すまで、指定された期間待機します。以前に配信されたメッセージが正常な確認応答を受け取らなかった場合、EMS サーバーは再配信用に設定された遅延期間を待機した後、キューでメッセージをもう一度取得可能にします。これは、トランザクションのロールバック、セッションまたはメッセージの回復、セッションまたは接続の切断、あるいはクライアントアプリケーションの突然の終了などが生じた場合によく起こります。ただし、遅延時間は正確ではなく、多くの場合、指定された `redeliveryDelay` を超過します。



再配信の遅延は、ルーティングされたキューには使用できません。

値は、秒、分、または時間で指定できます。値は、15 秒から 8 時間までの範囲で指定できます。

`redeliveryDelay` プロパティは、次の形式で設定します。

```
redeliveryDelay=time[sec|min|hour]
```

`time` には、秒数を指定します。ゼロは特別な値であり、再配信の遅延がないことを示します。

`time` 値の単位として、`sec` (秒)、`min` (分)、または `hour` (時間) を指定できます。例:

```
redeliveryDelay=30min
```

30 分の再配信遅延を指定します。

遅延間隔の間、メッセージは `$sys.redelivery.delay` キューに置かれます。このキューを参照することはできますが、コンシュームしたり、ページしたりすることはできません。ただし、遅延メッセージの送信元であるキューをページしたり、メッセージ ID を使用してメッセージを削除したりすると、`$sys.redelivery.delay` からメッセージが即時に削除されます。



メッセージが `$sys.redelivery.delay` キューにある間、メッセージの送信元であるキューには入れられないため、統計上のメッセージ数に含まれなくなります。これには、`maxmsgs`、`maxbytes`、`flowControl` などが含まれます。

## secure

デスティネーションの `secure` プロパティが有効になっている場合、ユーザーがそのデスティネーション上で操作の実行を試みるたびに、サーバーはユーザーのアクセス権限を確認します。

`secure` プロパティは、次の形式で設定します。

```
secure
```

デスティネーションに `secure` プロパティが設定されていない場合、そのデスティネーションに対するアクセス権限の確認は行われなため、適切な権限のある任意のユーザーはそのトピックまたはキューに対して自由に操作できます。



`secure` プロパティは SSL とは別の処理で、サーバー内での基本認証とアクセス権限の検証を制御します。クライアントとサーバー間のセキュアな接続の設定方法については、453 ページの「[SSL プロトコルの使用](#)」を参照してください。

サーバーの `authorization` プロパティは、アクセス権限を確認するマスタースイッチの役割を果たします。つまり、`authorization` プロパティが有効にされている場合のみ、サーバーは、セキュアなデスティネーションに対するユーザーのアクセス権限を確認します。アクセス制限を指定するには、`authorization` 設定パラメータを有効にし、影響を受ける各デスティネーションの `secure` プロパティを設定するという作業を両方とも行う必要があります。

アクセス権限と `secure` プロパティの設定方法についての詳細は、255 ページの [第8章、「認証とアクセス権限」](#) を参照してください。

## sender\_name

`sender_name` プロパティを指定して、このデスティネーションに送信されるメッセージに、サーバーが送信元のユーザー名を挿入することができます。

`sender_name` プロパティは、次の形式で設定します。

```
sender_name
```

`sender_name` プロパティを有効にすると、接続が確立したときにサーバーはメッセージプロデューサからユーザー名を受け取り、そのユーザー名をメッセージの `JMS_TIBCO_SENDER` プロパティに書き込みます。

メッセージプロデューサは、メッセージ内のプロパティを指定することで、この動作を無効にできます。メッセージプロデューサが `JMS_TIBCO_DISABLE_SENDER` プロパティに `true` を設定してメッセージを送信すると、サーバーは `sender_name` プロパティを無効にし、送信元のユーザー名をメッセージに追加しません。

サーバーの認証が無効にされている場合、サーバーは、サーバーへの接続時にメッセージプロデューサが作成したユーザー名を挿入します。サーバーの認証が有効にされている場合、サーバーは、サーバーへの接続時にメッセージプロデューサが作成したユーザー名を認証し、メッセージ内のプロパティに記述されたユーザー名が認証されたユーザー名となります。SSL が使用されている場合、SSL 接続プロトコルにより、クライアントのデジタル証明書を使用したクライアント認証が保証されます。

## sender\_name\_enforced

`sender_name_enforced` プロパティを指定すると、このデスティネーションに送信されるメッセージに、送信元のユーザー名を強制的に挿入することができます。前述の `sender_name` プロパティで説明した同じ手順で、サーバーはメッセージプロデューサのユーザー名を取得します。ただし、`sender_name` プロパティと異なり、メッセージプロデューサはこのプロパティを無効にできません。

`sender_name_enforced` プロパティは、次の形式で設定します。

```
sender_name_enforced
```

`sender_name` プロパティもデスティネーションに設定されている場合、このプロパティによって `sender_name` プロパティは無効にされます。



ビジネス状況によっては、クライアントはメッセージプロデューサのユーザー名を開示したくない場合があります。この場合、こうしたクライアントは、`sender_name` または `sender_name_enforced` のプロパティが有効にされているデスティネーションにメッセージが送信されないようにすることができます。

このような状況では、EMS サーバーのオペレータは、これらのプロパティが有効にされているデスティネーションの一覧を開示するポリシーを開発します。これにより、クライアントは、メッセージプロデューサのユーザー名を開示してしまうデスティネーションにメッセージを送信することを避けることができます。

## store

`store` プロパティは、このデスティネーションに送信されたメッセージが保存される場所を決定します。メッセージは、ファイルまたはデータベースに保存できます。複数ストアを使用および設定する方法については、29 ページの「[複数ストアへのメッセージの保存](#)」を参照してください。



`setprop` または `addprop` コマンドを使用してトピックまたはキューの `store` 設定を変更する前に、デスティネーションで着信メッセージのフローを止める必要があります。

`store` プロパティは、次の形式で設定します。

```
store=name
```

`name` には、`stores.conf` ファイルで定義されているストアの名前を指定します。

たとえば、次のように指定すると、デスティネーション `giants.games` に送信されたすべてのメッセージが、`baseball` という名前のストアに送信されます。また、他のすべてのデスティネーションに送信されたメッセージは、`everythingelse` に保存されます。

```
> store=everythingelse
giants.games store=baseball
```

デスティネーションごとにひとつのストアのみを使用できます。たとえば、重複するワイルドカードによってトピックが複数ストアのプロパティを継承するために競合が生じる場合、そのトピックの作成は失敗します。



このパラメータを使用するには、最初に `tibemsd.conf` ファイルでこの機能を有効にする必要があります。`stores.conf` ファイルも存在する必要がありますが、デスティネーションに関連付けられた唯一のストア名がデフォルトのストアファイルである場合は、空のままで構いません。

詳細は、29 ページの「[複数ストアへのメッセージの保存](#)」を参照してください。

## trace

`trace` プロパティは、このデスティネーションに対してトレースを有効にするかどうかを指定します。

`trace` プロパティは、次の形式で設定します。

```
trace = [body]
```

`=body` を指定せずに `trace` を指定すると、メッセージシーケンス、メッセージ ID、およびメッセージサイズが含まれたメッセージが生成されます。`trace=body` と指定すると、メッセージ本文を含むトレースメッセージを生成します。メッセージトレースについての詳細は、440 ページの「[メッセージのトレース](#)」を参照してください。

## デスティネーションの作成と変更

デスティネーションは JNDI サーバーまたは LDAP サーバーに保存することができる、「静的」な管理対象オブジェクトの典型です。管理対象オブジェクトは、EMS サーバーにも保存でき、EMS に実装された JNDI を使用して参照することができます。このセクションでは、第 6 章で説明されている EMS 管理ツールを使用して、EMS でデスティネーションオブジェクトを作成および変更する方法を説明します。

キューの作成には `create queue` コマンド、トピックの作成には `create topic` コマンドを使用します。たとえば、`myQueue` という名前のキューを新規作成する場合は、次のコマンドを入力します。

```
create queue myQueue
```

`myTopic` という名前のトピックを作成するには、次のコマンドを入力します。

```
create topic myTopic
```

キューとトピックのデータは、EMS サーバーの `queues.conf` および `topics.conf` の各ファイルに保存されます。`show queues` および `show topics` コマンドを使用して、EMS サーバー上のすべてのキューとトピックをリスト表示できます。また、`show queue` と `show topic` コマンドを使用して、特定のキューとトピックの設定の詳細を表示できます。

キューまたはトピックには、デスティネーションの固有の特性を定義するプロパティを含めることができます。これらのプロパティの詳細は、55 ページの「[デスティネーションのプロパティ](#)」で説明されています。`addprop queue`、`addprop topic`、`setprop queue`、`setprop topic`、`removeprop queue`、および `removeprop topic` のコマンドを使用して、キューまたはトピックを新規作成するときにこれらのプロパティを指定したり、既存のキューまたはトピックのプロパティを変更したりすることができます。

たとえば、メッセージ数が 1000 を超えたときに `myQueue` キューからメッセージが破棄されるようにするには、次のコマンドを入力して、`overflowPolicy` を設定します。

```
addprop queue myQueue maxmsgs=1000,overflowPolicy=discardOld
```

`overflowPolicy` を `discardOld` から `rejectIncoming` に変更するには、次のコマンドを入力します。

```
addprop queue myQueue overflowPolicy=rejectIncoming
```

`setprop queue` および `setprop topic` コマンドは、コマンドによって明示的に設定されていないプロパティを削除するときに使用します。たとえば、`maxmsgs` のプロパティ値を 100 に変更し、`overflowPolicy` パラメータを削除するには、次のコマンドを入力します。

```
setprop queue myQueue maxmsgs=100
```

## セキュアなデスティネーションの作成

デフォルトで、認証されたすべての EMS ユーザーに、トピックまたはキューに対して操作を実行できるアクセス権限が付与されます。トピックまたはキューの `secure` プロパティを設定し、`grant topic` または `grant queue` コマンドを使用して、ユーザーまたはグループごとにデスティネーションに対して実行可能な操作を指定できます。

`secure` プロパティを設定するには、EMS サーバーの `authorization` プロパティを有効にする必要があります。

たとえば、`myQueue` という名前のセキュアキューを作成し、「joe」と「eric」というユーザーだけがこのキューにメッセージを送信でき、「sally」というユーザーだけがそのメッセージを受信できるようにするには、EMS 管理ツールで以下のコマンドを入力します。

```
set server authorization=enabled
create queue myQueue secure
grant queue myQueue joe send
grant queue myQueue eric send
grant queue myQueue sally receive
```

詳細は、255 ページの第8章、「[認証とアクセス権限](#)」を参照してください。



## ワイルドカード

静的に作成したデスティネーションを `queues.conf` および `topics.conf` ファイルに指定するときにワイルドカードを使用できます。デスティネーション名にワイルドカードを使用して、「親」デスティネーションと「子」デスティネーションとの関係を定義できます。この場合、子デスティネーションは親のプロパティとアクセス権限を継承します。76 ページの「[継承](#)」で説明されている継承ルールを理解するには、まずワイルドカードを理解する必要があります。

### ワイルドカード \* と >

プロパティの継承ルールを理解するには、\* と > のふたつのワイルドカードを理解する必要があります。

- ワイルドカード「>」自体は、どのデスティネーション名とも一致します。
- テキストの末尾に置いた場合、「>」の部分は、そのテキストの後に続くひとつ以上の要素に一致します。例：

```
foo.>
```

`foo.bar`、`foo.boo`、`foo.boo.bar`、および `foo.bar.boo` のいずれとも一致します。

- ワイルドカード「\*」は、「\*」の箇所に任意のトークンが入ることを示します。例：

```
foo.*
```

`foo.bar` と `foo.boo` は一致しますが、`foo.bar.boo` は一致しません。

```
foo.*.bar
```

`foo.boo.bar` は一致しますが、`foo.bar` は一致しません。

### 重複するワイルドカードと独立プロパティ

デスティネーションプロパティには、独立プロパティがいくつかあります。独立プロパティは、サーバーから各デスティネーションに 1 回だけ設定できます。既存のデスティネーションで独立プロパティに値が設定されている場合、他の値を割り当てようとするとう失敗します。

ワイルドカードを使用するデスティネーションの指定で重複があると、独立プロパティで競合が発生する可能性があります。たとえば、`channel` プロパティで次のように設定している場合を考えてみます。

```
topic.multicast.> channel=multicast_1
topic.multicast.quotes.* channel=multicast_2
```

`topic.multicast.quotes.tibx` トピックは `multicast_1` と `multicast_2` の両方のチャンネルに割り当てられます。したがって、`topic.multicast.>` と `topic.multicast.quotes.*` は共存できません。これらを作成しようとしても、操作は失敗します。

独立プロパティには以下があります。

- `channel`
- `store`

## トピックでのワイルドカードの使用

TIBCO Enterprise Message Service では、状況に応じてトピック名にワイルドカードを使用できます。

- ワイルドカードを使用したトピックをサブスクライブすることができます。

ワイルドカードを含むトピックをサブスクライブすると、ワイルドカードに一致するトピックにパブリッシュされたメッセージをすべて受信することができます。たとえば、`foo.*` をサブスクライブすると、`foo.bar` という名前のトピックにパブリッシュされたメッセージを受け取ります。

設定ファイルにワイルドカードに一致するトピック名（例：`foo.*`、`foo.>`、`foo.bar`）があるかどうかにかかわらず、ワイルドカードを使用したトピック名（例：`foo.*`）をサブスクライブすることができます。ただし、設定ファイルに一致するトピック名がない場合、そのトピックにパブリッシュされるメッセージはありません。

- ワイルドカードを使用したトピックにパブリッシュすることはできません。
- 設定ファイルに `foo.bar` がない場合、設定ファイルに `foo.*` または `foo.>` のいずれかがあれば、`foo.bar` にパブリッシュすることができます。
- ルーティング型のトピックメッセージでは、設定されているグローバルトピックの直接のサブセットにあたるトピック（またはグローバルトピックそのもの）をサブスクライブが指定する必要があります。詳細は、515 ページの「ワイルドカード」を参照してください。

## キューでのワイルドカードの使用

TIBCO Enterprise Message Service では、状況に応じてキュー名にワイルドカードを使用できます。ワイルドカードを使用したキュー名でメッセージの送受信はできません。ただし、設定ファイルではキュー名にワイルドカードを使用できます。

たとえば、キューの設定ファイルに次の行が含まれている場合、

```
foo.*
```

ユーザーは、動的に `foo.bar` や `foo.bob` などを作成できますが、`foo.bar.bob` は作成できません。

## ワイルドカードとマルチキャスト

マルチキャスト対応のトピックにメッセージをパブリッシュすると、該当するトピックに定義されているマルチキャストチャネルにメッセージが送信されます。ひとつのワイルドカードで、それぞれ異なるマルチキャストチャネルに属する複数のマルチキャスト対応トピックを指定できます。

たとえば、次のような設定を考えてみます。

```
>
topic.info
topic.quotes
topic.multicast.info channel=channel-1
topic.multicast.quotes channel=channel-2
```

トピック > にサブスクライブしているメッセージコンシューマは、`topic.info` と `topic.quotes` にパブリッシュされているメッセージを TCP 経由で受信します。また、`topic.multicast.info` にパブリッシュされているメッセージはマルチキャストチャネル `channel-1` 経由で受信する一方、`topic.multicast.quotes` にパブリッシュされているメッセージはマルチキャストチャネル `channel-2` 経由で受信します。言い換えれば、ワイルドカードコンシューマはメッセージをマルチキャストと TCP の両方を介して受信する場合があります。

## ワイルドカードと動的に作成されたデスティネーション

クライアントの代わりに EMS サーバーが動的にデスティネーションを作成することができます (331 ページの「[トピックとキューの動的作成](#)」を参照)。`.conf` ファイルでワイルドカードを使用して、動的に作成されたデスティネーションに許可できる名前を制御できます。

動的に作成されたデスティネーションには、前述の静的デスティネーションで説明したのと同じワイルドカードの基本ルールが適用されます。

例 :

- `queues.conf` ファイルに、次のワイルドカードが含まれている場合、  

```
>
```

 EMS サーバーは任意の名前のキューを動的に作成できます。
- `topics.conf` ファイルに、ワイルドカードを使用した次のトピック名しか含まれていない場合、  

```
foo.>
```

 EMS サーバーは、動的に `foo.bar`、`foo.boo`、`foo.boo.bar`、`foo.bar.boo` などの名前の付いたトピックを作成できます。
- `queues.conf` ファイルに、ワイルドカードを使用した次のトピック名しか含まれていない場合、  

```
foo.*
```

 EMS サーバーは、`foo.bar`、`foo.boo` などの名前のキューを動的に作成できますが、`foo.bar.boo` などの名前のキューは作成できません。
- `topics.conf` ファイルに、ワイルドカードを使用した次のトピック名しか含まれていない場合、  

```
foo.*.bar
```

 EMS サーバーは、`foo.boo.bar` などの名前のトピックを動的に作成できますが、`foo.bar` などの名前のトピックは作成できません。

## 継承

このセクションでは、プロパティとアクセス権の継承について説明します。ワイルドカードについての詳細は、73 ページの「ワイルドカード」を参照してください。デスティネーションのプロパティについての詳細は、55 ページの「デスティネーションのプロパティ」を参照してください。アクセス権についての詳細は、255 ページの第8章、「認証とアクセス権限」を参照してください。

### プロパティの継承

トピックとキューのどちらも、すべてのデスティネーションプロパティが継承可能です。つまり、「ワイルドカード」を使用したデスティネーションのプロパティは、そのワイルドカードと一致する名前のすべてのデスティネーションに継承されます。

たとえば、`topics.conf` ファイル内に以下のトピック名がある場合：

```
foo.* secure
foo.bar
foo.bob
```

`foo.bar` と `foo.bob` のトピックは、それらの親である `foo.*` から `secure` を継承するため、`secure` のトピックになります。EMS サーバーが `foo.new` のトピックを動的に作成するように設定されている場合、そのトピックにも `secure` プロパティが継承されます。

親デスティネーションから継承されたプロパティは、子のデスティネーションで定義されたプロパティに追加されます。

たとえば、`topics.conf` ファイル内に以下のトピック名がある場合：

```
foo.* secure
foo.bar sender_name
```

`foo.bar` は、`secure` と `sender_name` の両方のプロパティを持つことになります。

前述の例では、`foo.*` のトピックをセキュアにするには、`foo.bar` をセキュアにする必要があります。これは、EMS では継承されたプロパティを削除することができないからです。ただし、値が設定されているプロパティについては、親デスティネーションで設定された値を無効にすることができます。

たとえば、`queues.conf` ファイル内で、キューが以下のように設定されている場合：

```
foo.* maxbytes=200
foo.bar maxbytes=2000
```

`foo.bar` キューの `maxbytes` の値は 2000 になります。

デスティネーションに複数の親デスティネーションがある場合、デスティネーションはすべての親からプロパティを継承します。例：

```
> sender_name
foo.* secure
foo.bar trace
```

foo.bar トピックは、[sender\\_name](#)、[secure](#)、および [trace](#) の三つのプロパティを継承します。

デスティネーションの複数ある親デスティネーションのプロパティ値が競合している場合、デスティネーションは最小の値を継承します。例：

```
> maxbytes=2000
foo.* maxbytes=200
foo.bar
```

foo.bar トピックの [maxbytes](#) の値は、200 になります。

プロパティの継承は強力な機能ですが、理解や管理するには複雑かもしれません。プロパティをトピックとキューに割り当てる前に、入念に計画する必要があります。プロパティの割り当てにワイルドカードを使用する場合は、細心の注意が必要です。たとえば、[topics.conf](#) ファイル内に次の行を入力した場合：

```
> store=mystore
```

後からどのようなエントリを追加しても、各トピックはストアメッセージになります。これにより、保存のためにメモリを大量に消費することになり、システムパフォーマンスが著しく低下することがあります。

## アクセス権限の継承

アクセス権限の継承はプロパティの継承と同様に行われます。親トピックにアクセス権限が付与されていれば、子にもその権限が継承されます。たとえば、Bob がグループ A に属していて、グループ A に、トピックへの [publish](#) 権限がある場合、Bob にもそのトピックへの [publish](#) 権限が与えられます。

1 人のユーザーに付与される権限は、そのユーザーの権限セットと、そのユーザーが属する各グループのすべての権限セットを合わせたものです。これらの権限のセットは、足し合わせることができます。アクセス権限は正のブール値を継承します。継承によって付与された権限は、削除できません。

ワイルドカードのすべてのルールが、アクセス権の継承に適用されます。たとえば、foo.\* のトピックにパブリッシュするアクセス権限が与えられているユーザーには、foo.bar および foo.new にパブリッシュするアクセス権限も与えられます。

ワイルドカードについての詳細は、73 ページの「[ワイルドカード](#)」を参照してください。アクセス権についての詳細は、273 ページの「[ユーザーのアクセス権限](#)」を参照してください。

## デスティネーションブリッジ

アプリケーションによっては、同じメッセージを複数の、タイプが異なるデスティネーションに送る必要があります。たとえば、あるアプリケーションは、メッセージをキューに送信して負荷を分散する必要がありますが、このアプリケーションは、複数の監視アプリケーションにメッセージをパブリッシュする必要もあります。別の例として、メッセージを複数のトピックにパブリッシュするアプリケーションが挙げられます。すべてのメッセージは、バックアップおよびデータマイニング目的でデータベースにも送信する必要があります。すべてのメッセージを収集し、データベースに送信するために、キューが使用されます。

アプリケーションは、メッセージを処理して、必要なデスティネーションに複数回送ることができます。ただし、そうした処理を行うには、アプリケーションの膨大なコーディングが必要になります。EMS では、サーバーベースのソリューションで、この問題を解決しています。ひとつのデスティネーションに送信されたメッセージが、ブリッジされた他のすべてのデスティネーションにも配信されるように、デスティネーション間にブリッジを作成できます。

ひとつのデスティネーションと他の同じタイプまたは異なるタイプのひとつ以上のデスティネーションとの間にブリッジを作成します。たとえば、トピックからキューへ、またはキューからトピックへブリッジを作成することができます。また、ひとつのデスティネーションと複数のデスティネーションの間にブリッジを作成することもできます。たとえば、トピック `a.b` からキュー `q.b` とトピック `a.c` へブリッジを作成することができます。

ブリッジを指定する場合は、個々のデスティネーション名を指定するか、ワイルドカードを使用できます。たとえば、トピック `foo.*` からキュー `foo.queue` へのブリッジを指定すると、`foo.*` のパターンと一致する名前のトピックに配信されたメッセージが、`foo.queue` へ送られます。



グローバルトピックはサーバー間でルーティングされますが、グローバルキューは近隣に限定されるため、ほとんどの場合、メッセージをトピックに送信してからキューにブリッジするのがベストプラクティスです。

複数のブリッジが存在するときは、ワイルドカードを使ってデスティネーション名を指定すると、メッセージが2回配信される場合があります。たとえば、キュー `Q.1` と `Q.>` の両方が `QX.1` にブリッジされている場合、キューに送信したメッセージがサーバーから `QX.1` に2回配信されます。

以下の三つの図に、ブリッジのシナリオ例を示します。

図 11 ひとつのトピックからひとつのキューへのブリッジ

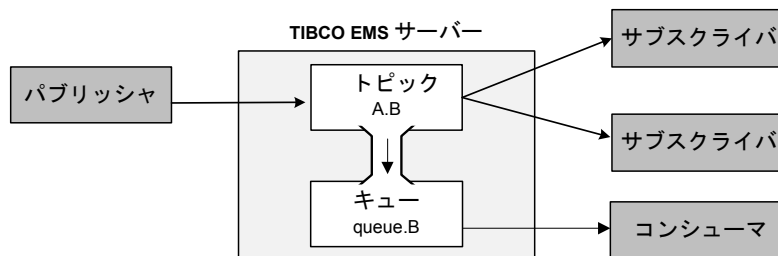


図 12 ひとつのトピックから複数のデスティネーションへのブリッジ

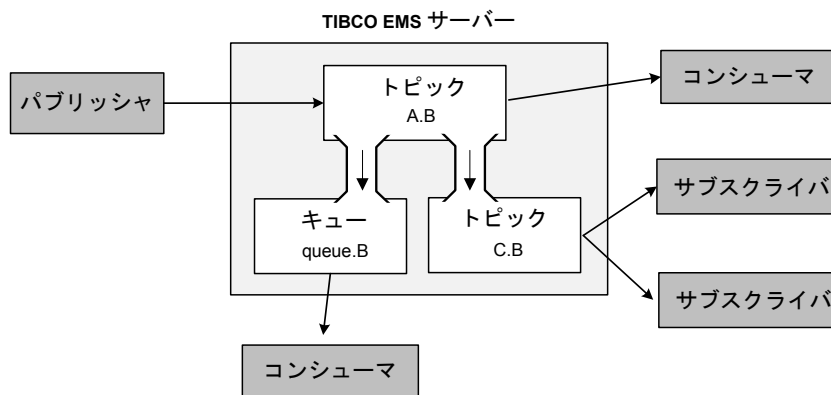
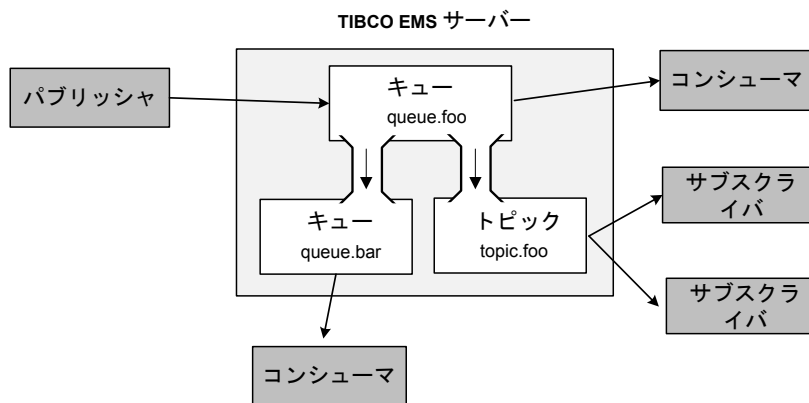


図 13 ひとつのキューから複数のデスティネーションへのブリッジ



ふたつのキュー間にブリッジが存在すると、メッセージは両方のキューに配信されます。各キューは独立して動作します。メッセージが一方のキューから取得されても、もう一方のキューのメッセージステータスには影響しません。

ブリッジは遷移しません。つまり、ブリッジが設定されたデスティネーションに送信されたメッセージは、ブリッジが指定されているデスティネーションのみに送信され、複数のブリッジ全体には配信されません。たとえば、トピック A.B からキュー Q.B へブリッジが作成されているとします。さらに、キュー Q.B からトピック B.C へブリッジが作成されているとします。トピック A.B に配信されたメッセージは、キュー Q.B にも配信されますが、トピック B.C には配信されません。

ブリッジによってターゲットデスティネーションにコピーされたソースメッセージには、ターゲットデスティネーションで新しいメッセージ識別子を与えられます。ターゲットデスティネーションのストアパラメータによっては、追加のストレージが必要になる場合があります。

## ブリッジの作成

ブリッジは、`bridges.conf` 設定ファイルで定義します。以下の構文を使ってブリッジを指定します。

```
[destinationType:destinationName]
    destinationType=destinationToBridgeTo selector="messageSelector"
```

ここで、`destinationType` はデスティネーションのタイプ（`topic` または `queue`）、`destinationName` はブリッジ作成元のデスティネーションの名前、`destinationToBridgeTo` はブリッジ作成先のデスティネーションの名前、`selector="messageSelector"` はデスティネーションが受信すべきメッセージのサブセットを指定するためのオプションのメッセージセレクタです。



各 *destinationName* には、ワイルドカードを指定できます。したがって、ワイルドカードのパターンに一致するデスティネーションへは、指定したブリッジが作成されます。各 *destinationName* には、複数の *destinationToBridgeTo* を指定できます。

たとえば、[図 11](#) と [図 12](#) に示したブリッジは、`bridges.conf` ファイルで次のように指定されます。

```
[topic:A.B]
  queue=queue.B
  topic=C.B
```

ブリッジが作成されているデスティネーションのメッセージセレクトアを指定する方法については、以下のセクションで説明します。

## ブリッジするメッセージの選択

デフォルトでは、ブリッジが設定されたデスティネーションに送信されたすべてのメッセージは、ブリッジされているすべてのデスティネーションに送信されます。これにより、ブリッジが設定された各デスティネーションが、元のデスティネーションに送信されたメッセージのサブセットのみに関わる場合、不要なネットワークトラフィックが発生します。オプションでブリッジごとにメッセージセレクトアを指定し、そのブリッジ経由で送信されるメッセージを指定することもできます。

ブリッジが設定されたデスティネーションのメッセージセレクトアは、ブリッジの `selector` プロパティとして指定します。以下は、前のセクションで定義したブリッジのメッセージセレクトアを指定する例です。

```
[topic:A.B]
  queue=queue.B
  topic=C.B selector="urgency in('medium', 'high')"
```

メッセージセレクトアの構文についての詳細は、関連する EMS API リファレンスマニュアルに記載されている `Message` クラスの説明を参照してください。

## アクセス制御とブリッジ

メッセージプロデューサがデスティネーションにメッセージを送信するには、そのデスティネーションへのアクセス権限を持っている必要があります。しかし、ブリッジには、そのターゲットデスティネーションにメッセージを送信することができるアクセス権限が自動的に与えられます。特別な設定は必要ありません。

## トランザクション

メッセージプロデューサがトランザクション内でメッセージを送信すると、ブリッジに送信されたすべてのメッセージはトランザクションの一部になります。このため、トランザクションが成功すると、ブリッジが設定されているすべてのデスティネーションにすべてのメッセージが配信されます。トランザクションが失敗すると、ブリッジが設定されているデスティネーションのコンシューマはメッセージを受け取りません。

ブリッジが設定されているデスティネーションに対してメッセージプロデューサが適切なアクセス権限を持っていないために、そのデスティネーションにメッセージを配信できない場合、トランザクションは完了できず（失敗）、ロールバックされます。

## フロー制御

状況によっては、メッセージコンシューマがメッセージを受信する速度よりもメッセージプロデューサがメッセージを送信する速度が速い場合があります。デスティネーションに対して保留中のメッセージは、配信できるようになるまで、サーバーに保存されます。したがって、メッセージコンシューマが適度な速さでメッセージを受信しないと、サーバーのストレージがメッセージでいっぱいになってしまうことがあります。これを避けるため、EMS ではデスティネーションへのメッセージのフローを制御することができます。各デスティネーションは、保留メッセージを保存する最大サイズ（目標値）を指定できます。この目標値に達すると、新しいメッセージが送られてきたときに EMS はメッセージプロデューサをブロックします。これにより、メッセージコンシューマが保留中のメッセージを受信できるようになるまで、メッセージプロデューサがメッセージを送信する速度を下げます。

### フロー制御の有効化

`tibemspd.conf` の `flow_control` パラメータを設定して、EMS サーバーのフロー制御をグローバルに有効または無効にできます。`flow_control` パラメータが `disabled` の場合（デフォルト設定）、サーバーはデスティネーションのフロー制御を行いません。`flow_control` パラメータが `enabled` の場合、サーバーは各デスティネーションに指定されたフロー制御の設定を実行します。設定パラメータの操作についての詳細は、175 ページの [第 7 章、「設定ファイルの使用」](#) を参照してください。

デスティネーションへのメッセージのフローを制御するには、そのデスティネーションの `flowControl` プロパティを設定します。`flowControl` プロパティは、デスティネーションの保留中のメッセージを保存する最大サイズ（目標値）を指定します。サイズの単位を指定しない場合、サイズの単位はバイトになります。単位としては、KB、MB、または GB を指定できます。たとえば、`flowControl = 60MB` は、デスティネーションの保留中のメッセージを保存する最大サイズ（目標値）を 60 メガバイトに指定します。

### フロー制御の実行

デスティネーションの `flowControl` プロパティに設定した値は、保留されるメッセージを保存する最大サイズ（目標値）となります。フロー制御を有効にした場合に使用されるストレージ容量は、有効にする前と比べて、メッセージのサイズ、メッセージプロデューサの数、およびその他の要因により、わずかに上下する程度です。デスティネーションに `flowControl` プロパティを設定して、値を指定しない場合、デフォルトでサーバーは 256KB の値を使用します。

保留中のメッセージのストレージ容量が、指定した制限値に近くなると、サーバーはメッセージプロデューサがメッセージを送信するための新しい呼び出しをすべてブロックします。ストレージの使用量が指定した制限値を下回るか、`flowControl` の制限値が緩和されるまで、呼び出しは返されません。メッセージコンシューマがメッセージを受け取り、保留中のメッセージのストレージ使用量が指定した制限値以下に下がると、サーバーは `send` メッセージ呼び出しを呼び出し元に返すことを許可し、メッセージプロデューサは処理を続行することができます。

デスティネーションのメッセージコンシューマが存在しない場合、サーバーはデスティネーションのフロー制御を行いません。すなわち、開始されたレシーバがキューに存在しない場合、サーバーはそのキューのフロー制御を実行することができません。また、トピックに非アクティブな持続サブスクライバがいるか、現在サブスクライバがいない場合、サーバーはそのトピックのフロー制御を実行することができません。トピックの場合、特定のトピック (`foo.bar` など) にフロー制御を設定すると、そのトピックまたは親トピックのサブスクライバがいる限り (たとえば、`foo.*` のサブスクライバがいる場合)、フロー制御が実行されます。

## マルチキャストとフロー制御

マルチキャストチャネルの転送率が `channels.conf` 設定ファイルのチャネル定義で `maxrate` に指定した最大転送率を上回ると、サーバーでメッセージのバックログが作成される場合があります。`tibemsd.conf` ファイルの `flow_control` パラメータを無効にすると、これらのメッセージはマルチキャスト経由の送信が可能となるまでサーバーにバッファされます。チャネルのバックログは、管理ツールから `show channel` コマンドを使用するか、管理 API の `ChannelInfo` オブジェクトを使用して確認できます。

`flow_control` パラメータを有効にすると、EMS サーバーはバックログを確認してから、マルチキャスト対応のトピックにメッセージをパブリッシュするプロデューサに応答を送信します。メッセージのバックログが存在する場合、バックログがクリアされるまで、サーバーはメッセージプロデューサに対する応答の送信を遅延します。これによって、メッセージプロデューサからトピックへのメッセージ送信率が減ります。



マルチキャストチャネルでフロー制御を起動または停止するかどうかの判断には、デスティネーションプロパティの `flowControl` は使用されません。

## ルートとフロー制御

サーバー間でメッセージがルーティングされるグローバルトピックの場合、メッセージを作成するサーバーまたはメッセージを受信するサーバーのいずれかのサーバーで、トピックにフロー制御を指定できます。フロー制御は、別のサーバーにルーティングされているキューのメッセージには行われません。

メッセージを受信するサーバーのトピックの `flowControl` プロパティを設定すると、保留中のメッセージが制限値に達した場合、トピックのサブスクライバがメッセージを受信して、保留中のメッセージのサイズが指定した制限値以下に下がるまで、メッセージは、ルート経由で転送されません。

メッセージを送信するサーバーのトピックの `flowControl` プロパティを設定すると、ルート経由でメッセージを転送する速度が遅い場合、新しいメッセージを送信する際に、サーバーはトピックパブリッシャをブロックすることがあります。このような状態が起きるのは、ルーティング先のサーバー間でネットワークのレイテンシーが発生するためか、他のサーバーへのフロー制御によって、新しいメッセージが送信されないためです。

## デスティネーションのブリッジとフロー制御

フロー制御は、ブリッジが設定されているデスティネーションに指定できます。ブリッジ先デスティネーションのレシーバでメッセージの処理速度が十分でない場合に、ブリッジ経由で送られるメッセージのフロー速度を下げるには、ブリッジの両側のふたつのデスティネーションに `flowControl` プロパティを設定する必要があります。

## フロー制御、スレッド、デッドロック

フロー制御を行う場合、デッドロックが起きないように注意する必要があります。

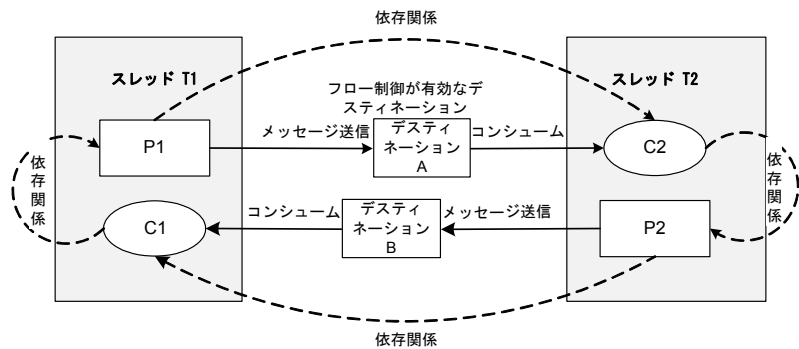
フロー制御がデスティネーションに設定されている場合、そのデスティネーションのメッセージプロデューサは、デスティネーションのコンシューマからのフロー制御信号を待機するスレッドをブロックすることがあります。このようなコンシューマが同じプログラム制御のスレッド内にある場合、デッドロックが発生する可能性があります。つまり、デスティネーションのメッセージが減少するまでプロデューサのブロックは解除されず、ブロックされたスレッドのコンシューマはメッセージの数を減らすことができません。

この状態を簡単に検出できるケースは、プロデューサとコンシューマが同じセッション（セッションはひとつのスレッドに限定される）に参加している場合です。しかし、複雑なケースもあります。デッドロックは、依存関係によって結ばれている場合、複数のスレッドでも（異なるホストのプログラムでも）発生することがあります。たとえば、[図 14](#) のような状況を考えてみます。

- スレッド T1 のプロデューサ P1 にスレッド T2 のコンシューマ C2 がいます。
- スレッド T2 のプロデューサ P2 にスレッド T1 のコンシューマ C1 がいます。
- 循環的な依存関係があるため、プロデューサがフロー制御信号を待機するスレッドをブロックすると、デッドロックが発生することがあります。

依存関係の分析は、`mutex` デッドロックの場合と似ています。デッドロックの発生を避けるには、プログラムと分散システムを同じ方法で分析する必要があります。

図 14 ふたつのスレッドに発生したフロー制御によるデッドロック



## 第 4 章

# 基本操作

この章では、EMS の基本的な設定と、サンプルクライアントアプリケーションを実行して、メッセージをパブリッシュし、ユーザーにトピックをサブスクライブさせる方法の概要を紹介します。

### トピック

---

- [「サンプルクライアントについて」、88 ページ](#)
- [「サンプルクライアントのコンパイル」、89 ページ](#)
- [「EMS 管理ツールを使用したユーザーの作成」、90 ページ](#)
- [「ポイントツーポイントメッセージングの例」、92 ページ](#)
- [「パブリッシュとサブスクライブメッセージングの例」、93 ページ](#)
- [「マルチキャストメッセージングの例」、98 ページ](#)

## サンプルクライアントについて

---

EMS サンプルクライアントは、最短の準備時間と最低限のコーディングで、TIBCO Enterprise Message Service を実行できるように設計されています。

`EMS_HOME/samples` ディレクトリには、`/c`、`/cs`、および `/java` のサブディレクトリがあり、それぞれに C、.NET、および Java のサンプルクライアントが格納されています。この章では、Java サンプルクライアントをコンパイルして実行します。C および .NET のサンプルクライアントを実行する方法についての詳細は、各ディレクトリにある `Readme` ファイルを参照してください。

`EMS_HOME/samples/java` ディレクトリには、次の三つのファイルセットが含まれています。

- TIBCO Enterprise Message Service の実装に使用するサンプルクライアント。
- `jms/samples/java/JNDI` サブディレクトリには、JNDI 照合技術を使用するサンプルクライアントが含まれています。
- `jms/samples/java/tibrv` サブディレクトリには、TIBCO Rendezvous アプリケーションと TIBCO Enterprise Message Service との相互運用を示すサンプルクライアントが含まれています。

この章では、`EMS_HOME/samples/java` ディレクトリ内のサンプルクライアントをいくつか使用します。他のサンプルクライアントのコンパイルと実行についての詳細は、それぞれのディレクトリにある `Readme` ファイルを参照してください。



## サンプルクライアントのコンパイル

---

サンプルクライアントをコンパイルして実行するには、*EMS\_HOME/samples/java* ディレクトリにある "setup" スクリプトを実行する必要があります。Windows システムのセットアップファイルは、*setup.bat* です。Unix システムのセットアップファイルは、*setup.sh* です。

サンプルクライアントのファイルをコンパイルするには、以下の手順に従います。

1. Java JDK 1.6 以降がインストールされていて、PATH 変数に *bin* ディレクトリが追加されていることを確認します。
2. コマンドラインウィンドウまたはコンソールウィンドウを開き、ディレクトリ *EMS\_HOME/samples/java* に移動します。
3. 適切な *setup* スクリプトファイルを開き、*TIBEMS\_ROOT* 環境変数に *EMS\_HOME* ディレクトリへの正しいパス名が指定されていることを確認します。たとえば、Windows システムでは、次のようになります。

```
set TIBEMS_ROOT=C:\¥tibco¥ems¥7.0
```

4. 次のように *setup* と入力して、環境変数とクラスパスを設定します。

```
> setup
```

5. 次のサンプルをコンパイルします。

```
> javac -d . *.java
```

これで、JNDI および *tibrv* のサブディレクトリ内のサンプルを除いて、ディレクトリ内のすべてのサンプルがコンパイルされます。

ファイルのコンパイルが成功すると、クラスファイルが *EMS\_HOME/samples/java* ディレクトリに表示されます。正しくコンパイルできなかった場合は、エラーメッセージが表示されます。

## EMS 管理ツールを使用したユーザーの作成

このセクションでは、EMS サーバーを起動し、管理ツールを使用してふたりの新規ユーザーを作成する方法を説明します。



この章のパラメータ設定では管理ツールを使用しますが、これらのパラメータはすべて設定ファイルを編集して設定することもできます (175 ページの「[設定ファイルの使用](#)」を参照)。さらに、C、.NET、または Java API を使用したプログラムの記述でパラメータを設定することもできます。クライアントによってプログラムの的に設定されるパラメータは、セッション中にのみ設定されます。

### EMS サーバーと EMS 管理ツールの起動

この例では、EMS 管理ツールを使ってトピックとユーザーを作成します。EMS 管理ツールを起動する前に、まず EMS サーバーを起動する必要があります。

#### EMS サーバーの起動

101 ページの「[EMS サーバーの実行](#)」の説明に従って、EMS サーバーを起動します。



Windows が起動しているコンピュータで、スタートメニューから [プログラム] > [TIBCO] > [TIBCO EMS 7.0] > [Start EMS server] を選択して、EMS サーバーを起動することもできます。

#### EMS 管理ツールの起動と EMS サーバーへの接続

118 ページの「[EMS 管理ツールの起動](#)」の説明に従って、EMS 管理ツールを起動します。



Windows が起動しているコンピュータで、スタートメニューから [プログラム] > [TIBCO] > [TIBCO EMS 7.0] > [Start EMS administration tool] を選択して、管理ツールを起動することもできます。

EMS 管理ツールを起動したら、それを EMS サーバーに接続します。

EMS 管理ツールを EMS サーバーに接続するには、以下のいずれかのコマンドを実行します。

- 1 台のコンピュータで TIBCO Enterprise Message Service を使用する場合は、管理ツールのコマンドラインで、次のように **connect** と入力します。  
> connect

ログイン名の入力を求められます。初めて EMS 管理ツールを使用する場合は、120 ページの「[tibemsadmin の最初の起動](#)」で説明されている手順に従います。

ログインすると、次のような画面が表示されます。

```
connected to tcp://localhost:7222
tcp://localhost:7222>
```

- ネットワーク内の TIBCO Enterprise Message Service を使用する場合は、以下のコマンドを使用してサーバーに接続します。

```
> connect [server URL] [user-name] [password]
```

このコマンドについての詳細は、[124 ページの「connect」](#)を参照してください。

管理ツールについての詳細は、118 ページの「[EMS 管理ツールの起動](#)」と 122 ページの「[コマンドのリスト](#)」を参照してください。

## ユーザーの作成

管理ツールをサーバーに接続したら、`create user` コマンドを使用してふたりのユーザーを作成します。

管理ツールで、次のように入力します。

```
tcp://localhost:7222> create user user1
tcp://localhost:7222> create user user2
```

管理ツールの画面に、`user1` と `user2` が作成されたことを示すメッセージ (`created`) が表示されます。

これで、ふたりのユーザーが作成されました。次のように `show users` コマンドを使って、作成したユーザーを確認することができます。

```
tcp://localhost:7222> show users
User Name      Description
user1
user2
```

`create user` コマンドについての詳細は、[127 ページの「create user」](#)を参照してください。

## ポイントツーポイントメッセージングの例

このセクションでは、3 ページの「ポイントツーポイント」で説明されている、ポイントツーポイントメッセージングの使い方について説明します。

### キューの作成

ポイントツーポイントメッセージングモデルでは、クライアントはキューとの間でメッセージを送受信します。

`myQueue` という名前の新規キューを管理ツールで作成するには、`create queue` コマンドを使用します。

```
tcp://localhost:7222> create queue myQueue
```

`create queue` コマンドについての詳細は、126 ページの「`create queue`」を参照してください。`commit` コマンドについての詳細は、123 ページの「`commit`」と 123 ページの「`autocommit`」を参照してください。

### セNDERクライアントとレシーバクライアントの起動

1. ふたつのコマンドラインウィンドウを開き、各ウィンドウから `EMS_HOME/samples/java` ディレクトリに移動します。
2. 各コマンドラインウィンドウで、次のように `setup` と入力して、環境変数とクラスパスを設定します。  

```
> setup
```
3. 最初のコマンドラインウィンドウで、`tibjmsMsgProducer` アプリケーションを実行し、`myQueue` キューにメッセージを入れるよう `user1` に指示します。  

```
> java tibjmsMsgProducer -queue myQueue -user user1 Hello User2
```
4. 2 番目のコマンドラインウィンドウで、`tibjmsMsgConsumer` クライアントを実行し、メッセージキューからメッセージを読み取るよう `user2` に指示します。  

```
> java tibjmsMsgConsumer -queue myQueue -user user2
```

キューに入れられたメッセージは、レシーバの画面に表示されます。



セNDERがキューに入れたメッセージは、レシーバによって読み取られるまでキュー内に残るため、セNDERとレシーバのどちらのクライアントから先に起動してもかまいません。

## パブリッシュとサブスクライブメッセージングの例

このセクションでは、メッセージプロデューサのクライアントとふたつのメッセージコンシューマクライアントを実行し、パブリッシュ / サブスクライブのメッセージングモデルを説明します (4 ページの「[パブリッシュとサブスクライブ](#)」を参照)。この例は、堅牢なアプリケーションを包括的に説明するためのものでもなく、代表例として説明するためのものでもありません。

クライアントのサンプルを実行するには、コンパイルされたサンプルを含むサンプルディレクトリで、それらのサンプルクライアントを実行します。この演習では、三つのコマンドラインウィンドウを開き、各ウィンドウから

`EMS_HOME/samples/java` ディレクトリに移動します。

サンプルについての詳細は、サンプルディレクトリにある `readme` ファイルを参照してください。サンプルのコンパイルについての詳細は、89 ページの「[サンプルクライアントのコンパイル](#)」を参照してください。

### トピックの作成

パブリッシュ / サブスクライブモデルでは、トピックをパブリッシュしてサブスクライブします。

`myTopic` という名前の新規トピックを管理ツールで作成するには、`create topic` コマンドを使用します。

```
tcp://localhost:7222> create topic myTopic
```

`create topic` コマンドについての詳細は、[127 ページの「create topic」](#)を参照してください。`commit` コマンドについての詳細は、[123 ページの「commit」](#)と [123 ページの「autocommit」](#)を参照してください。

### サブスクライバクライアントの起動

パブリッシャを起動したときに、受信されるメッセージを監視できるようになるため、最初にサブスクライバを起動します。

`user1` をサブスクライバとして起動するには、以下の手順に従います。

1. 最初のコマンドラインウィンドウから、`EMS_HOME/samples/java` ディレクトリに移動します。

2. 次のように `setup` と入力して、環境変数とクラスパスを設定します。

```
> setup
```

3. `tibjmsMsgConsumer` クライアントを実行し、`user1` を `myTopic` トピックのサブスクライバに割り当てます。

```
> java tibjmsMsgConsumer -topic myTopic -user user1
```

`user1` が `myTopic` をサブスクライブしていることを示すメッセージが管理ツールの画面に表示されます。

`user2` をサブスクライバとして起動するには、以下の手順に従います。

1. 2 番目のコマンドラインウィンドウから `EMS_HOME/samples/java` ディレクトリに移動します。
2. 次のように `setup` と入力して、環境変数とクラスパスを設定します。  
`> setup`
3. `tibjmsMsgConsumer` アプリケーションを実行し、`user2` を `myTopic` トピックのサブスクライバに割り当てます。  
`> java tibjmsMsgConsumer -topic myTopic -user user2`

`user2` が `myTopic` をサブスクライブしていることを示すメッセージが管理ツールの画面に表示されます。



サブスクライバの実行中は、コマンドウィンドウはプロンプトに戻りません。

## パブリッシャクライアントの起動とメッセージの送信

パブリッシャの設定は、サブスクライバの設定とよく似ています。異なる点は、サブスクライバがトピック名とユーザー名を必要とするのに対して、パブリッシャはメッセージも必要とします。

パブリッシャを起動するには、以下の手順に従います。

1. 3 番目のコマンドラインウィンドウから `EMS_HOME/samples/java` ディレクトリに移動します。
2. 次のように `setup` と入力して、環境変数とクラスパスを設定します。  
`> setup`
3. `tibjmsMsgProducer` クライアントを実行し、`myTopic` トピックにメッセージをパブリッシュするよう、`user1` に指示します。  
`> java tibjmsMsgProducer -topic myTopic -user user1 hello user2`  
 ここで、`'hello'` と `'user2'` は異なるメッセージです。



この例では、`user1` をパブリッシャとサブスクライバの両方にしています。

コマンドラインウィンドウに、両方のメッセージがパブリッシュされたことを示すメッセージが表示されます。

```
Publishing on topic 'myTopic'
Published message: hello
Published message: user2
```

メッセージがパブリッシュされると、パブリッシャのコマンドウィンドウはプロンプトに戻り、続けてさらにメッセージをパブリッシュすることができます。



次の形式を使用する場合は、次の点に注意します。

```
java tibjmsMsgProducer -topic myTopic -user user1
```

メッセージを追加しないと、メッセージテキストを少なくともひとつ作成する必要があることを警告するエラーメッセージが表示されます。

サブスクライバを含む最初と 2 番目のコマンドラインウィンドウには、次のように、各サブスクライバがふたつのメッセージを受け取ったことが示されます。

```
Subscribing to topic: myTopic
Received message: TextMessage={ Header={
JMSMessageID={ID:EMS-SERVER.97C44203CDF A:1}
JMSDestination={Topic[myTopic]} JMSReplyTo={null}
JMSDeliveryMode={PERSISTENT} JMSRedelivered={false}
JMSCorrelationID={null} JMSType={null} JMSTimestamp={Tue Mar 21
12:04:56 PST 2006} JMSExpiration={0} JMSPriority={4} }
Properties={ } Text={hello} }
Received message: TextMessage={ Header={
JMSMessageID={ID:EMS-SERVER.97C44203CDFA:2}
JMSDestination={Topic[myTopic]} JMSReplyTo={null}
JMSDeliveryMode={PERSISTENT} JMSRedelivered={false}
JMSCorrelationID={null} JMSType={null} JMSTimestamp={Tue Mar 21
12:04:56 PST 2006} JMSExpiration={0} JMSPriority={4} }
Properties={ } Text={user2} }
```

## セキュアトピックの作成

この例では、myTopic を secure なトピックにし、user1 に myTopic にパブリッシュするアクセス権限を許可し、user2 に myTopic をサブスクライブするアクセス権限を許可します。

### トピックへの secure プロパティの追加

トピックに secure プロパティを追加すると、特定のアクセス権限が割り当てられているユーザーのみがそのアクセス権限に許可されたアクションを実行できるようになります。たとえば、トピックにパブリッシュする権限のあるユーザーだけがパブリッシュでき、それ以外のユーザーはパブリッシュすることはできません。

トピックに secure プロパティを追加しない場合、認可されたすべてのユーザーが、そのトピックに対してすべてのアクセス権限（パブリッシュ、サブスクライブ、持続サブスクライバの作成）を持つようになります。

secure プロパティについての詳細は、[67 ページの「secure」](#)のセクションを参照してください。トピックのアクセス権限についての詳細は、255 ページの[第 8 章、「認証とアクセス権限」](#)を参照してください。

サーバー認可を有効にし、**secure** プロパティをトピックに追加するには、以下の手順に従います。

1. 各サブスクライバウィンドウで **Ctrl + C** キーを押して、各サブスクライバを停止します。
2. 管理ツールで、**set server** コマンドを使って次のように入力し、**authorization** プロパティを有効にします。

```
tcp://localhost:7222> set server authorization=enabled
```

**authorization** プロパティを有効にすると、デスティネーションに設定されているアクセス権限がチェックされるようになります。

3. 次のコマンドを入力して、**secure** プロパティを **myTopic** という名前のトピックに追加します。

```
tcp://localhost:7222> addprop topic myTopic secure
```

**set server** コマンドについての詳細は、[137 ページの「set server」](#)を参照してください。**addprop topic** コマンドについての詳細は、[123 ページの「addprop topic」](#)を参照してください。

## ユーザーへのトピックアクセス権限の設定

アクセス権限がメッセージのパブリッシュと受信に対してどのような影響を持っているかを確認するために、パブリッシュ権限を **user1** に付与し、サブスクライブ権限を **user2** に付与します。

以下の手順に従い、**grant topic** コマンドを使用して、**myTopic** トピックへのアクセス権限をユーザーに付与します。

管理ツールで、次のように入力します。

```
tcp://localhost:7222> grant topic myTopic user1 publish
tcp://localhost:7222> grant topic myTopic user2 subscribe
```

**grant topic** コマンドについての詳細は、[131 ページの「grant topic」](#)を参照してください。

## サブスクライバクライアントとパブリッシャクライアントの起動

93 ページの「[サブスクライバクライアントの起動](#)」の説明に従って、サブスクライバを起動します。**user1** にはパブリッシュするアクセス権限が付与されていますが、サブスクライブするアクセス権限が付与されていないため、**user1** をサブスクライバとして起動することはできません。その結果、以下の例外メッセージを受け取ります。

```
Operation not permitted.
```

上記の方法に従って、**User2** をサブスクライバとして起動します。



これで、94 ページの「パブリッシュクライアントの起動とメッセージの送信」で説明されているように、`user1` をパブリッシャーとして起動し、`user2` にメッセージを送れるようになりました。

## 持続サブスクライバの作成

デフォルトでは、メッセージを受信するのはアクティブなサブスクライバのみです (4 ページの「パブリッシュとサブスクライブ」を参照)。メッセージがパブリッシュされたときにサブスクライバがアクティブでない場合、サブスクライバはそれらのメッセージを受信しません。サブスクリプションをサーバーに保存する持続サブスクライバを作成することで、メッセージが配信されたときにサブスクライバがアクティブでなかった場合でも、サブスクライバがメッセージを受け取れるようにすることができます。

この例では、EMS サーバー上の `myTopic` トピックにパブリッシュされたメッセージを保存する持続サブスクライバを作成します。

`user2` を持続サブスクライバとして起動するには、以下の手順に従います。

1. コマンドラインウィンドウから `EMS_HOME/samples/java` ディレクトリに移動します。

2. 次のように `setup` と入力して、環境変数とクラスパスを設定します。

```
> setup
```

3. `tibjmsDurable` アプリケーションを実行し、`user2` を `myTopic` トピックの持続サブスクライバに割り当てます。

```
> java tibjmsDurable -topic myTopic -user user2
```

4. 管理ツールで、`show durables` コマンドを使って次のように入力し、`user2` が `myTopic` の持続サブスクライバであることを確認します。

```
tcp://localhost:7222> show durables
Topic Name      Durable      User      Msgs      Size
* myTopic       subscriber    user2      0          0.0 Kb
```

5. サブスクライバウィンドウで `Ctrl + C` キーを押して、サブスクライバを停止します。

6. 別のコマンドラインウィンドウで、`tibjmsMsgProducer` クライアントを実行します (94 ページの「パブリッシュクライアントの起動とメッセージの送信」を参照)。

```
> java tibjmsMsgProducer -topic myTopic -user user1 hello user2
```

7. サブスクライバを再起動します。

```
> java tibjmsDurable -topic myTopic -user user2
```

サブスクライバウィンドウに、保存されたメッセージが表示されます。

8. `Ctrl + C` キーを押してサブスクライバを停止し、持続サブスクリプションの登録を解除します。

```
> java tibjmsDurable -unsubscribe
```

サブスクライバは持続サブスクライバでなくなり、`myTopic` トピックにパブリッシュされた追加メッセージは失われます。

## マルチキャストメッセージングの例

---

このセクションでは、5 ページの「[マルチキャスト](#)」で説明されている、マルチキャストメッセージングの使い方について説明します。

この例では、マルチキャストメッセージをサーバーからブロードキャストできるよう、EMS サーバーでマルチキャストを有効にし、マルチキャストチャンネルを設定します。また、`multicastTopic` という名前でマルチキャスト対応のトピックを作成し、そのトピックをマルチキャストチャンネルに関連付けて、マルチキャストで `multicastTopic` にパブリッシュされたメッセージをサブスクライバが受信できるようにします。

マルチキャストチャンネルを設定するには、設定ファイルの修正による静的な設定を行う必要があります。マルチキャストチャンネルを設定するためのコマンドは管理ツールに用意されていません。

### EMS サーバーの停止

管理ツールの `shutdown` コマンドを使用してサーバーを停止します。

```
tcp://localhost:7222> shutdown
```

マルチキャストの設定が完了すると、サーバーの再起動を促すメッセージが表示されます。

### EMS サーバーのマルチキャストの有効化

サーバーでマルチキャストを有効にするには、`tibemsd.conf` 設定ファイルの `multicast` プロパティを `enabled` に設定します。

```
multicast = enabled
```

### マルチキャストチャンネルの作成

EMS サーバーは、マルチキャストチャンネル経由でメッセージをコンシューマにブロードキャストします。チャンネルには、それぞれマルチキャストアドレスとポートを定義します。マルチキャスト対応のトピックにメッセージをパブリッシュすると、メッセージはこれらのマルチキャストチャンネル経由でサーバーから送信され、サブスクライバに届けられます。

マルチキャストチャンネルを作成するには、マルチキャストチャンネル設定ファイルの `channels.conf` に次の定義を追加します。

```
[multicast-1]  
address=234.5.6.7:1
```

## EMS サーバーの起動

101 ページの「[EMS サーバーの実行](#)」の説明に従って、EMS サーバーを起動します。



Windows が起動しているコンピュータで、スタートメニューから [プログラム] > [TIBCO] > [TIBCO EMS 7.0] > [Start EMS server] を選択して、EMS サーバーを起動することもできます。

管理ツールで `show topics` コマンドを実行し、M 列に「+」が表示されていれば、`multicastTopic` はマルチキャストが有効になっています。

```
tcp://localhost:7222> show topics
Topic Name          SNFGEIBCTM    Subs    Durs    Msgs    Size
multicastTopic      -----+      0        0        0      0.0 Kb
```

## トピックのマルチキャストの有効化

トピックをマルチキャスト対応にするには、`channel` プロパティでトピックをマルチキャストチャンネルに関連付ける必要があります。

マルチキャスト対応トピックを作成するには、管理ツールで次のコマンドを実行します。

```
> create topic multicastTopic channel=multicast-1
```

## マルチキャスト デーモンの起動

367 ページの「[マルチキャスト デーモンの起動](#)」の説明に従って、マルチキャスト デーモン を起動します。

## サブスクライバクライアントの起動

マルチキャストサブスクライバの作成手順は、セッションの確認応答モードに `com.tibco.tibjms.Tibjms.NO_ACKNOWLEDGE` を使用する必要がある点を除けば、マルチキャストでないサブスクライバの作成と同じです。

`user1` をマルチキャストサブスクライバとして起動するには、以下の手順に従います。

1. コマンドラインウィンドウから、`EMS_HOME/samples/java` ディレクトリに移動します。
2. 次のように `setup` と入力して、環境変数とクラスパスを設定します。  

```
> setup
```

3. `tibjmsMsgConsumer` クライアントを実行し、セッションの確認応答モードを `NO_ACKNOWLEDGE` にして `user1` を `multicastTopic` トピックのサブスクライバに割り当てます。

```
> java tibjmsMsgConsumer -topic multicastTopic -user user1 -ackmode NO
```

4. 管理ツールで `show consumers` コマンドを実行し、**M** 列に「+」が表示されていれば、`user1` は `multicastTopic` のマルチキャストサブスクライバになっています。

```
tcp://optimist:7222> show consumers topic=multicastTopic
```

Id	Conn	User	T	Topic	SASNM	Msgs	Pend Size	Pend Uptime
2	4	user1	T	multicastTopic	+N--+	0	0	0:03:17

## パブリッシュクライアントの起動とメッセージの送信

クライアントでマルチキャストメッセージをパブリッシュするための設定は、パブリッシュメッセージとサブスクライブメッセージを送信するための設定と同様です。トピックのマルチキャストの有効化は EMS サーバーで設定するため、メッセージプロデューサで他の設定手順を実行する必要ありません。

メッセージパブリッシャを作成するには、以下の手順に従います。

1. 新しいコマンドラインウィンドウから、`EMS_HOME/samples/java` ディレクトリに移動します。
2. 次のように `setup` と入力して、環境変数とクラスパスを設定します。

```
> setup
```

3. `tibjmsMsgProducer` クライアントを実行し、`multicastTopic` トピックにメッセージをパブリッシュするよう、`user1` に指示します。

```
> java tibjmsMsgProducer -topic multicastTopic -user user1 hello multicast
```

ここで、`'hello'` と `'multicast'` は異なるメッセージです。



この例では、`user1` をパブリッシャとサブスクライバの両方にしています。

メッセージはサブスクライバのウィンドウに表示されます。

## 第 5 章

# EMS サーバーの実行

TIBCO Enterprise Message Service を既存のアプリケーションと統合して使用するには、TIBCO Enterprise Message Service サーバーが実行している必要があります。サーバーとクライアントとが協調して、TIBCO Enterprise Message Service を実装します。サーバー上にはあらゆるタイプのメッセージが保持され、クライアント側にはメッセージは保存されません。

### トピック

---

- [「EMS サーバーの起動と停止」、102 ページ](#)
- [「Windows サービスとしての EMS サーバーの起動」、106 ページ](#)
- [「エラーの回復ポリシー」、109 ページ](#)
- [「セキュリティに関する注意事項」、110 ページ](#)
- [「EMS による共有ストアファイルへのアクセスの管理方法」、114 ページ](#)

## EMS サーバーの起動と停止

このセクションでは、EMS サーバーの起動および停止の方法について説明します。



Microsoft Windows が起動しているコンピュータで、スタートメニューから [ プログラム ] > [ TIBCO ] > [ TIBCO EMS 7.0 ] > [ Start EMS Server ] を選択して、EMS サーバーを起動できます。



Windows システムでは、`tibemsd.exe` ファイルと `tibemsmcd.exe` ファイルが管理者として実行され、マルチキャスト機能が有効になり、`tibemsd.exe` からの設定ファイル修正が可能になります。

### デフォルト設定での EMS サーバーの起動

コマンドラインからあらかじめセットアップされている内容で EMS サーバーを起動するには、`EMS_HOME/bin` に移動して次のスクリプトを実行します。

UNIX の場合      `tibemsd.sh`

Windows の場合      `tibemsd.bat`

スクリプトを実行すると、デフォルトの場所である `config-file-directory¥cfmgmt¥ems¥data` ディレクトリの設定ファイルを使って EMS サーバーが起動します。`config-file-directory` はインストール時に指定した設定ディレクトリです。

### JSON 設定を使った EMS サーバーの起動



Central Administration 機能を使用するユーザーは、JSON モードで EMS サーバーを起動する必要があります。JSON モードで起動するには、コマンドラインで `-config` オプションを使って JSON 設定ファイルを指定します。詳細は、『TIBCO Enterprise Message Service Central Administration ガイド』の JSON Configuration Files を参照してください。

JSON を設定した `tibemsd` は、Central Administration サーバーリストに追加する前に実行しておく必要があります。JSON 設定ファイルで TIBCO Enterprise Message Service サーバーを起動するには、以下の手順に従います。

1. コマンドラインウィンドウから `EMS_HOME/bin` に移動します。
2. 次のコマンドとオプションを入力します。

```
tibemsd -config json-file-path
```

ここで、`json-file-path` は JSON 設定ファイルへのパスです。例：

```
tibemspd -config /tibemspdconfig/tibemspd.json
```

JSON 設定を使用して `tibemspd` を起動するとき、不明なパラメータは何もせずに無視されます。たとえば、`tibemspd.json` ファイルに廃止されたパラメータが含まれていても、エラーはスローされません。



.conf 設定ファイルから JSON 設定ファイルへの変換についての詳細は、『TIBCO Enterprise Message Service Central Administration ガイド』の **Converting Server Configuration Files to JSON** を参照してください。

## フォールトトレランスサーバーのペアの起動

Central Administration では、ひとつの JSON 設定ファイルをフォールトトレランスのペアが共有します。プライマリサーバーの役割とセカンダリサーバーの役割は、サーバーの起動時に決定します。

プライマリ EMS サーバーは通常どおり起動します。セカンダリサーバーは `-secondary` フラグを使用して起動します。たとえば、JSON 設定ファイルが `tibemspd.json` の場合は、次のように指定します。

プライマリサーバーの起動: `tibemspd -config tibemspd.json`

セカンダリサーバーの起動: `tibemspd -config tibemspd.json -secondary`

詳細は、490 ページの「[Central Administration](#) でのフォールトトレランスの設定」を参照してください。

## オプションを使用した EMS サーバーの起動

コマンドラインからオプションを使用して EMS サーバーを起動するには、以下の手順に従います。

### タスク A データサブディレクトリに移動します。

あらかじめ設定されている EMS サーバーファイルは `config-file-directory¥cfmgmt¥ems¥data` ディレクトリにあります。`config-file-directory` はインストール時に指定した設定ディレクトリです。詳細は、『TIBCO Enterprise Message Service のインストール』の **Installing TIBCO Enterprise Message Service** を参照してください。

インストールした `data` ディレクトリに移動します。

UNIX の場合 `./tibemspd -config "config-file-directory/cfmgmt/ems/data/tibemspd.conf"`

Windows の場合 `¥tibemspd -config "config-file-directory¥cfmgmt¥ems¥data¥tibemspd.conf"`

または、`EMS_HOME¥samples¥config` ディレクトリに移動し、データストアディレクトリ（または必要に応じてその他のディレクトリ）を移動先に作成して、そこでサンプル設定ファイルを使用します。

EMS サーバーは SSL と圧縮共有ライブラリを動的に読み込みます（静的にリンクしません）。`tibemsd` 実行可能ファイルを `data` ディレクトリから実行すると、ライブラリを自動的に検索します。サーバーを他の場所に移動する場合は、共有ライブラリのディレクトリも移動する必要があります。

タスク Btibemsd の起動

`tibemsd [options]` と入力します。

*options* については、表 14 を参照してください。`tibemsd` のコマンドオプションは、`tibemsd.conf` ファイルで設定したパラメータと似ていますが、パラメータで指定されている値よりもこのコマンドオプションの値が優先されます。設定パラメータについての詳細は、177 ページの「`tibemsd.conf`」を参照してください。

表 14 tibemsd オプション

オプション	説明
<code>-config config file name</code>	<code>config file name</code> には <code>tibemsd</code> サーバーのメイン設定ファイルの名前を指定します。デフォルトは <code>tibemsd.conf</code> です。  たとえば、デフォルトの JSON 設定ファイルを使用して EMS サーバーを起動するには、以下の手順に従います。  <code>tibemsd -config tibemsd.json</code>
<code>-trace items</code>	トレースアイテムを指定します。これらのアイテムは設定ファイルには含まれていません。この値は、管理ツールの <code>set server</code> コマンドで指定した <code>log_trace</code> パラメータの値と同じ形式です（435 ページの「サーバー上でのトレース」を参照）。
<code>-secondary</code>	フォールトトレランスのペアのセカンダリサーバーを指定します。このオプションは、JSON 設定で EMS サーバーを起動した場合にのみ有効です。
<code>-ssl_password string</code>	秘密鍵のパスワード。
<code>-ssl_trace</code>	サーバーに読み込まれている証明書を表示し、SSL 関連の状況をさらに詳しくトレースします。
<code>-ssl_debug_trace</code>	SSL 接続のトレースをオンにします。
<code>-ft_active active_url</code>	アクティブなサーバーの URL。アクティブなサーバーにこのサーバーが接続できた場合、このサーバーはバックアップサーバーとして機能します。アクティブなサーバーにこのサーバーが接続できなかった場合、このサーバーがアクティブなサーバーになります。



表 14 tibemsd オプション (続き)

オプション	説明
-ft_heartbeat <i>seconds</i>	アクティブなサーバーのハートビート信号 (秒単位)。デフォルトは 3 です。
-ft_activation <i>seconds</i>	アクティブなサーバーで障害が発生したことを示すアクティブ化間隔 (ハートビート信号間の間隔の最大値) を指定します。秒単位で設定し、デフォルトは 10 秒です。この間隔には、ハートビート間隔の少なくとも 2 倍の値を設定する必要があります。
-forceStart	<p>ストアファイルの破損メッセージが削除され、エラーが発生してもサーバーを起動できるようになります。</p> <p>このオプションを使用するとデータが消失するため、<b>-forceStart</b> を使うときは、あらかじめストアファイルをバックアップしておくことが重要です。詳細は、109 ページの「<a href="#">エラーの回復ポリシー</a>」を参照してください。</p>

## EMS サーバーの停止

EMS サーバーを停止するには、EMS 管理ツールから **shutdown** コマンドを実行します。

## Windows サービスとしての EMS サーバーの起動

場合によっては、EMS サーバーとマルチキャスト デーモン を自動起動させる必要があります。Windows のサービスマネージャにこれらを登録することで、この要求を満たすことができます。登録するには、`emsntsrv` ユーティリティを使用します。

### emsntsrv

`emsntsrv` ユーティリティを使用すると、EMS サーバーのデーモンまたは EMS マルチキャスト デーモン を Windows サービスとして登録または登録解除します。



このユーティリティは、Microsoft Windows（2003、XP、Vista など、サポートされるすべてのバージョン）にのみ適用されます。

- 構文

`emsntsrv /i [/a] service_name emsntscd_directory service_directory [arguments] [suffix]`  
`emsntsrv /r [service_name] [suffix]`
- 備考

場合によっては、EMS サーバープロセスを自動起動させる必要があります。Windows のサービスマネージャにこれらを登録することで、この要求を満たすことができます。このユーティリティはレジストリを利用します。
- 制限事項

Windows レジストリを変更するには管理者権限が必要です。
- 場所

このユーティリティプログラムは、EMS の `bin` ディレクトリにある実行可能ファイルです。

パラメータ	説明
<code>/i</code>	レジストリに新しいサービスを挿入します（新しいサービスの登録）。
<code>/a</code>	新しいサービスを自動的に開始します。 <code>/i</code> を指定するときのオプション。
<code>/?</code>	使用方法を表示します。
<code>service_name</code>	この名前に基づいてサービスを挿入または削除します。  サービスを挿入する場合、このパラメータに <code>tibemsd</code> または <code>tibemsmcd</code> を指定する必要があります。  サービスを削除する場合、このパラメータの指定は省略できます。ただし、指定する場合は、 <code>tibemsd</code> または <code>tibemsmcd</code> を指定する必要があります。

パラメータ	説明
<i>emsntsct_directory</i>	このディレクトリのパス名を使用して、 <b>emsntsct.exe</b> 実行可能ファイルの場所を指定します。 <b>emsntsrg</b> ユーティリティは、 <b>emsntsct.exe</b> プログラムを Windows サービスとして登録します。登録された <b>emsntsct.exe</b> プログラムは、関連する <b>tibemsd</b> または <b>tibemsmcd</b> を呼び出します。  デフォルトでは、 <b>emsntsct.exe</b> は <b>EMS_HOME¥bin</b> にあります。  このパラメータは、サービスをインストールする場合にのみ必要です。
<i>service_directory</i>	このディレクトリのパス名を使用して、実行可能ファイルである <b>tibemsd</b> または <b>tibemsmcd</b> を検索します。必須。
<i>arguments</i>	コマンドライン引数を指定します。/i を指定するときのオプション。  引数の文字列全体を二重引用符で囲みます。
<i>suffix</i>	サービスのインスタンスを複数登録する場合、このサフィックスを使って、Windows サービスアプレットで各サービスを識別できるようにします。オプション。
/r	レジストリからサービスを削除します。

**登録**      **tibemsd** を Windows サービスとして登録するには、次のコマンドラインでユーティリティを実行します。

```
emsntsrg /i [/a] tibemsd emsntsct_directory tibemsd_directory [arguments] [suffix]
```

**tibemsmcd** を Windows サービスとして登録するには、次のコマンドラインでユーティリティを実行します。

```
emsntsrg /i [/a] tibemsmcd emsntsct_directory tibemsmcd_directory [arguments] [suffix]
```

- 例 1 :**      この簡単な例では、ひとつの **tibemsd** サービスを登録します。

```
emsntsrg /i tibemsd C:¥tibco¥ems¥7.0¥bin C:¥tibco¥ems¥7.0¥bin
```
- 例 2 :**      この例では、コマンドライン引数を使ってサービスを登録します。

```
emsntsrg /i tibemsd C:¥tibco¥ems¥7.0¥bin C:¥tibco¥ems¥7.0¥bin  
"-trace DEFAULT"
```
- 例 3 :**      このペアのコマンド例では、ふたつの **tibemsd** サービスをそれぞれ異なる設定ファイルに登録します。この例では、数字のサフィックスと設定ディレクトリの両方に、サービスで使用するポート番号が反映されています。

```
emsntsrg /i tibemsd C:¥tibco¥ems¥7.0¥bin C:¥tibco¥ems¥7.0¥bin  
"-config C:¥tibco¥ems¥7.0¥7222¥tibemsd.conf" 7222
```

```
emsntsrg /i tibemsd C:¥tibco¥ems¥7.0¥bin C:¥tibco¥ems¥7.0¥bin  
"-config C:¥tibco¥ems¥7.0¥7223¥tibemsd.conf" 7223
```

この例では次の点に注意します。

- `tibemspd` のインストール時に `-config` 引数を指定すると、主要設定ファイル (`tibemspd.conf`) が含まれているディレクトリがサービスのプロセスによって検出され、そのディレクトリにすべてのセカンダリ設定ファイルが作成されます。この例では、各サービスはそれぞれ異なる設定ディレクトリを使用しています。
- 複数の EMS サービスを登録する場合は、サービスどうしの設定の競合を避ける必要があります。たとえば、`tibemspd` のふたつのインスタンスは同じポートをリスンすることはできません。

**例 4 :** この例では、ひとつのマルチキャスト デーモン サービスを登録します。

```
emsntsrg /i tibemsmcd C:¥tibco¥ems¥7.0¥bin C:¥tibco¥ems¥7.0¥bin
```

**例 5 :** この例では、コマンドライン引数を使ってマルチキャスト デーモン サービスを登録します。

```
emsntsrg /i tibemsmcd C:¥tibco¥ems¥7.0¥bin C:¥tibco¥ems¥7.0¥bin
"-logfile c:¥tibemsmcd.log"
```

ログファイルを指定すると、マルチキャスト デーモン サービスを起動できない場合に、その原因となっている競合を特定するのに役立つ場合があります。

**例 6 :** このペアのコマンド例では、ふたつのマルチキャスト デーモン サービスをそれぞれ異なるポートに登録します。この例では、サービスで使用するポート番号が数字のサフィックスで示されています。

```
emsntsrg /i tibemsmcd C:¥tibco¥ems¥7.0¥bin C:¥tibco¥ems¥7.0¥bin
"-listen 7345" 7345
```

```
emsntsrg /i tibemsmcd C:¥tibco¥ems¥7.0¥bin C:¥tibco¥ems¥7.0¥bin
"-listen 7346" 7346
```

**Remove** サービスの登録を解除するには、次のコマンドラインでユーティリティを実行します。

```
emsntsrg /r [service_name] [suffix]
```

パラメータの指定はどちらもオプションです。`service_name` を指定する場合は、`tibemspd` または `tibemsmcd` を指定する必要があります。`suffix` パラメータを指定する場合は、`service_name` も指定する必要があります。どちらのパラメータも指定しない場合は、`tibemspd` と `tibemsmcd` という名前のサービスがユーティリティによって削除されます。

**コマンドの概要** コマンドラインの概要を表示するには、次のコマンドラインでユーティリティを実行します。

```
emsntsrg
```

**Windows サービスアプレット** Windows サービスアプレットには、登録されている各サービスの名前が表示されます。EMS サービスについては、次の追加情報が表示されます。

- サフィックス (指定した場合)
- プロセス ID (PID) (サービスが実行中の場合)

## エラーの回復ポリシー

EMS サーバーの起動中、ストアファイルからの情報回復時にいくつかのエラーが発生する場合があります。発生する可能性のあるエラーには、次のようなものがあります。

- ファイルのローレベルエラー。たとえば、ディスクレコードの破損などです。
- オブジェクト固有のローレベルエラー。たとえば、必要なフィールドが欠落しているレコードです。
- オブジェクト内部のエラー。たとえば、対応する接続レコードが存在しないセッションレコードなどです。

起動中の EMS サーバーでこのようなエラーのいずれかが発生した場合の回復ポリシーは、次のとおりです。

- デフォルトでは、ディスクレコードの破損エラーが検出されると、サーバーの起動が完全に停止されます。これは、サーバーが安全に状態を復元できないと、残りの回復処理を続行できないためです。起動の停止後、設定にエラーがないかを確認します。必要に応じて、ストアファイルと設定ファイルをコピーして TIBCO サポートに検証するよう要求してください。
- `-forceStart` コマンドラインオプションを使って、サーバーに不良レコードの削除を指示します。これによって、サーバーのランタイム状態の破損を防止します。
- 起動中にメモリ不足が発生した場合、サーバーは停止します。

**重要：** `-forceStart` オプションを使ってサーバーを再起動する前に、ストアファイルのバックアップを行ってください。問題があるレコードを削除するとデータも消失します。

ストアファイルには、さまざまなレコードが保存されます。アプリケーションから送信される持続 JMS メッセージはもちろんですが、その他の内部レコードも保存されています。持続サブスクライバの状態情報を保存するために使用されるコンシューマレコードが破損し、該当するレコードを `-forceStart` オプションで後から削除した場合、保存されていたすべての JMS メッセージが同時に（メッセージ自体は破損しておらず、有効であっても）失われてしまいます。これは、持続サブスクライバ情報が回復不能なためです。

このモードでサーバーを実行すると、回復処理中に見つかったエラーはすべて報告され、問題のあるレコードが削除された上で回復処理が続行されます。このモードでは、`-forceStart` オプションを指定しない場合よりも多数の問題が報告される場合があります。これは同オプションを指定しないと、最初のエラー検出時にサーバーが停止するからです。



`-forceStart` オプションを使ってサーバーを再起動する前に、すべてのストアファイルをバックアップすることをお勧めします。`-forceStart` オプションの指定によって削除されたレコードが何かを事後解析するときにバックアップが役立ちます。

## セキュリティに関する注意事項



このセクションでは、セキュアな実装に関連する情報を主に説明します。管理者は全員、このセクションを一読するようお勧めします。

### セキュアな環境

セキュアな実装を確実に行うには、EMS の管理で次の基準を満たす必要があります。

- **適切なインストール**：EMS を正しくインストールして、設定します。
- **物理的制御**：不正アクセスを避けるため、EMS がインストールされているコンピュータを、物理的エントリが制御される領域に配置します。適切な権限のある管理者だけがアクセスでき、管理者が健全な環境を共有します。
- **ドメイン管理**：サーバー、設定ファイル、LDAP サーバーなどの不正アクセスを避けるため、オペレーティングシステム、ファイルシステム、およびネットワークプロトコルなどによって、EMS のドメイン領域を確実に切り離します。
- **健全な環境**：適切な権限のある管理者だけが物理的なアクセスまたはドメインアクセスを行い、管理者が健全な環境を共有します。

### デスティネーションのセキュリティ

相互に作用する三つの要素が、トピックとキューというデスティネーションのセキュリティに影響します。セキュアな実装では、これら三つのアイテムをすべて適切に設定する必要があります。

- サーバーの `authorization` パラメータ (以下の [authorization パラメータ](#) を参照)
- 各デスティネーションの `secure` プロパティ (67 ページの「[secure](#)」を参照)
- 各デスティネーションに適用される ACL アクセス権限 (255 ページの「[認証とアクセス権限](#)」を参照)

### authorization パラメータ

サーバーの `authorization` パラメータは、セキュアなデスティネーションへの接続要求と操作を可能にするアクセス権限を確認するためのマスタースイッチの役割を果たします。このパラメータのデフォルト値は `disabled` です。サーバーはアクセス権限を確認せず、あらゆる操作を許可します。セキュアな実装にするには、このパラメータを有効にする必要があります。

## Admin パスワード

インストールと初期テストを簡単にするため、**admin** パスワードのデフォルト設定ではパスワードは設定されていません。実際のパスワードが設定されるまで、**admin** ユーザーは、パスワードを使わずにログインできます。管理者のパスワードを設定すると、サーバーから常にそのパスワードの入力が要求されます。

セキュアな実装を設定するには、管理者はインストール後、直ちに **admin** パスワードを変更する必要があります（120 ページの「[管理者へのパスワードの割り当て](#)」を参照）。

## 接続セキュリティ

**authorization** を有効にすると、ユーザーが接続できるようサーバーはユーザー名とパスワードの入力を求めます。適切な権限のあるユーザーのみがサーバーに接続できます。認証の形式としては、X.509 証明書またはユーザー名とパスワードの組み合わせのいずれか（または両方）を使用できます。

**authorization** プロパティが無効に設定されている場合、サーバーはユーザー認証の確認は行わず、すべてのユーザー接続を許可します。ただし、**authorization** プロパティが無効に設定されている場合でも、**admin** ユーザーは、サーバーに接続するために正しいパスワードを入力する必要があります。

**authorization** プロパティが有効に設定されている場合でも、管理者 (**admin**) は、パスワードの認証を必要としない、匿名ユーザー接続を明示的に許可することができます。これらの接続を許可するには、**anonymous** という名前で、パスワードのないユーザーを作成します。



**anonymous** ユーザーを作成するということは、**anonymous** ユーザーにすべてのアクセス権限が付与されるという意味ではありません。各トピックとキューはまだセキュアであり、これらのデスティネーションを使用（送信または受信）できるかどうかは、これらのデスティネーションに対するアクセス権限のアクセス制御リストによって制御します。**anonymous** ユーザーがアクセスできるのは、セキュアでないデスティネーションのみです。

それでもなお、この機能（匿名ユーザー接続）は、EMS セキュリティ証明書のテスト済み設定には含まれていません。

デスティネーションのセキュリティについての詳細は、デスティネーションプロパティ（67 ページの「[secure](#)」および 91 ページの「[ユーザーの作成](#)」）を参照してください。

## 通信セキュリティ

サーバーとクライアント間、およびサーバー同士で通信を行う場合、EMS で明示的に SSL を設定する必要があります（453 ページの「[SSL プロトコルの使用](#)」を参照）。

SSL 通信では、ソフトウェアでサーバーとクライアントの両方に SSL を実装することが必要です。EMS サーバーには、OpenSSL 実装が含まれています。Java クライアントプログラムは、JSSE（Java 環境の一部）または Entrust 社から別途購入した SSL のいずれかを使用する必要があります。どちらも EMS 製品には含まれていません。C クライアントプログラムは、EMS に含まれている OpenSSL ライブラリを使用できます。

## 認証データのソース

サーバーは、X.509 証明書の認証データ、つまりサーバーパラメータ `ssl_server_trusted`（この値は、EMS 設定ファイルに設定されます）を唯一のソースとして使用します。217 ページの「[ssl\\_server\\_trusted](#)」を参照してください。

サーバーは、セキュアパスワードの認証データとして次の三つのソースを使用します。

- EMS 設定ファイルから取得するローカルデータ。
- LDAP の外部データ。
- ユーザー指定の JAAS LoginModule。

EMS 設定ファイルおよび LDAP サーバーのセキュリティを堅牢にする必要があります。

## タイムスタンプ

管理ツールを使用して、各コマンドによる出力画面で、タイムスタンプの表示 / 非表示を切り替えることができます。セキュアな実装を確実に行うには、タイムスタンプ機能を明示的に有効にする必要があります。次の管理ツールコマンドを使用します。

```
time on
```



## パスワード



パスワードは、企業にとって不正アクセスを防ぐための重要なポイントです。パスワードは、厳格な基準で管理することをお勧めします。

シングル DES (業界の最低限の基準) 相当のセキュリティを確保するためにセキュリティの専門家が推奨するパスワードは、8～14 文字の長さで、ひとつ以上の大文字、ひとつ以上の数字、およびひとつ以上の句読文字を含むパスワードです。

EMS ソフトウェアでは、このようなパスワードの基準は自動的に設定されていません。組織内でこのようなポリシーを設定する必要があります。

## 監査トレースログ

監査情報はログファイル（および `stderr`）に出力され、サーバーの `log_trace` と `console_trace` のパラメータで設定されます（210 ページの「[ログファイルパラメータのトレース](#)」を参照）。

DEFAULT 設定には `+ADMIN` が含まれているため、すべての管理上の操作に対して監査ログが出力されます。詳細は、436 ページの表 72、「[サーバーのトレースオプション](#)」を参照してください。

ログファイル内の監査情報には常にタイムスタンプが記録されます。

管理者は、すべての IT 環境で共通して利用できるツール（テキストエディタなど）を使用して、ログファイルを読んだり、印刷して監査内容を確認することができます。EMS ソフトウェアには、監査内容を確認するための特別なツールは用意されていません。

## EMS による共有ストアファイルへのアクセスの管理方法

---

ふたつの EMS サーバーが同じストアファイルを使用しないよう、あるサーバーの処理中は、どのサーバーもそのストアファイルへのアクセスが制限されます。このセクションでは、ロックされたストアファイルを EMS で管理する方法を説明します。

**Windows** Windows プラットフォームでは、サーバーは Windows の標準の `CreateFile` 関数を使用して、`dwShareMode`（三番目のパラメータ位置）に `FILE_SHARE_READ` を設定し、他のサーバーに対してストアファイルへのアクセスを制限します。

**UNIX** UNIX プラットフォームでは、サーバーは標準の `fcntl` システム呼び出しを使用して、ファイルの協調ロックを実装します。

```
struct flock fl;  
int err;  
  
fl.l_type = F_WRLCK;  
fl.l_whence = 0;  
fl.l_start = 0;  
fl.l_len = 0;  
  
err = fcntl(file, F_SETLK, &fl);
```

ファイルが確実にロックされるようにするには、オペレーティングシステムのマニュアルを参照して、この呼び出しに関する説明を確認することをお勧めします。これは UNIX の種類によって実装方法が異なるためです。

## パフォーマンス調整

デフォルトでは、TIBCO Enterprise Message Service サーバーの一般的なスレッドアーキテクチャは次のとおりです。

- ネットワークトラフィックを処理するスレッドがひとつ。
- 各ストアにひとつのスレッド。
- メッセージの期限切れ処理、ルートの接続など、各種バックグラウンドタスク用の追加スレッド。

### スループットを向上するためのスレッド関係の設定

デフォルトの動作では EMS サーバーのスループットが不十分で、EMS サーバーマシンに複数のコアが存在する場合は、ネットワークトラフィックとストアを処理する EMS スレッドに特定のコアを割り当てます。

たとえば、4 コアのマシンで `processor_ids` パラメータを使用して、コア 0 とコア 1 をネットワークトラフィックの処理に割り当てます。ストア設定の `processor_id` パラメータを使用して、コア 2 を `$sys.failSAFE` ストアの処理に割り当ててもできます。このように設定すると、EMS サーバーにネットワークトラフィックの処理を行うスレッドがふたつ作成され、それぞれコア 0 とコア 1 との割り当て関係が設定されます。また、ストア `$sys.failSAFE` を処理するスレッドにはコア 2 との割り当て関係が設定されます。他のスレッドに対しては、割り当て関係は設定されません。

### コア割り当ての決定

EMS のパフォーマンス調整におけるベストプラクティスは、「少ないほうが良い」ということです。

1. デフォルトの動作で十分なスループットが得られている場合は、変更しないでください。

EMS サーバーでスレッドの割り当て関係を設定しない方が、オペレーティングシステムに EMS サーバーのスレッド調整を任せることで、マシン負荷の変化に対応しやすくなります。また、デフォルトの動作に変更を加える前に、アプリケーションで API が効率的に利用されているかどうかを確認してください。たとえば、永続的メッセージ操作を実行するときは、アプリケーションでの複数スレッド使用（各セッションで使用する）や、ローカルランザクションでの送信と確認応答のバッチ処理を検討します。

2. ネットワークトラフィックの処理には、できるだけ少ない数のコアを使います。コアをひとつだけバインドすると、デフォルト動作よりも十分なパフォーマンスの改善が得られる場合があります。まず、ここから割り当て関係の設

定を試してください。使うコアの数が多すぎると全体のデータ構造でスレッドの競合が増え、スループットを低下させ、マシンのリソースを浪費することになる場合があります。また、コア数が多すぎると不均衡な接続の割り当てが増える可能性もあります。ネットワークトラフィックに割り当てるのに有効なコア数は三つ（一部の負荷では四つ）であることが TIBCO の試験で実証されています。

3. 永続的メッセージングのネットワークトラフィックにコアを割り当てるときは、ストアにもコアを割り当ててください。これは、ストア関連のタスクを処理するスレッドとの競合を防ぐためです。

## ネットワーク I/O 接続

クライアントが EMS サーバーに接続すると、クライアントは EMS サーバーによってネットワーク処理を行うスレッドのいずれかに割り当てられます。このとき、既存の接続数が最も少ないスレッドが選択されて、各スレッドの合計接続数が均等となるように調整されます。

ひとつのスレッドですべての接続が閉じて、EMS サーバーは他のスレッドから既存の接続を移動して再調整は行いません。

また、EMS サーバーは、接続によって発生するトラフィックは考慮しません。たとえば、EMS サーバーで 10 個の接続がひとつのスレッドに割り当てられ、さらに 10 個の接続が他のスレッドに割り当てられている場合でも、最初の 10 個の接続が EMS サーバーに対するネットワークトラフィック全体の 90% を占めている場合には、依然として不均衡な状態です。

## その他の考慮事項

- コアを EMS での使用に割り当てるときは、オペレーティングシステムで該当するコアを他のプロセスに割り当てていないことを確認してください。
- コアは、できる限り同一ダイ上のコアを割り当ててください。これは、ダイ間のキャッシュ共有を少なくするためです。ダイ間のキャッシュ共有が多いと、メモリのパフォーマンスが低下します。
- ハイパースレッドは実コアではありません。可能であれば、ハイパースレッディングは無効にしてください。実際には同一の物理コアを共有しているふたつのハイパースレッディング「コア」に、EMS サーバーとの関係を設定するようなコア割り当ては行わないでください。

## 第 6 章

# EMS 管理ツールの使用

この章では、TIBCO Enterprise Message Service 管理ツールのコマンドと使用方法の概要を説明します。

### トピック

---

- [「EMS 管理ツールの起動」](#)、118 ページ
- [「命名規則」](#)、121 ページ
- [「コマンドのリスト」](#)、122 ページ

## EMS 管理ツールの起動

EMS 管理ツールは、`EMS_HOME/bin` ディレクトリにインストールされているスタンドアロン型の実行可能ファイルです。UNIX の場合は `tibemsadmin`、Windows の場合は `tibemsadmin.exe` というファイル名です。

Windows が起動しているコンピュータで、スタートメニューから [ プログラム ] > [ TIBCO ] > [ TIBCO EMS 7.0 ] > [ Start EMS administration tool ] を選択して、管理ツールを起動することもできます。

EMS 管理ツールを使用するには、101 ページの第 5 章、「EMS サーバーの実行」の説明に従って、EMS サーバーを起動する必要があります。



システムで共有設定ファイルを使用する場合は、アクティブサーバーに接続しているときにのみ、管理ツールを使って実行するアクションが有効になります。フォールトトレランスサーバーを設定している場合にスタンバイサーバーに接続すると、管理ツールに通知メッセージが表示されます。また、管理ツールがバックアップサーバーに接続している場合は、スイッチオーバーが発生するとサーバーから切断されます。

`tibemsadmin` 起動パラメータについての情報を表示するには、`tibemsadmin -help` と入力します。`tibemsadmin` パラメータの指定はすべてオプションです。

表 15 に `tibemsadmin` のオプションをリストします。

表 15 `tibemsadmin` のオプション

オプション	説明
<code>-help</code> または <code>-h</code>	ヘルプ画面を表示します。
<code>-script script-file</code>	<code>tibemsadmin</code> コマンドが記述されているテキストファイルを指定して実行して、終了します。この章で説明されている有効な <code>tibemsadmin</code> コマンドのいずれも実行できます。  ファイル内の各コマンドは行区切りで区切られます。つまり、ひとつのコマンドは 1 行で記述し (コマンドの途中で改行しない)、各コマンドは改行で区切ります。
<code>-server server-url</code>	指定したサーバーへ接続します。
<code>-user user-name</code>	サーバーへのログインに使用するユーザー名。
<code>-password password</code>	サーバーへのログインに使用するパスワード。

表 15 tibemsadmin のオプション

オプション	説明
<code>-ignore</code>	スクリプトファイルの実行時に発生するエラーを無視します。このパラメータで無視されるのは、コマンド実行のエラーのみです。スクリプトの構文エラーは無視されません。
<code>-mangle [password]</code>	<p>パスワードを解読不能にして、終了します。出力で解読不能にした文字列は、設定ファイルの次のいずれかのパスワードの値として設定できます。</p> <ul style="list-style-type: none"> <li>• サーバーのパスワード</li> <li>• サーバーの SSL パスワード</li> <li>• LDAP 管理者のパスワード</li> <li>• データベースのパスワード</li> </ul> <p>パスワードを入力しないと、入力が求められます。</p>
<code>-ssl_trusted filename</code>	信頼する証明書を含むファイル。このパラメータは、必要に応じて複数回入力することができます。
<code>-ssl_identity filename</code>	クライアント証明書と（オプションで）発行元の証明書、および秘密鍵を含むファイル。
<code>-ssl_issuer filename</code>	クライアント側の身元確認を行うための発行元の証明書を含むファイル。
<code>-ssl_password password</code>	秘密鍵または PKCS#12 パスワード。パスワードが必要だが指定されていない場合は、入力するように要求されます。
<code>-ssl_noverifyhostname</code>	ホスト名を証明書の名前と照合しません。
<code>-ssl_hostname name</code>	ホストから送信された証明書に必要な名前。
<code>-ssl_trace</code>	ロードされている証明書とホストから送信された証明書を表示します。
<code>-ssl_debug_trace</code>	デバッグに有用なトレースを追加で表示します。



コマンドで `-user` と `-password` を指定すると、その情報を後で使用するために保存することはできません。その情報は同じコマンド行で指定したサーバーへの接続のみに使用されます。ひとつのコマンド行に入力したユーザー名とパスワードは、スクリプトファイルにまたは対話的に入力された後続の接続コマンドで再使用されません。

**例：**

```
tibemsadmin -server "tcp://host:7222"  
tibemsadmin -server "tcp://host:7222" -user admin -password secret
```

SSL 接続の設定を選択すると、いくつかのオプションが必要になります。SSL の接続についての詳細は、453 ページの [第 18 章、「SSL プロトコルの使用」](#) を参照してください。

## tibemsadmin の最初の起動

管理ツールのデフォルトユーザーは、`admin` という名前です。このデフォルトのユーザー名を使用して、管理ツールにログインします。

サーバーの設定へのアクセスを保護するには、ユーザー `admin` にパスワードを割り当てる必要があります。

### 管理者へのパスワードの割り当て

1. 前述の説明に従って、管理ツールにログインして接続します。
2. パスワードを変更する場合は、`set password` コマンドを使用します。

```
set password admin password
```

管理ツールを再起動し、`connect` と入力すると、サーバーに接続する前に管理ツールはパスワードの入力を求めます。

パスワードの設定と再設定についての詳細は、136 ページの [「set password」](#) を参照してください。



## 命名規則

---

ユーザー、グループ、トピック、キューの名前を作成する際に、以下の規則が適用されます。

- \$ の文字は、キュー名とトピック名の先頭には使用できませんが、ユーザー名とグループ名の先頭には使用できます。
- ユーザー名にはコロン (:) を使用できません。
- 空白文字は説明フィールドで使用できます。ただし、説明フィールド全体が二重引用符で囲まれている場合に限りです (例: "description field")。
- \* と > はワイルドカード文字であり、ワイルドカードとして以外では名前に使用できません。

ワイルドカードについての詳細は、73 ページの「[ワイルドカード](#)」を参照してください。

- ドット (.) は、デスティネーション名の要素の区切りに使用します (例: **foo.bar.\***)。この用途以外には使用できません。

### 名前の長さの制限

以下のパラメータ名には、それぞれ長さの制限が適用されます。

- デスティネーション名: 249 文字以内にする必要があります。トピックとキューの命名規則についての詳細は、52 ページの「[デスティネーション名の構文](#)」を参照してください。
- ユーザー名: 127 文字以内にする必要があります。username パラメータについては、252 ページの「[users.conf](#)」を参照してください。
- グループ名: 127 文字以内にする必要があります。group-name パラメータについては、239 ページの「[groups.conf](#)」を参照してください。
- クライアント ID: 255 文字以内にする必要があります。clientID パラメータについては、235 ページの「[factories.conf](#)」を参照してください。
- 接続 URL: 1000 文字以内にする必要があります。url パラメータについては、235 ページの「[factories.conf](#)」を参照してください。
- パスワード: 4096 文字以内にする必要があります。この長さの制限は、接続するクライアントまたはサーバーの認証に **tibemsd** が使うパスワードに適用されます。

## コマンドのリスト

管理ツールのコマンドラインインタフェースにより、さまざまな機能を実行できます。システムで共有設定ファイルを使用する場合は、アクティブサーバーに接続しているときにのみ、管理ツールを使って実行するアクションが有効になります。



以下に挙げたコマンドの多くは、ユーザー、グループ、トピック、キューの名前を指定する引数を受け入れます。これらの名前に適用される構文の詳細は、121 ページの「[命名規則](#)」を参照してください。



この表には SSL コマンドは含まれていません。SSL コマンドについては、453 ページの第 18 章、「[SSL プロトコルの使用](#)」の各表を参照してください。

以下に、各コマンドについてアルファベット順に、コマンド構文を含めて説明します。

### add member

```
add member group_name user_name [,user2,user3,...]
```

グループに 1 人以上のユーザーを追加します。定義されていないユーザー名は、外部ユーザーとしてグループに追加されます。詳細は、270 ページの「[管理コマンドおよび外部ユーザーと外部グループ](#)」を参照してください。

### addprop factory

```
addprop factory factory-name properties ...
```

ファクトリにプロパティを追加します。各プロパティ名はスペースで区切ります。ファクトリのプロパティのリストについては、235 ページの「[factories.conf](#)」を参照してください。

構文例：

```
addprop factory MyTopicFactory ssl_trusted=cert1.pem
ssl_trusted=cert2.pem ssl_verify_host=disabled
```

### addprop queue

```
addprop queue queue-name properties,...
```

キューにプロパティを追加します。各プロパティ名はコンマで区切ります。

キューに割り当て可能なプロパティについての詳細は、55 ページの「[デステーションのプロパティ](#)」を参照してください。

## addprop route

```
addprop route route-name prop=value[ prop-value...]
```

ルートにプロパティを追加します。

デスティネーション（トピックとキュー）の各プロパティは、コンマで区切る必要がありますが、ルートとファクトリの各プロパティはスペースで区切ります。

ルートを作成する際に **zone\_name** と **zone\_type** のパラメータを使用できますが、これらのパラメータは後で変更することができません。

ルートのプロパティについては、506 ページの「[ルートとゾーンの設定](#)」を参照してください。

設定ファイル **routes.conf** については、241 ページの「**routes.conf**」を参照してください。

## addprop topic

```
addprop topic topic_name properties,...
```

トピックにプロパティを追加します。各プロパティ名はコンマで区切ります。

トピックに割り当て可能なプロパティについての詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

## autocommit

```
autocommit [on|off]
```

**autocommit** を **on** に設定すると、設定ファイルに加えられた変更は、各コマンドの実行後にディスクに自動保存されます。**autocommit** を **off** に設定するときは、設定の変更をディスクに保存するために手動で **commit** コマンドを実行する必要があります。

デフォルトでは、対話的にコマンドを実行できるよう、**autocommit** は **on** に設定されています。

パラメータを指定せずに **autocommit** を入力すると、**autocommit** の現在の設定 (**on** または **off**) が表示されます。



**autocommit** の設定にかかわらず、各 **admin** コマンドを実行すると、EMS サーバーは直ちにその変更を設定に反映します。**autocommit** 機能は、設定がファイルに書き込まれるタイミングのみを指定します。

## commit

```
commit
```

変更したすべての設定をディスク上のファイルに反映します。

## compact

```
compact store_name max_time
```

指定したストアのストアファイルを圧縮します。タイプが `mstore` のストアは圧縮できません。

圧縮は長時間を要する場合があります、他のデータベース処理を中断させるため、`max_time` を使って制限時間（秒単位）を設定できます。`max_time` にはゼロより大きい数を指定する必要があります。

ストアファイルで切り捨てを有効にしていない場合は、`compact` コマンドを使ってもファイルサイズは小さくなりません。切り捨てを有効にするには、`stores.conf` の `file_truncate` パラメータを使用します。詳細は、[243 ページの「stores.conf」](#) を参照してください。

ストアファイルの圧縮は、Used Space の使用率が 30% 以下の場合にのみ実行することをお勧めします（[162 ページの「show store」](#) を参照）。

## connect

```
connect [server-url {admin|user_name} password]
```

管理ツールをサーバーに接続します。管理者であれば、誰でも接続可能です。管理者とは、`admin` ユーザー、`$admin` グループに属するユーザー、または管理者のアクセス権限が有効にされているユーザーのいずれかです。管理者のアクセス権限についての詳細は、[257 ページの「管理者のアクセス権限」](#) を参照してください。

`server-url` は通常、次の形式で指定します。

`protocol://host-name:port-number`

例：

```
tcp://myhost:7222
```

プロトコルには、`tcp` または `ssl` を指定できます。

ユーザー名またはパスワードを指定していないユーザーには、ユーザー名とパスワードの入力が求められます。コマンドでユーザー名を指定済みの場合は、パスワードのみの入力が求められます。

他のオプションを指定せずに、`connect` を入力することができます。この場合、管理ツールはローカルサーバーの 7222 のデフォルトポートへの接続を試みます。

## create bridge

```
create bridge source=type:dest_name target=type:dest_name [selector=selector]
```

デスティネーション間にブリッジを作成します。

*type* は *topic* または *queue* です。

詳細は、[230 ページ](#)の「[bridges.conf](#)」を参照してください。

## create durable

```
create durable topic-name durable-name [property, ... ,property]
```

持続サブスクライバを静的に作成します。

パラメータとプロパティの説明および競合状況についての詳細は、[234 ページ](#)の「[durables.conf](#)」を参照してください。

## create factory

```
create factory factory_name factory_parameters
```

接続ファクトリを新規作成します。

ファクトリのパラメータについての説明は、[235 ページ](#)の「[factories.conf](#)」を参照してください。

## create group

```
create group group_name "description"
```

ユーザーグループを新規作成します。

作成した状態では、グループ内は空です。[add member](#) コマンドを使用して、グループにユーザーを追加します。

## create jndiname

```
create jndiname new_jndiname topic|queue|jndiname name
```

トピックまたはキューの JNDI 名を作成します。JNDI 名が作成済みのトピックには、別の JNDI 名が作成されます。

例：

```
create FOO jndiname BAR
```

この例では、JNDI 名 **BAR** によって参照されるオブジェクトと同じオブジェクトを参照する **FOO** という JNDI 名が新規に作成されます。

## create queue

```
create queue queue_name [properties]
```

指定した名前とプロパティでキューを新規作成します。設定可能なキューのプロパティについては、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。複数のプロパティを列挙する場合は、コンマで区切ります（240 ページの「[queues.conf](#)」を参照）。

## create route

```
create route name url=URL [properties ...]
```

ルートを作成します。

*name* には、ルートの接続先となる他のサーバーの名前を指定します。

ローカルサーバーは、指定した URL のデスティネーションサーバーに接続します。複数のサーバーをフォールトトレランスの設定にしている場合は、各サーバーの URL をコンマ区切りで列挙して指定できます。

ルートのプロパティについては、241 ページの「[routes.conf](#)」を参照してください。複数のプロパティを指定する場合は、パラメータ名と値のペアをスペース区切りで列挙します。

ルートを作成する際に *zone\_name* と *zone\_type* のパラメータを使用できますが、これらのパラメータは後で変更することができません。

指定した名前のパッシブルルートがすでに存在する場合、そのルートはこのコマンドによって「アクティブルートからアクティブルートへの」ルートに変更されます。505 ページの「[アクティブルートとパッシブルルート](#)」を参照してください。

ルートのパラメータについての詳細は、506 ページの「[ルートとゾーンの設定](#)」を参照してください。

## create rvcmlistener

```
create rvcmlistener transport_name cm_name subject
```

RVCМ リスナーをサーバーに登録します。これにより、*tibrvcм* トランスポートにエクスポートされたメッセージ（最初に送信されたメッセージも含む）が指定したリスナーに対して保証されます。サーバーは、TIBCO Rendezvous の呼び出し *tibrvcмTransport\_AddListener* を実行します。

以下のパラメータがあります。

- *transport\_name* : この RVCМ リスナーが適用されるトランスポートの名前。
- *cm\_name* : トピックメッセージのエクスポート先となる RVCМ リスナーの名前。
- *subject* : メッセージのパブリッシュ先となる RVCМ サブジェクトの名前。  
*export* プロパティを指定するトピック名と同じ名前です。

詳細は、[247 ページの「`tibrvcn.conf`」](#)と [394 ページの「Rendezvous 保証メッセージ \(RVCM\) のパラメータ」](#)を参照してください。

## create topic

```
create topic topic_name [properties]
```

指定した名前とプロパティでトピックを新規作成します。設定可能なトピックのプロパティについては、[55 ページの「デスティネーションのプロパティ」](#)を参照してください。複数のプロパティを列挙する場合は、コンマで区切ります ([247 ページの「`topics.conf`」](#)を参照)。

## create user

```
create user user_name ["user_description"] [password=password]
```

新規ユーザーを作成します。ユーザー名の後に、ユーザーの説明を引用符で囲んで追加することもできます。パスワードの割り当ては任意です。`set password` コマンドを使用して後で割り当てることもできます。



ユーザー名にはコロン (:) を使用できません。

## delete all

```
delete all users|groups|topics|queues|durables  
[topic-name-pattern|queue-name-pattern]
```

オプションのパラメータを設定せずに `delete all users|groups|topics|queues|durables` と指定すると、指定したユーザー、グループ、トピック、またはキューがまとめて削除されます。

トピックまたはキュー、そしてオプションのパラメータを使用する場合は次のように指定します。

```
delete all topics|queues topic-name-pattern|queue-name-pattern
```

指定したトピック名またはキュー名のパターンに一致するトピックとキューがすべて削除されます。

## delete bridge

```
delete bridge source=type:dest_name target=type:dest_name
```

指定したソースデスティネーションとターゲットデスティネーション間のブリッジを削除します。

`type` は `topic` または `queue` です。

ブリッジについての詳細は、[78 ページの「デスティネーションブリッジ」](#)を参照してください。

## delete connection

```
delete connection connection-id
```

指定したクライアントの接続を削除します。接続 ID は、`show connection` で表示される接続の説明の先頭の列に表示されます。

## delete durable

```
delete durable durable-name clientID
```

指定した持続サブスクライバを削除します。

持続名とクライアント ID の両方を指定すると、EMS サーバーは `clientID:durable-name` という名前の持続を持続リストで検索します。一致する持続サブスクライバが見つからない場合、完全修飾持続名を含むエラーメッセージが管理ツールから出力されます。

234 ページの「[指定の競合](#)」も参照してください。

## delete factory

```
delete factory factory-name
```

指定した接続ファクトリを削除します。

## delete group

```
delete group group-name
```

指定したグループを削除します。

## delete jndiname

```
delete jndiname jndiname
```

指定した JNDI 名を削除します。最後に残った接続ファクトリオブジェクトの JNDI 名を削除すると、接続ファクトリオブジェクト自体が削除されてしまうので注意してください。

詳細は、347 ページの[第 12 章、「JNDI を実装した EMS の使用」](#)を参照してください。

## delete message

```
delete message messageID
```

指定したメッセージ ID のメッセージを削除します。



## delete queue

```
delete queue queue-name
```

指定したキューを削除します。

## delete route

```
delete route route-name
```

指定したルートを削除します。

## delete rvcmlistener

```
delete rvcmlistener transport_name cm_name subject
```

RVCМ リスナーのサーバー登録を解除します。これにより、指定リスナー用に RVCМ 元帳内に保持されていたメッセージが解放されます。サーバーは、TIBCO Rendezvous の呼び出し `tibrvcmlTransport_RemoveListener` を実行します。

以下のパラメータがあります。

- `transport_name` : この RVCМ リスナーが適用されるトランスポートの名前。
- `cm_name` : トピックメッセージのエクスポート先 RVCМ リスナーの名前。
- `subject` : メッセージのパブリッシュ先となる RVCМ サブジェクトの名前。  
`export` プロパティを指定するトピック名と同じ名前です。

詳細は、[247 ページの「tibrvcml.conf」](#)と [394 ページの「Rendezvous 保証メッセージ \(RVCМ\) のパラメータ」](#)を参照してください。

## delete topic

```
delete topic topic-name
```

指定したトピックを削除します。

## delete user

```
delete user user-name
```

指定したユーザーを削除します。

## disconnect

```
disconnect
```

サーバーから管理ツールの接続を解除します。

## echo

```
echo [on|off]
```

**echo** は、標準出力に出力されるレポートの内容を制御します。**echo** を **off** に設定すると、管理ツールはエラーとクエリのみを出力します。**echo** を **on** に設定すると、管理ツールのレポートに成功したコマンド実行の記録が含まれます。

このコマンドのパラメータの **on/off** を選択することで、**echo** を制御します。パラメータを指定せずに **echo** を設定すると、**echo** の現在の設定 (**on** または **off**) が表示されます。このコマンドは主に、スクリプトに使用されます。

**echo** のデフォルト設定は **on** です。

## exit

```
exit (aliases: quit, q, bye, end)
```

管理ツールを終了します。

管理者が **exit** を設定したときに、設定に加えた変更がディスクに保存されていないことがあります。この場合、管理ツールを終了する前に、**commit** コマンドの使用が求められます。

## grant queue

```
grant queue queue-name user=name | group=name permissions
```

指定したユーザーまたはグループに、指定したキューへの指定したアクセス権限を許可します。キュー名の後の名前は、グループ名が最初にチェックされてから、ユーザー名がチェックされます。

指定したアクセス権限が、既存の権限に追加されます。各アクセス権限はコマンドで区切ります。付与可能なユーザー権限をすべて許可する場合は、*permissions* に **all** を指定します。

以下のユーザー権限があります。

- **receive**
- **send**
- **browse**

キューのアクセス権限についての詳細は、273 ページの「[ユーザーのアクセス権限](#)」の表 47 を参照してください。

管理者のデスティネーションレベルのアクセス権限も、このコマンドで許可できます。管理者のキューのアクセス権限には、以下のものがあります。

- `view`
- `create`
- `delete`
- `modify`
- `purge`

デスティネーションのアクセス権限についての詳細は、262 ページの「[デスティネーションレベルのアクセス権限](#)」を参照してください。

## grant topic

`grant topic topic-name user=name | group=name permissions`

指定したユーザーまたはグループに、指定したトピックへの指定したアクセス権限を許可します。トピック名の後の名前は、グループ名が最初にチェックされてから、ユーザー名がチェックされます。

指定したアクセス権限が、既存の権限に追加されます。各アクセス権限はコンマで区切ります。付与可能な権限をすべて許可する場合は、*permissions* に **all** を指定します。

トピックのアクセス権限：

- `subscribe`
- `publish`
- `durable`
- `use_durable`

トピックのアクセス権限についての詳細は、273 ページの「[ユーザーのアクセス権限](#)」の表 48 を参照してください。

管理者のデスティネーションレベルのアクセス権限も、このコマンドで許可できます。管理者のトピックのアクセス権限には、以下のものがあります。

- `view`
- `create`
- `delete`

- `modify`
- `purge`

デスティネーションのアクセス権限についての詳細は、262 ページの「[デスティネーションレベルのアクセス権限](#)」を参照してください。

## grant admin

```
grant admin user=name | group=name admin_permissions
```

指定したグローバルな管理者アクセス権限を指定したユーザーまたはグループに許可します。グローバル管理者のアクセス権限の全リストについては、259 ページの「[グローバルな管理者アクセス権限](#)」を参照してください。

## help

```
help (aliases: h, ?)
```

ヘルプの情報を表示します。

利用可能な全コマンドの概要を表示するには、`help commands` と入力します。

特定のコマンドのヘルプを表示するには、`help command` を入力します。

## info

```
info (alias: i)
```

接続されているサーバーの名前とそのサーバーに関する情報を表示します。

## jaci clear

```
jaci clear
```

JACI 権限キャッシュのすべてのエントリを空にします。

## jaci resetstats

```
jaci resetstats
```

JACI キャッシュのすべての統計情報カウンタをゼロにリセットします。

## jaci showstats

```
jaci showstats
```

JACI キャッシュのパフォーマンスの統計情報を出力します。

## purge all queues

```
purge all queues [pattern]
```

すべてのキューまたは選択したキューをパージします。

パターンのパラメータを指定しないでこのコマンドを使用すると、すべてのレシーバーに宛てたすべてのキュー内の全メッセージが消去されます。

*pattern* パラメータを指定してこのコマンドを使用すると、すべてのキューからパターン（例：foo.\*）に一致するすべてのメッセージが消去されます。

## purge all topics

```
purge all topics [pattern]
```

すべてのトピックまたは選択したトピックをパージします。

パターンのパラメータを指定しないでこのコマンドを使用すると、すべてのサブスクライバに宛てたすべてのトピック内の全メッセージが消去されます。

*pattern* パラメータを指定してこのコマンドを使用すると、すべてのトピックからパターン（例：foo.\*）に一致するすべてのメッセージが消去されます。

## purge durable

```
purge durable durable-name
```

指定した持続サブスクライバ宛てのトピックのすべてのメッセージをパージします。

## purge queue

```
purge queue queue-name
```

指定したキュー内のすべてのメッセージをパージします。

## purge topic

```
purge topic topic-name
```

指定したトピックのすべてのサブスクライバ宛てのすべてのメッセージをパージします。

## remove member

```
remove member group-name user-name[,user2,user3,...]
```

指定したグループから指定した 1 人以上のユーザーを削除します。

## removeprop factory

```
removeprop factory factory-name properties
```

指定したファクトリから指定したプロパティを削除します。プロパティのリストは、235 ページの「[接続ファクトリのパラメータ](#)」を参照してください。

## removeprop queue

```
removeprop queue queue-name properties
```

指定したキューから指定したプロパティを削除します。

## removeprop route

```
removeprop route route-name properties
```

指定したルートから指定したプロパティを削除します。

URL は削除できません。

ルートを作成する際に `zone_name` と `zone_type` のパラメータを使用できますが、これらのパラメータは後で変更することができません。

ルートのパラメータについては、506 ページの「[ルートとゾーンの設定](#)」を参照してください。

設定ファイル `routes.conf` については、241 ページの「[routes.conf](#)」を参照してください。

## removeprop topic

```
removeprop topic topic-name properties
```

指定したトピックから指定したプロパティを削除します。

## resume route

```
resume route route-name
```

[suspend route](#) コマンドでメッセージ送信を一時停止している場合、指定ルートへのメッセージ送信を再開します。

## revoke admin

```
revoke admin user=name | group=name permissions
```

指定したユーザーまたはグループから指定したグローバルな管理者アクセス権限を取り消します。管理者のアクセス権限についての詳細は、255 ページの[第 8 章](#)、「[認証とアクセス権限](#)」を参照してください。

## revoke queue

```
revoke queue queue-name user=name | group=name permissions
revoke queue queue-name * [user | admin | both]
```

指定したキューのユーザーまたはグループから指定したアクセス権限を取り消します。

ユーザーまたはグループに許可されているキューへのアクセス権限は、**receive**、**send**、**browse**、および **all** です。キューの管理者のアクセス権限は、**view**、**create**、**delete**、**modify**、および **purge** です。

アスタリスク (\*) を指定して、このキューのユーザーレベルのアクセス権限をまとめて削除することができます。オプションで **admin** パラメータを使用して、すべての管理者のアクセス権限を取り消すか、**both** パラメータを使用して、キューのユーザーレベルの権限と管理者のアクセス権限の両方を取り消すことができます。

詳細は、255 ページの [第 8 章、「認証とアクセス権限」](#) を参照してください。

## revoke topic

```
revoke topic topic-name user=name | group=name permissions
revoke topic topic-name * [user | admin | both]
```

指定したトピックのユーザーまたはグループから指定したアクセス権限を取り消します。

ユーザーおよびグループのトピックのアクセス権限は、**subscribe**、**publish**、**durable**、**use\_durable**、および **all** です。トピックの管理者のアクセス権限は、**view**、**create**、**delete**、**modify**、および **purge** です。

アスタリスク (\*) を指定して、このトピックのユーザーレベルのアクセス権限をまとめて削除することができます。オプションで **admin** パラメータを使用して、すべての管理者のアクセス権限を取り消すか、**both** パラメータを使用して、トピックのユーザーレベルの権限と管理者のアクセス権限の両方を取り消すことができます。

詳細は、255 ページの [第 8 章、「認証とアクセス権限」](#) を参照してください。

## rotatelog

### rotatelog

現在のログファイルを強制的にバックアップし、切り詰めます。サーバーは、新しく空のログファイルへの記録を開始します。

バックアップファイルの名前は、現在のログファイルの名前に連番を付けた形式で作成されます。サーバーは、現在のログファイルのディレクトリを照会し、連番の最も大きな番号を特定すると、その次の番号を新しいバックアップファイルの名前に割り当てます。たとえば、ログファイル名が `tibems.log` で、すでに `tibems.log.1` と `tibems.log.2` のバックアップファイルが作成されている場合、サーバーは、次のバックアップファイル名を `tibems.log.3` とします。

## set password

```
set password user-name [password]
```

指定したユーザーのパスワードを設定します。

コマンドにパスワードを指定していない場合、サーバーからパスワードの入力を求められます。

- パスワードをリセットするには、次のように入力します。

```
set password user-name
```

コマンドプロンプトで、新しいパスワードを入力します。

- パスワードを削除するには、パスワードを指定しないでこのコマンドを使用し、プロンプトに対して **Enter** キーを押します。



パスワードは、企業にとって不正アクセスを防ぐための重要なポイントです。パスワードは、厳格な基準で管理することをお勧めします。

シングル DES（業界の最低限の基準）相当のセキュリティを確保するためにセキュリティの専門家が推奨するパスワードは、8～14 文字の長さで、ひとつ以上の大文字、ひとつ以上の数字、およびひとつ以上の句読文字を含むパスワードです。



# set server

`set server parameter=value [parameter=value ...]`

`set server` コマンドで、多くのパラメータを制御できます。複数のパラメータはスペースで区切って指定します。表 16 に、このコマンドで設定できるパラメータを示します。

表 16 Set server のパラメータ （1 / 6 シート）

パラメータ	説明
<code>password [= string]</code>	<p>ルーティングされた他のサーバーへ接続するために、サーバーが使用するサーバーのパスワードを設定します。値を入力しないと、管理ツールによって入力が求められます。入力した値は、解読不能な形式で（ただし、暗号化されずに）メインサーバーの設定ファイルに、保存されます。</p> <p>このパスワードをリセットするには、コマンドプロンプトで空の文字列を 2 回入力します。</p>
<code>authorization=enabled disabled</code>	<p><code>tibemsd.conf</code> ファイルに <code>authorization</code> モードを設定します。</p> <p>無効から有効にすると、サーバーは以降のすべての要求の ACL 権限をチェックします。サーバーは、既存のプロデューサとコンシューマの有効な認証を要求しても、後からそれらを再認証することはありません。以前に有効な認証がない場合、ユーザーへのアクセスは拒否されます。</p>

表 16 Set server のパラメータ ( 2 / 6 シート)

パラメータ	説明
log_trace=trace-items	<p>logfile パラメータによって定義されたファイルのトレースの基本設定を設定します。logfile が設定されていない場合、指定した値は保存されますが、反映されません。</p> <p>このパラメータの値には、トレースオプションをコンマで区切って指定します。トレースオプションのリストとその意味については、436 ページの表 72、「サーバーのトレースオプション」を参照してください。</p> <p>トレースオプションは、次の三つの形式で指定できます。</p> <ul style="list-style-type: none"><li>• 符号なし：プリフィクス文字が指定されていないトレースオプション。既存のすべてのトレースオプションを置換する。</li><li>• ++ で始まるトレースオプション。指定したオプションを現在のトレースオプションセットに追加する。</li><li>• -- で始まるトレースオプション。指定したオプションを、現在のトレースオプションセットから削除する。</li></ul> <p><b>例：</b></p> <p>次のコード例では、アクセス制御違反に関するメッセージのみを表示するようにトレースログを設定します。</p> <p>log_trace=ACL</p> <p>次のコード例では、SSL メッセージに加えて、すべてのデフォルトのトレースメッセージを表示するように設定します。ただし、ADMIN メッセージを除きます。</p> <p>log_trace=DEFAULT, -ADMIN, +SSL</p>

表 16 Set server のパラメータ (3 / 6 シート)

パラメータ	説明
<p><code>console_trace=console-trace-items</code></p>	<p><code>stderr</code> に出力するトレースオプションを設定します。値の意味は、<code>log_trace</code> の場合と同じです。ただし、コンソールのトレースは、ログファイルとは独立して行われます。</p> <p><code>logfile</code> が定義されている場合は、次のように指定することによって、コンソール出力を停止できます。</p> <p><code>console_trace=-DEFAULT</code></p> <p>重要なエラーメッセージ（およびその他一部メッセージ）は、このトレース設定より優先され、常に出力されます。</p> <p><b>例：</b></p> <p>次のコード例では、TIBCO Rendezvous アドバイザリメッセージを受け取ったときに、トレースメッセージをコンソールに送信します。</p> <p><code>console_trace=RVADV</code></p>
<p><code>client_trace={enabled disabled}</code>  <code>[target=location] [filter=value]</code></p>	<p>管理者は、接続または接続のグループをトレースすることができます。このプロパティが <code>enabled</code> の場合、クライアントは接続のオープンとクローズ、メッセージのアクティビティ、およびトランザクションのアクティビティのトレース出力を生成します。このタイプのトレースでは、クライアントプログラムを再起動する必要はありません。</p> <p>クライアントはトレース出力を、<code>stderr</code>（デフォルト）または <code>stdout</code> のいずれかの <code>location</code> に送信します。</p> <p>フィルタを指定し、特定の接続を選択してトレースすることもできます。<code>filter</code> には、<code>user</code>、<code>connid</code>、または <code>clientid</code> のいずれかを指定できます。<code>value</code> には、フィルタの指定に応じて、ユーザー名またはユーザー ID を指定できます。</p> <p><code>filter</code> および <code>value</code> 句を指定しない場合、デフォルトですべての接続がトレースされます。</p> <p>管理ツールを使用してこのパラメータを設定した場合、設定ファイル <code>tibemsd.conf</code> 内のこの値は変更されません。</p>

表 16 Set server のパラメータ ( 4 / 6 シート)

パラメータ	説明
max_msg_memory= <i>value</i>	<p>サーバーがメッセージに使用できるメモリの最大量です。</p> <p>詳しい説明については、<a href="#">177 ページの「tibemspd.conf」</a> の max_msg_memory を参照してください。</p> <p>単位としては、KB、MB、または GB を指定できます 最小値は 8MB です。ゼロは特別な値で、制限がないことを示します。</p> <p>この値を低くすると、メッセージによって占められたメモリがすぐには解放されません。</p>
msg_swapping=enabled disabled	メッセージをディスクにスワップする機能を有効または無効にします。
track_message_ids=enabled disabled	MessageID によるメッセージのトレースを有効または無効にします。
track_correlation_ids=enabled disabled	CorrelationID によるメッセージのトレースを有効または無効にします。
ssl_password[= <i>string</i> ]	<p>SSL 専用のパスワードを設定します。</p> <p>サーバー ID ファイルの内容を復号するために、サーバーが使用する秘密鍵または PKCS#12 ファイルのパスワードを設定します。パスワードは解読不能な形式で保存されます。</p>
ft_ssl_password[= <i>string</i> ]	<p>フォールトトレランスでの SSL 使用のパスワードを設定します。</p> <p>FT ID ファイルの内容を復号するために、サーバーが使用する秘密鍵または PKCS#12 ファイルのパスワードを設定します。パスワードは解読不能な形式で保存されます。</p>

表 16 Set server のパラメータ (5 / 6 シート)

パラメータ	説明
<code>server_rate_interval=num</code>	<p>サーバーの全般統計情報の平均値を求める間隔 (秒単位) を設定します。このパラメータには、ゼロより大きい任意の正の整数を設定できます。</p> <p>サーバーの全般統計情報は常に収集されるため、このパラメータにゼロを設定することはできません。デフォルトではこのパラメータは 1 に設定されています。</p> <p>このパラメータを設定すると、指定した間隔で、メッセージの送信率とメッセージサイズの平均値が得られます。</p>
<code>statistics=enabled disabled</code>	<p>プロデューサ、コンシューマ、デスティネーション、およびルートを集める統計情報を有効または無効にします。デフォルトではこのパラメータは <b>disabled</b> に設定されています。</p> <p>統計情報の収集を無効にすると、各オブジェクトの統計情報で合計値がゼロにリセットされます。</p>
<code>rate_interval=num</code>	<p>ルート、デスティネーション、プロデューサ、コンシューマの統計情報の平均値を求める間隔 (秒単位) を設定します。デフォルトではこのパラメータは 3 秒に設定されています。このパラメータにゼロを設定すると、平均計算が無効になります。</p>
<code>detailed_statistics=NONE   PRODUCERS, CONSUMERS, ROUTES, CHANNELS</code>	<p>オブジェクトに対して詳細統計情報の追跡を行うかどうかを指定します。詳細統計情報の追跡は、ルート、チャネル、デスティネーションを指定しないプロデューサ、ワイルドカードのデスティネーションを指定するコンシューマにのみ適しています。詳細統計情報の追跡が有効になっている場合、各デスティネーションの統計情報がそのオブジェクトのために記録されます。</p> <p>このパラメータに <b>NONE</b> を設定すると、詳細統計情報の追跡が無効になります。各オブジェクトのトラックを有効にする場合、<b>PRODUCERS</b>、<b>CONSUMERS</b>、<b>ROUTES</b>、または <b>CHANNELS</b> を任意の組み合わせで指定できます。複数の種類の詳細追跡を指定する場合、各アイテムをコンマで区切ります。</p>

表 16 Set server のパラメータ ( 6 / 6 シート)

パラメータ	説明
statistics_cleanup_interval=num	デスティネーションにアクティビティがない場合、サーバーが詳細統計情報を保管する期間 (秒単位) を指定します。このパラメータは、詳細統計情報の追跡に使用されるメモリの最大容量を制御する場合に役立ちます。指定された間隔に達すると、アクティビティのないデスティネーションの統計情報は削除されます。
max_stat_memory=num	<p>詳細統計情報の収集に使用するメモリの最大容量を指定します。単位を指定しない場合、ファイルサイズの単位にはバイトが使用されます。単位を指定する場合は、KB、MB、または GB を使用できます。</p> <p>最大メモリ制限に達すると、サーバーは詳細統計情報の収集を中止します。統計情報が削除され、メモリが利用可能になると、サーバーは詳細統計情報の収集を再開します。</p>

setprop factory

```
setprop factory factory-name properties ...
```

接続ファクトリのプロパティを設定し、既存のプロパティを無効にします。複数のプロパティを指定する場合は、スペースで区切って指定します。設定可能な接続ファクトリのプロパティのリストは、235 ページの「[接続ファクトリのパラメータ](#)」を参照してください。

setprop queue

```
setprop queue queue-name properties, ...
```

キューのプロパティを設定し、既存のプロパティを無効にします。このコマンドで明示的に設定されていないキューのプロパティは削除されます。

複数のプロパティを指定する場合は、コンマで区切って指定します。設定可能なキューのプロパティのリストは、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

## setprop route

```
setprop route route-name properties ...
```

ルートのプロパティを設定し、既存のプロパティを無効にします。このコマンドで明示的に設定されていないルートのプロパティは削除されます。

ルートを作成する際に `zone_name` と `zone_type` のパラメータを使用できますが、これらのパラメータは後で変更することができません。

複数のプロパティを指定する場合は、スペースで区切って指定します。ルートのパラメータについては、[241 ページの「`routes.conf`」](#) および [506 ページの「ルートとゾーンの設定」](#) を参照してください。

## setprop topic

```
setprop topic topic-name properties
```

トピックのプロパティを設定し、既存のプロパティを無効にします。このコマンドで明示的に設定されていないトピックのプロパティは削除されます。

複数のプロパティを指定する場合は、コンマで区切って指定します。設定可能なトピックのプロパティのリストは、[55 ページの「デスティネーションのプロパティ」](#) を参照してください。

## show bridge

```
show bridge topic|queue bridge_source
```

指定したトピックまたはキューに設定されたブリッジに関する情報を表示します。`bridge_source` には、ブリッジのソースとして設定されたトピックまたはキューの名前を指定します。

このコマンドの出力例を以下に示します。

Target Name	Type Selector
queue.dest	Q
topic.dest.1	T "urgency in ('high', 'medium')"
topic.dest.2	T

指定したデスティネーションに設定されているブリッジ先のデスティネーションの名前は、**Target Name** カラムにリストされます。ブリッジのタイプとメッセージセクタ（定義されている場合）が、それぞれ **Type** と **Selector** カラムに表示されます。

show bridges

```
show bridges [type=topic|queue] [pattern]
```

現在設定されているデスティネーションブリッジの概要を表示します。**type** オプションには、ブリッジのソースとして設定されたデスティネーションのタイプを指定します。たとえば、**show bridges topic** と指定すると、ブリッジのソースとして設定されたすべてのトピックに対して設定されたブリッジの概要が表示されます。**pattern** には、ソースのデスティネーション名に一致するパターンを指定します。たとえば、**show bridges foo.\*** と指定すると、**foo.\*** の名前に一致するすべてのソースデスティネーションに対して設定されたブリッジの概要が表示されます。**type** と **pattern** はオプションです。

このコマンドの出力例を以下に示します。

Source Name	Queue Targets	Topic Targets
Q queue.source	1	1
T topic.source	1	2

指定したパターンまたはタイプと一致するデスティネーションは、Source Name カラムに表示されます。各デスティネーションのキューへのブリッジの数は、Queue Targets カラムに表示されます。各デスティネーションのトピックへのブリッジの数は、Topic Targets カラムに表示されます。

show channel

```
show channel channel-name
```

特定のマルチキャストチャネルの詳細を表示します。**channel-name** は、特定のチャネルの正確な名前である必要があります。ワイルドカードを指定した名前や名前の一部は無効です。

このコマンドを指定すると、表 17 で説明されている情報を含むテーブルが表示されます。

表 17 出力フィールドの説明

見出し	説明
Channel	マルチキャストチャネルの名前。
Address	メッセージのブロードキャスト先のマルチキャストグループの IP アドレスおよびポートのデスティネーション。次の形式で指定します。 <multicast-group-IP-address> : <multicast-port>
TTL	チャネルのデータに対するネットワークホップの最大許容数。
Priority	EMS サーバーが帯域幅を割り当てるときのこのチャネルのメッセージの送信優先度。 最も高い優先度は - 5 で、最も低い優先度は 5 です。



表 17 出力フィールドの説明

見出し	説明
Max Rate	サーバーがチャネルを介してメッセージをブロードキャストする最大速度。
Max Time	サーバーが送信済みメッセージを再送信用に保持する秒単位の最長時間。
Interface	サーバーがこのチャネルでマルチキャストトラフィックを送信するために使用する IP アドレス。0.0.0.0 という値はデフォルトのインタフェースが使用されていることを示します。
Status	チャネルのステータス。active、inactive のいずれか。
Server Backlog	チャネルでブロードキャストが保留されている合計メッセージ数および合計バイト数。  バックログの制御についての詳細は、84 ページの「 <a href="#">マルチキャストとフロー制御</a> 」を参照してください。
Transmitted	チャネルで送信される合計バイト数。この数字には再送信は含まれません。
Retransmitted	チャネルで再送信される合計バイト数。
Retransmission Buffer	再送信のための現在バッファされる合計バイト数。

show channels

show channels

各チャネルのマルチキャストアドレスおよびステータスを含む、サーバーのマルチキャストチャネルの概要を表示します。

## show config

```
show config
```

接続されているサーバーの設定パラメータを表示します。以下の情報が出力されます。

- configuration files
- server database
- server JVM
- server JDBC database
- listen ports
- configuration settings
- message tracking
- server tracing parameters
- statistics settings
- fault-tolerant setup
- external transport setup
- server SSL setup

## show consumer

```
show consumer consumerID
```

指定したコンシューマの詳細情報を表示します。*consumerID* は、[show consumers](#) の出力から取得されます。

## show consumers

```
show consumers [topic=name | queue=name] [durable] [user=name]  
[connection=id] [sort=conn|user|dest|msgs] [full]
```

すべてのコンシューマまたは指定したフィルタに一致するコンシューマのみに関する情報を表示します。コマンドの出力を制御するには、**sort** または **full** のパラメータを指定します。**topic** または **queue** のパラメータを指定すると、指定したキューまたはトピックに一致するデスティネーションのコンシューマのみが表示されます。**user** または **connection** のパラメータを指定すると、指定したユーザーまたは接続のみに対応するコンシューマが表示されます。キューブラウザが開いている間は、EMS サーバー内のコンシューマとして表示されます。

`durable` パラメータを指定すると、持続的なトピックサブスクライバとキューレシーバのみが表示されますが、キューのコンシューマの表示は妨げません。持続的なトピックコンシューマのみを表示するには、次のコマンドを使用します。

`show consumers topic=> durable`

`sort` パラメータを指定すると、接続 ID、ユーザー名、デスティネーション名、または保留中のメッセージ数などの条件でコンシューマをソートできます。`full` パラメータを指定すると、以下に示すすべてのカラムが表示され、カラムは 120 ～ 140 文字以上表示できるように広げることができます。トピックとキューのコンシューマはそれぞれ別のテーブルに表示されます。トピックのコンシューマが先に表示され、キューコンシューマはその後に表示されます。

表 18 出力フィールドの説明

見出し	説明
Id	コンシューマの ID。
Conn	コンシューマの接続 ID または、これが切断されたトピックの持続サブスクライバの場合は、「-」。
Sess	コンシューマのセッション ID または、これが切断されたトピックの持続サブスクライバの場合は、「-」。
T	次のいずれかの文字で指定されるコンシューマタイプ。 トピックコンシューマ： <ul style="list-style-type: none"> <li>• T：トピックの非持続サブスクライバ。</li> <li>• D：トピックの持続サブスクライバ。</li> <li>• R：ルーティングされたトピックのためのシステム作成の持続サブスクライバ。</li> <li>• P：ルートのテンポラリトピックのプロキシサブスクライバ。</li> </ul> キューコンシューマ： <ul style="list-style-type: none"> <li>• Q：標準のキューレシーバ。</li> <li>• q：非アクティブのキューレシーバ。</li> <li>• P：ルートのひとつに作成されたレシーバのための、グローバルキューのシステム作成のレシーバ。</li> </ul>
Topic/Queue	サブスクリプションのトピックまたはキューの名前。
Name	(トピックのみ) 持続サブスクリプションの名前。少なくとも 1 人のトピックサブスクライバが持続サブスクライバである場合、このカラムが表示されます。

表 18 出力フィールドの説明

見出し	説明
SAS[NMB]	<p>カラムの説明：</p> <ul style="list-style-type: none"><li>• S：コンシューマの接続が開始されているときは「+」、それ以外は「-」。</li><li>• A：コンシューマのセッションの確認応答モード。値とその意味は次のとおり。<ul style="list-style-type: none"><li>— N：確認応答しない</li><li>— A：自動確認応答</li><li>— D：dups_ok 確認応答</li><li>— C：クライアント確認応答</li><li>— T：トランザクションのセッション</li><li>— X：XA または MS DTC セッション</li><li>— Z：接続コンシューマ</li></ul></li><li>• S：コンシューマにセレクトがあるときは「+」、それ以外は「-」。</li><li>• N：（トピックのみ）サブスクライバが「NoLocal」のときは「+」。</li><li>• M：（トピックのみ）サブスクライバが受信マルチキャストのときは「+」。</li><li>• B：（キューのみ）コンシューマがキューブラウザのときは「+」。</li></ul>
Pre	コンシューマのデスティネーションのプリフェッチ値。
Pre Dlv	コンシューマに配信されるプリフェッチウインドウメッセージの数。
Msgs Sent	コンシューマに送信されたメッセージのうち、コンシューマのセッションで確認応答されていないメッセージの数。
Size Sent	コンシューマに現在送信されているメッセージのうち、確認応答されていないメッセージの合計サイズ。値は丸められて、バイト、K（キロバイト）、M（メガバイト）、または G（ギガバイト）の単位で表示されます。
Pend Msgs	（トピックのみ）トピックのコンシューマのために保留中のメッセージの合計数。
Pend Size	（トピックのみ）トピックのコンシューマのために保留中のメッセージの合計サイズ。値は丸められて、バイト、K（キロバイト）、M（メガバイト）、または G（ギガバイト）の単位で表示されます。
Uptime	コンシューマの稼動時間。
Last Sent	サーバーがコンシューマにメッセージを最後に配信してから経過したおよその時間。値は、秒単位の近似値です。

表 18 出力フィールドの説明

見出し	説明
Last Akcd	コンシューマに配信されたメッセージがコンシューマのセッションで確認応答されてから経過したおよその時間。値は、秒単位の近似値です。
Total Sent	作成以降にコンシューマに送信されたメッセージの合計数。これには、セッションの回復またはロールバックによって再送されたメッセージの数も含まれます。
Total Acked	コンシューマに送信されたメッセージのうち、コンシューマの作成以降にコンシューマのセッションで確認応答されたメッセージの数。

## show connections

```
show connections [type=q|t|s] [host=hostname] [user=username]
[version] [address] [counts] [full]
```

クライアントとサーバー間の接続を表示します。150 ページの「表 20」に、出力内容を示します。

**type** パラメータで、表示する接続のサブセットを選択します（表 19 を参照）。**host** または **user** のパラメータを指定すると、特定のホストまたはユーザーに関する接続のみを表示することができます。**version** フラグが表示されている場合、クライアントのバージョン番号が表示されます。

**address** パラメータを指定すると、出力テーブルに IP アドレスが表示されます。**counts** パラメータを指定すると、プロデューサ、コンシューマ、およびテンポラリデスティネーションの数が表示されます。**full** パラメータを指定すると、出力可能なすべての情報が表示されます。

表 19 show connections コマンドの type パラメータ

Type	説明
type=q	キュー接続のみを表示します。
type=t	トピック接続のみを表示します。
type=s	システム接続のみを表示します。
指定なし	キューとトピックの接続は表示しますが、システム接続は表示しません。

表 20 出力フィールドの説明

見出し	説明
L	クライアントのタイプ。次のいずれかを指定できます。 <ul style="list-style-type: none"><li>• J : Java クライアント</li><li>• c : C クライアント</li><li>• # : C# クライアント</li><li>• - : 不明なシステムの接続</li></ul>
Version	EMS クライアントのバージョン。
ID	固有の接続 ID。接続の削除に使用できる、各接続に割り当てられている固有の数値 ID です。
FSXT	接続タイプの情報。 <p>F カラムは、接続がフォールトトレランスであるかどうかを示します。</p> <ul style="list-style-type: none"><li>• - : フォールトトレランス接続ではない。つまり、この接続には代替の URL がない。</li><li>• + : フォールトトレランス接続。つまり、この接続には代替の URL がある。</li></ul> <p>S カラムは、接続が SSL であるかどうかを示します。</p> <ul style="list-style-type: none"><li>• - : 接続が SSL ではない。</li><li>• + : 接続が SSL である。</li></ul> <p>X カラムは、接続が XA トランザクションであるか MS DTC トランザクションであるかどうかを示します。</p> <ul style="list-style-type: none"><li>• - : 接続が XA または MS DTC ではない。</li><li>• + : 接続が XA 接続または MS DTC 接続である。</li></ul> <p>T カラムは、接続のタイプを示します。</p> <ul style="list-style-type: none"><li>• C : 汎用的なユーザー接続</li><li>• T : ユーザー TopicConnection</li><li>• Q : ユーザー QueueConnection</li><li>• A : 管理上の接続</li><li>• R : 他のルートサーバーへのシステム接続</li><li>• F : フォールトトレランスサーバーへのシステム接続</li></ul>

表 20 出力フィールドの説明

見出し	説明
S	開始された接続のステータス。開始されている場合は「+」、停止中の場合は「-」。
IP Address	クライアントの IP アドレス。  このフィールドを表示するには、 <code>address</code> または <code>full</code> パラメータの指定が必要です。
Host	接続のホスト名（ホスト名を取得できない場合、このカラムにはホストの IP アドレスが表示されます）。
Address	接続の IP アドレス。  キーワード <code>address</code> を指定すると、このテーブルにこのカラムが表示されます。
User	接続ユーザー名。接続の作成時にユーザー名を指定しなかった場合、 <code>anonymous</code> がデフォルトのユーザー名として割り当てられます。
ClientID	接続のクライアント ID。
Sess	この接続のセッションの数。
Prod	この接続のプロデューサの数。  このフィールドを表示するには、 <code>counts</code> または <code>full</code> パラメータの指定が必要です。
Cons	この接続のコンシューマの数。  このフィールドを表示するには、 <code>counts</code> または <code>full</code> パラメータの指定が必要です。
TmpT	この接続で作成されたテンポラリトピックの数。バージョン 4.4 より前のクライアントでは、識別できないため「?」が表示されます。  このフィールドを表示するには、 <code>counts</code> または <code>full</code> パラメータの指定が必要です。
TmpQ	この接続で作成されたテンポラリキューの数。バージョン 4.4 より前のクライアントでは、識別できないため「?」が表示されます。  このフィールドを表示するには、 <code>counts</code> または <code>full</code> パラメータの指定が必要です。

表 20 出力フィールドの説明

見出し	説明
Uncomm	その接続でコミットされていないトランザクションのメッセージの数。  このフィールドを表示するには、 <code>counts</code> または <code>full</code> パラメータの指定が必要です。
UncommSize	その接続でコミットされていないトランザクションのバイト単位のメッセージの合計サイズ。  このフィールドを表示するには、 <code>counts</code> または <code>full</code> パラメータの指定が必要です。
Uptime	接続の稼働時間。

show db

`show db`

サーバーのデータベースの概要を表示します。推奨コマンドである `show stores` でもデータベースを表示します。

特定のデータベースについての詳細は、[162 ページ](#)の「`show store`」を参照してください。

show durable

`show durable durable-name`

持続サブスクライバの情報を表示します。

表 21 show durable テーブルの情報

見出し	説明
Durable Subscriber	持続サブスクライバの完全修飾名。この名前は、コロンで区切られたクライアント ID (ある場合) とサブスクリプション名で構成されます。
Subscription name	持続サブスクライバのフルネーム。
Client ID	サブスクライバの接続のクライアント ID。
Topic	持続サブスクライバがメッセージを受信するトピック。



表 21 show durable テーブルの情報（続き）

見出し	説明
Type	dynamic : クライアントによる作成 static : 管理者による設定
Status	online offline
UserName	持続サブスクライバ（クライアントの接続）のユーザー名。 持続サブスクライバがオフラインの場合、このカラムの値は offline になります。
Consumer ID	この内部 ID 番号が設定されていない場合、サーバーの外からは値を取得できません。
No Local	enabled : サブスクライバはローカル接続（サブスクライバと同じ接続）から送信されたメッセージを受信しません。 disabled : サブスクライバはすべての接続からメッセージを受信します。
Selector	サブスクライバはこのセレクトと一致するメッセージのみを受信します。
Pending Msgs	トピック内の全メッセージ数（この数には、配信済みのメッセージの数も含まれます）。
Delivered Msgs	持続サブスクライバに配信されているが、確認応答されていない、トピック内のメッセージ数。
Pending Msgs Size	保留中のすべてのメッセージの合計サイズ。

## show durables

show durables [pattern]

パターンを指定しない場合、すべてのトピックのすべての持続サブスクライバのリストを表示します。

パターンを指定した場合（foo.\* など）、パターンに一致する、トピックの持続サブスクライバのリストを表示します。

このコマンドを指定すると、表 22 で説明されている情報を含むテーブルが表示されます。

表 22 show durables テーブルの情報

見出し	説明
Topic Name	トピックの名前。  先頭にアスタリスク (*) が付いている名前は、動的な持続サブスクライバを示します。アスタリスク (*) の付いていない名前は、静的（管理者によって設定される）サブスクライバです。
Durable	持続サブスクライバのフルネーム。
User	この持続サブスクライバのユーザー名。持続サブスクライバがオフラインの場合、このカラムの値は <b>offline</b> になります。  外部定義されているユーザーは、名前の前にアスタリスク (*) が付いています。
Msgs	保留中のメッセージ数。
Size	保留中のメッセージの合計サイズ。

詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

show factory

```
show factory factory-name
```

指定したファクトリのプロパティを表示します。

show factories

```
show factories [generic|topic|queue]
```

すべてのファクトリを表示します。汎用、トピック、キューのいずれかのファクトリだけを出力するように指定して、表示されるリストを絞り込むことができます。

show jndiname

```
show jndiname jndi-name
```

指定された名前が JNDI サーバーによって結び付けられるオブジェクトを表示します。

## show jndinames

```
show jndinames [type]
```

オプションの *type* には、以下のパラメータを設定できます。

- destination
- topic
- queue
- factory
- topicConnectionFactory
- queueConnectionFactory

タイプを指定すると、指定されたタイプのオブジェクトに関連づけられている JNDI 名のみが表示されます。タイプを指定しない場合、すべての JNDI 名が表示されます。

## show group

```
show group group-name
```

グループ名、説明、グループ内のメンバーの数を表示します。

外部定義されているグループは、グループ名の前にアスタリスク (\*) が付いています。現在接続しているユーザーが少なくとも 1 人いる外部グループが表示されます。

## show groups

```
show groups
```

すべてのユーザーグループを表示します。

外部定義されているグループは、グループ名の前にアスタリスク (\*) が付いています。

## show members

```
show members group-name
```

指定したユーザーグループのすべてのメンバーを表示します。

## show message

```
show message messageID
```

指定したメッセージ ID のメッセージを表示します。

このコマンドを使用するには、`track_message_ids` 設定パラメータをオンにして、メッセージ ID による追跡を有効にする必要があります。

## show messages

```
show messages correlationID
```

JMSCorrelationID メッセージヘッダフィールドに指定した関連 ID のすべてのメッセージのメッセージ ID を表示します。`show message messageID` コマンドを使用すると、このコマンドで返される各 ID のメッセージを表示できます。

このコマンドを使用するには、`track_correlation_ids` 設定パラメータをオンにして、関連 ID による追跡を有効にする必要があります。

## show parents

```
show parents user-name
```

ユーザーの親グループを表示します。このコマンドは、ユーザーのアクセス権限を理解するのに役立ちます。

# show queue

`show queue queue-name`

指定したキューの詳細を表示します。



ルーティングされているキューの場合は、キュー名のみを指定します (queue-name@server の形式でサーバーを指定しないでください)。

表 23 show queue テーブルの情報

見出し	説明
Queue	キューのフルネーム。
Type	dynamic : クライアントによる作成 static : 管理者による設定
Properties	キューに設定されるプロパティの名前とその値を示すリスト。プロパティ名のリストについては、55 ページの「 <a href="#">デスティネーションのプロパティ</a> 」を参照してください。
JNDI Names	このキューを参照する、明示的に割り当てられた JNDI 名のリスト。
Bridges	このキューから他のデスティネーションへのブリッジのリスト。
Receivers	このキューのコンシューマの数。
Pending Msgs	このキュー内の全メッセージ数。後ろにかっこで囲まれた永続的メッセージ数が続きます。 この数には、配信済みのメッセージの数も含まれます。
Delivered Msgs	コンシューマに配信されているが、確認応答されていない、キュー内のメッセージ数。
Pending Msgs Size	すべての保留メッセージの合計サイズ。後ろにかっこで囲まれたすべての永続的メッセージ数のサイズが続きます。

## show queues

```
show queues [pattern-name [notemp|static|dynamic]
[first=n|next=n|last=n]]
```

*pattern-name* を指定しない場合、すべてのキューのリストを表示します。

*pattern-name* を指定した場合 (`foo.*` など)、そのパターンに一致するキューのリストを表示します。パターンに一致するトピックのリストをさらに絞るには、以下のパラメータのいずれかを使用します。

- `notemp` : テンポラリキューを表示しない
- `static` : 静的キューのみを表示する
- `dynamic` : 動的キューのみを表示する

*pattern-name* が指定されている場合、次のいずれかのコマンドを使用してキューのリストをカーソルで移動することもできます。ここで、*n* は整数です。

- `first=n` : 最初の *n* 個のキューを表示する
- `next=n` : 次の *n* 個のキューを表示する
- `last=n` : 次の *n* 個のキューを表示し、カーソルを終了する

カーソルは *n* 個のキューを調べ、*pattern-name* に一致するキューを表示します。キューのリストを完全に検索するわけではないため、カーソルがゼロ個または *n* 個未満のキューを戻す場合があります。一致するすべてのキューを検索するには、`Cursor complete` メッセージを受信するまで、続けて `next` を使用します。

`show queues` コマンドを指定すると、表 24 で説明されている情報を含むテーブルが表示されます。先頭にアスタリスク (\*) が付いているキューは、動的キューを示します。

表 24 show queues テーブルの情報

見出し	説明
Queue Name	キューの名前。名前の前にアスタリスク (*) が付いているキューは、テンポラリキューまたは動的に作成されたキューを示します。動的キューとテンポラリキューのプロパティは変更できません。

表 24 show queues テーブルの情報（続き）

見出し	説明
SNFGXIBCT	トピックのプロパティに関する情報を次の順番に表示します。 (S)ecure (N)sender_name or sender_name_enforced (F)ailsafe (G)lobal e(X)clusive (I)mport (B)ridge (C)flowControl (T)race 値の部分には以下の文字が指定されます。 - : プロパティがありません。 + : プロパティがあり、トピック自体に設定されています。 * : プロパティがあり、別のキューから継承されています。 継承されているプロパティは削除できません。
Pre	プリフェッチ値。アスタリスク (*) が後ろに付けられている値は、別のキューから継承された値か、デフォルトの値です。
Rcvrs	現在アクティブなレシーバの数。
All Msgs	
Msgs	保留中のメッセージ数。
Size	保留中のメッセージの合計サイズ。
Persistent Msgs	
Msgs	保留中の永続的メッセージ数。
Size	保留中の永続的メッセージの合計サイズ。

詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

show route

show route route-name

ルートのプロパティ（URL または SSL プロパティ）を表示します。

show routes

show routes

作成されているすべてのルートのプロパティ（URL または SSL プロパティ）を表示します。

これらのコマンドを指定すると、表 25 に示されている情報が表示されます。

表 25 show routes テーブルの情報

見出し	説明
Route	ルートの名前。
T	ルートのタイプ： <ul style="list-style-type: none"><li>• A：アクティブルート。</li><li>• P：パッシブルート。</li></ul>
ConnID	このサーバーからルートの末端のサーバーへの接続の固有の ID 番号。  このカラムのハイフン（-）は、ルートの接続先のサーバーが接続されていないことを示します。
URL	ルートの接続先のサーバーの URL。
ZoneName	ルートのゾーン名。
ZoneType	ゾーンのタイプ： <ul style="list-style-type: none"><li>• m：マルチホップゾーン。</li><li>• 1：1 ホップゾーン。</li></ul>

show rvctransportledger

show rvctransportledger transport\_name [subject-or-wildcard]

指定したトランスポートと指定したサブジェクトに対応する、TIBCO Rendezvous 保証メッセージ（RVCM:TIBCO Rendezvous Certified Messaging）元帳ファイルのエントリを表示します。サブジェクト名を指定するか、ワイルドカードを使用して一致するすべてのサブジェクトを取得するか、またはサブジェクト名を指定しないで、元帳ファイルのすべてのエントリを取得することができます。

元帳ファイルと元帳ファイルのエントリの形式についての詳細は、TIBCO Rendezvous のマニュアルを参照してください。



## show rvcmlisteners

```
show rvcmlisteners
```

`create rvcmlistener` コマンドを使用するか、`tibrvcml.conf` ファイルを編集して作成された、すべての RVCML リスナーを表示します。

## show server

```
show server (aliases: info, i)
```

接続されているサーバーの名前とそのサーバーに関する情報を表示します。

## show stat

```
show stat channel name [topic=name]
```

```
show stat consumers [topic=name|queue=name] [user=name]
                    [connection=id] [total]
```

```
show stat producers [topic=name|queue=name] [user=name]
                    [connection=id] [total]
```

```
show stat route name [topic=name|queue=name] [total] [wide]
```

```
show stat topic name [total] [wide]
```

```
show stat queue name [total] [wide]
```

指定したアイテムの統計情報を表示します。コンシューマ、プロデューサ、ルート、デスティネーション、またはチャンネルの統計情報を表示できます。統計情報を表示するには、統計情報の収集を有効にする必要があります。また、詳細情報統計の追跡を有効にすると、各アイテムの詳細な情報を表示できます。`rate_interval` 設定パラメータで間隔を指定すると、インバウンド / アウトバウンドメッセージの数とメッセージサイズの平均値を求めることができます。

`total` キーワードを指定すると、メッセージの合計数とアイテムのメッセージサイズの合計のみが表示されます。`wide` キーワードを指定すると、インバウンドとアウトバウンドのメッセージ統計情報を同じ行に表示します。

統計情報の詳細な説明および統計情報収集オプションの有効化 / 無効化についての詳細は、448 ページの「[サーバーの統計情報の操作](#)」を参照してください。

show store

```
show store store-name
```

特定のストアの詳細を表示します。このコマンドを使用して、ファイルベースのストアまたはデータベースストアのいずれかの詳細を取得できます。

store-name は、特定のストアの正確な名前である必要があります。

このコマンドを指定すると、表 26 で説明されている情報を含むテーブルが表示されます。

表 26 show store テーブルの情報

見出し	説明
ストア	ストアの名前。
Type	ストアのタイプ： <ul style="list-style-type: none"><li>file はファイルベースのストアを示します。</li><li>dbstore はデータベースストアを示します。</li><li>mstore は mstore を示します。</li></ul>
Message Count	ファイルに保存されているメッセージ数。
Swapped Count	プロセスメモリからストアファイルにスワップされたメッセージ数。
Average Write Time	書き込み呼び出しにかかる秒単位の平均時間（非同期ファイルストアでは使用できない）。
Write Usage	server_rate_interval で指定された時間に対する書き込み呼び出しにかかった時間の割合。（非同期ファイルストアでは使用できない）。
ファイルベースのストアに固有の見出し	
File	このストアファイルに関連付けられているファイル名。stores.conf ファイルの file パラメータで設定されます。
Access Mode	asynchronous：サーバーは非同期 I/O 呼び出しを使用してファイルにメッセージを保存します。 synchronous：サーバーは同期 I/O 呼び出しを使用してファイルにメッセージを保存します。
Pre-allocation Minimum	このファイルに事前に割り当てられているディスク領域がある場合、そのディスク領域の容量。

表 26 show store テーブルの情報

見出し	説明
CRC	<p><b>enabled</b> : サーバーは CRC を使用して、ストアファイルの読み取り時にチェックサムデータを検証します。</p> <p><b>disabled</b> : サーバーはストアファイルの読み取り時にチェックサムデータを検証しません。</p>
Periodic Truncation	<p><b>enabled</b> : EMS サーバーが未使用のディスク領域を解放するために不定期にストアファイルの切り捨てを行います。</p> <p><b>disabled</b> : EMS サーバーが未使用のディスク領域を解放するためにストアファイルの切り捨てを行いません。</p>
Destination Defrag Batch Size	デスティネーションデフラグ機能で使用するバッチのサイズ。
File Size	未使用の割り当て済みのファイル領域含む、ストアファイルのサイズ。
Free Space	未使用の割り当て済みのファイル領域のサイズ。
Fragmentation	ファイルのフラグメンテーションのレベル。
Used Space	ファイルの使用領域のサイズ。
Message Size	ファイル内の全メッセージの合計サイズ。
Swapped Size	ファイル内のスワップされたメッセージの合計サイズ。
Disk Write Rate	1 秒あたりに書き込まれるバイト数。
<b>mstore に固有の見出し</b>	
mstore の出力には、ファイルベースのストアで使用できるフィールドと同じフィールドが多くあります。	
Discard Scan Interval	EMS サーバーが mstore の全メッセージの検査にかかる最長時間。この期間は、 <b>stores.conf</b> ファイルの <b>scan_iter_interval</b> パラメータにより制御されます。
Discard Scan Interval Bytes	Discard Scan Interval ごとに読み取り、処理されるバイト数。この数は、mstore のファイルサイズに比例しており、ストレージメディアの制限内に保持される必要があります。詳細は、32 ページの「 <a href="#">mstore の間隔について</a> 」を参照してください。

表 26 show store テーブルの情報

見出し	説明
First Scan Finished	<p><b>true</b> : EMS サーバーの起動以降、ストアのすべてのデータが少なくとも一度検査された。</p> <p><b>false</b> : EMS サーバーの前回の起動以降、検査されていないデータがある。<b>false</b> の場合、特定のサーバー統計 ([Message Count] フィールドなど) は、ストア内にまだある期限切れのメッセージやパージされたメッセージにより少なく報告される場合があります。詳細は、34 ページの「統計情報への影響」を参照してください。</p>
Disk Write Rate	1 秒あたりに書き込まれるバイト数。
データベースストアに固有の見出し	
JDBC Driver Name	JDBC データベースサーバーの名前。
JDBC URL	JDBC データベースサーバーの場所。
UserName	データベースへのアクセスに使用する EMS サーバーのユーザー名。
Dialect	SQL コマンドの構築に使用する SQL ダイアレクト。

show stores

show stores

サーバーのストアのリストを表示します。

show topic

show topic *topic-name*

表 27 show topic テーブルの情報

見出し	説明
Topic	トピックのフルネーム。
Type	<p><b>dynamic</b> : クライアントによる作成</p> <p><b>static</b> : 管理者による設定</p>

表 27 show topic テーブルの情報 (続き)

見出し	説明
Properties	トピックに設定されるプロパティの名前とその値を示すリスト。プロパティ名のリストについては、55 ページの「 <a href="#">デスティネーションのプロパティ</a> 」を参照してください。
JNDI Names	このトピックを参照する、明示的に割り当てられた JNDI 名のリスト。
Bridges	このトピックから他のデスティネーションへのブリッジのリスト。
コンシューマ	このトピックのコンシューマの数 (持続コンシューマの数も含まれます)。
Durable Consumers	このトピックの持続コンシューマの数。
Pending Msgs	送信済みであるものの、コンシューマによって確認応答されていないメッセージの合計数。後ろにかっこで囲まれた永続的メッセージ数が続きます。これらの数には複数のサブスクリバに送信されたコピーが含まれます。
Pending Msgs Size	すべての保留メッセージの合計サイズ。後ろにかっこで囲まれたすべての永続的メッセージ数のサイズが続きます。
サーバーが以下の統計情報を蓄積するには、管理者が統計情報を有効にする必要があります。有効にしない場合、これらのアイテムの値はゼロです。	
Total Inbound Msgs	トピックに配信されたすべてのメッセージの累積数。
Total Inbound Bytes	トピックに配信されたすべてのメッセージサイズを累積した合計。
Total Outbound Msgs	コンシューマによってトピックからコンシュームされたメッセージの累積数。メッセージの各コンシューマは他のコンシューマとは独立してこの数字を増やすため、ひとつのインバウンドメッセージは、 $n$ のアウトバウンドメッセージ (コンシューマひとつあたりひとつ) を持ちます。
Total Outbound Bytes	コンシューマによって、トピックからコンシュームされたすべてのメッセージサイズを累積した合計。メッセージの各コンシューマは、他のコンシューマとは独立して、この数値を増やします。

show topics

```
show topics [pattern-name [notemp|static|dynamic]
[first=n|next=n|last=n]]
```

pattern-name を指定しない場合、すべてのトピックのリストを表示します。

pattern-name を指定した場合 (foo.\* など)、そのパターンに一致するトピックのリストを表示します。パターンに一致するトピックのリストをさらに絞るには、以下のパラメータのいずれかを使用します。

- notemp : テンポラリトピックを表示しない
- static : 静的トピックのみを表示する
- dynamic : 動的トピックのみを表示する

pattern-name が指定されている場合、次のいずれかのコマンドを使用してトピックのリストをカーソルで移動することもできます。ここで、n は整数です。

- first=n : 最初の n 個のトピックを表示する
- next=n : 次の n 個のトピックを表示する
- last=n : 次の n 個のトピックを表示し、カーソルを終了する

カーソルは n 個のトピックを検査し、pattern-name に一致するトピックを表示します。トピックのリストを完全に検索するわけではないため、カーソルがゼロ個または n 個未満のトピックを戻す場合があります。一致するすべてのトピックを検索するには、Cursor complete メッセージを受信するまで、続けて next を使用します。

show topics コマンドを指定すると、表 28 で説明されている情報を含むテーブルが表示されます。

表 28 show topics テーブルの情報 ( 1 / 2 シート)

見出し	説明
Topic Name	トピックの名前。名前の前にアスタリスク (*) が付けられているトピックは、テンポラリトピックまたは動的に作成されたトピックを示します。動的トピックとテンポラリトピックのプロパティは変更できません。

表 28 show topics テーブルの情報 （2 / 2 シート）

見出し	説明
SNFGEIBCTM	<p>トピックのプロパティに関する情報を次の順番に表示します。</p> <p>(S)ecure (N)sender_name or sender_name_enforced (F)ailsafe (G)lobal (E)xport (I)mport (B)ridge (C)flowControl (T)race (M)ulticast</p> <p>値の部分には以下の文字が指定されます。</p> <p>- : プロパティがありません。</p> <p>+ : プロパティがあり、トピック自体に設定されています。</p> <p>* : プロパティがあり、別のトピックから継承されています。</p> <p>継承されているプロパティは削除できません。</p>
Subs	このトピックの現在のサブスクライバの数（持続サブスクライバも含む）。
Durs	このトピックの持続サブスクライバの数。
All Msgs	
Msgs	<p>送信済みであるものの、コンシューマによって確認応答されていない永続的メッセージの合計数。この数には複数のサブスクライバに送信されたコピーが含まれます。</p> <p>すべてのデスティネーションに送信される実際のメッセージ数（トピックのサブスクライバ数で乗算しない）を確認するには、<code>show server</code> コマンドを使用します。</p>
Size	保留中のメッセージの合計サイズ。
Persistent Msgs	
Msgs	送信済みであるものの、コンシューマによって確認応答されていないメッセージの合計数。
Size	保留中の永続的メッセージの合計サイズ。

詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

show transaction

```
show transaction XID
```

指定されたトランザクション内で送受信されたメッセージのリストを表示します。このコマンドでは、prepared（準備完了）、ended（終了）、roll back（ロールバック）状態のみのトランザクションの情報を返します。suspended（停止中）または active（アクティブ）状態は含まれません。

表 29 に各カラムに表示される情報を説明します。

表 29 show transactions テーブルの情報（1 / 2 シート）

見出し	説明
State	トランザクションの状態： <ul style="list-style-type: none"><li>• <b>A</b> : active（アクティブ）</li><li>• <b>E</b> : ended（終了）</li><li>• <b>R</b> : rollback only（ロールバックのみ）</li><li>• <b>P</b> : prepared（準備完了）</li><li>• <b>S</b> : suspended（停止中）</li></ul> 停止中のトランザクションはロールバックできますが、ロールフォワード（コミット）はできません。
Remaining time before timeout	TX タイムアウトに達するまでの残りの秒数。たとえば、3 sec などです。 このフィールドは、ENDSUCCESS 状態または ROLLBACKONLY 状態のトランザクションのみに適用できます。
コンシュームされるメッセージ	
Message ID	メッセージのメッセージ ID。null では、取得できなかったか、無効にされたメッセージ ID を示します。 track_message_ids を有効化しない場合、このフィールドには Disabled が表示されます。
Type	メッセージの送信先のデスティネーションタイプ： <ul style="list-style-type: none"><li>• <b>Q</b> キュー</li><li>• <b>T</b> トピック</li></ul>
Destination	メッセージが送信されたデスティネーションの名前。null は、デスティネーションが見つからなかったことを示します。



表 29 show transactions テーブルの情報 (2 / 2 シート)

見出し	説明
Consumer ID	メッセージをコンシュームするコンシューマのコンシューマ ID。ゼロはコンシューマがオフラインであることを示します。
生成されるメッセージ	
Message ID	メッセージのメッセージ ID。null では、取得できなかったか、無効にされたメッセージ ID を示します。 <code>track_message_ids</code> を有効化しない場合、このフィールドには Disabled が表示されます。
Type	メッセージの送信先のデスティネーションタイプ : <ul style="list-style-type: none"> <li>• Q キュー</li> <li>• T トピック</li> </ul>
Destination	メッセージが送信されたデスティネーションの名前。null は、デスティネーションが見つからなかったことを示します。
JMSTimestamp	メッセージの作成時刻を示すタイムスタンプ。

## show transactions

show transactions

XA または MS DTC インタフェースを使用して作成されたすべてのクライアントトランザクションの **XID** を表示します。各行には、ひとつのトランザクションに関する情報が表示されます。**XID** は、トランザクションの [Format ID]、[GTrid Len]、[Bqual Len] および [Data] フィールドの連結です。たとえば、トランザクションが次の行を返したとします。

State	Format ID	GTrid Len	Bqual Len	Data
E	0	6	2	branchid

この場合、XID は 0 6 2 branchid です。スペースは必須です。

表 30 に各カラムに表示される情報を説明します。

表 30 show transactions テーブルの情報

見出し	説明
State	トランザクションの状態： <ul style="list-style-type: none"><li>• <b>A</b> : active (アクティブ)</li><li>• <b>E</b> : ended (終了)</li><li>• <b>R</b> : rollback only (ロールバックのみ)</li><li>• <b>P</b> : prepared (準備完了)</li><li>• <b>S</b> : suspended (停止中)</li></ul> 停止中のトランザクションはロールバックできますが、ロールフォワード (コミット) はできません。
Format ID	XA トランザクション形式の ID。 0 = OSI CCR 命名が使用されます。 >0 = 他の形式が使用されます。 -1 = NULL
GTrid Len	グローバルトランザクション ID を構成するバイト数。
Bqual Len	ブランチ修飾子を構成するバイト数。
Data	グローバルトランザクション ID (gtrid) とブランチ修飾子 (bqual)。

show transport

```
show transport transport
```

[transports.conf](#) に定義されている、指定したトランスポートの設定を表示します。

詳細は、392 ページの「[Rendezvous のためのトランスポートの設定](#)」と 415 ページの「[SmartSockets のためのトランスポートの設定](#)」を参照してください。

show transports

```
show transports
```

[transports.conf](#) で設定されているトランスポートの名前をすべて表示します。

## show user

```
show user user-name
```

ユーザー名とその説明を表示します。ユーザー名を指定しない場合、現在ログインしているユーザーを表示します。

外部定義されているユーザーは、名前の前にアスタリスク (\*) が付いています。

## show users

```
show users
```

すべてのユーザーを表示します。

外部定義されているユーザーは、名前の前にアスタリスク (\*) が付いています。現在接続している外部ユーザーのみを表示します。

## showacl admin

```
showacl admin
```

すべてのユーザーとグループの管理アクセス権限をすべて表示しますが、デスティネーションの管理アクセス権限は表示しません。

## showacl group

```
showacl group group-name [admin]
```

指定したグループのすべてのアクセス権限を表示します。グループとアクセス権限のセットを表示します。オプションで **admin** を指定すると、デスティネーションまたはプリンシパルの管理アクセス権限のみを表示することができます。**showacl admin** を指定すると、すべてのユーザーとグループの管理アクセス権限をすべて表示します（デスティネーションの管理アクセス権限は含まれません）。

## showacl queue

```
showacl queue queue-name [admin]
```

キューのすべてのアクセス権限を表示します。**acl** ファイルのすべてのエントリをリストします。各エントリは、「アクセス権限を付与された」ユーザーまたはグループとアクセス権限のセットを表示します。オプションで **admin** を指定すると、デスティネーションまたはプリンシパルの管理アクセス権限のみを表示することができます。**showacl admin** を指定すると、すべてのユーザーとグループの管理アクセス権限をすべて表示します（デスティネーションの管理アクセス権限は含まれません）。

## showacl topic

```
showacl topic topic-name [admin]
```

トピックのすべてのアクセス権を表示します。**acl** ファイルのすべてのエントリをリストします。各エントリは、「アクセス権を付与された」ユーザーまたはグループとアクセス権のセットを表示します。オプションで **admin** を指定すると、デスティネーションまたはプリンシパルの管理アクセス権のみを表示することができます。**showacl admin** を指定すると、すべてのユーザーとグループの管理アクセス権をすべて表示します（デスティネーションの管理アクセス権は含まれません）。

## showacl user

```
showacl user user-name [admin | all | admin-all]
```

ユーザーおよび、ユーザーに付与されている、デスティネーションとプリンシパルへのアクセス権のセットを表示します。

**showacl user username** : ユーザーに直接付与されているアクセス権を表示します（管理者はこの形式のコマンドを使用して、他のユーザーのアクセス権を表示する権限がない場合でも、管理者自身のアクセス権を表示できます）。

**showacl user username admin** : ユーザーに直接付与されている管理アクセス権を表示します。

**showacl user username all** : 直接付与されたアクセス権および（そのユーザーが属するグループから）継承されたアクセス権を表示します。

**showacl user username admin-all** : ユーザーに対し、直接付与された、または継承したすべての管理アクセス権を表示します。



このコマンドによる出力では、継承されたアクセス権は先頭にアスタリスク (\*) が付けられて表示されます。継承されたアクセス権は変更できません。プリンシパルユーザーに継承されているアクセス権を取り消そうとしても、アクセス権は変更されません。

## shutdown

```
shutdown
```

現在接続されているサーバーをシャットダウンします。

## suspend route

```
suspend route route-name
```

指定したルートに送信されるメッセージを停止します。

メッセージのフローはコマンド **resume route** を使用して後で回復できます。

## time

```
time [on | off]
```

on を指定すると、各コマンドが出力される前に、タイムスタンプが作成されます。デフォルトでは、タイムスタンプは off にされています。

## timeout

```
timeout [seconds]
```

現在のコマンドのタイムアウト値を表示または変更します。タイムアウト値は、コマンドを送信した後に、管理ツールがサーバーからの応答を待機する秒数を指定します。

デフォルトでは、タイムアウトは 30 秒に設定されています。オプションの *seconds* パラメータとともに **timeout** が設定されると、タイムアウト値は、指定された秒数にリセットされます。パラメータを指定せずに設定すると、現在のタイムアウト値が返されます。

## transaction commit

```
transaction commit XID
```

トランザクション ID で識別されたトランザクションをコミットします。トランザクションの状態は、ended または prepared のいずれかです。トランザクション ID を取得するには、show transactions コマンドを実行し、XID をこのコマンドに貼り付けます。

## transaction rollback

```
transaction rollback XID
```

トランザクション ID で識別されたトランザクションをロールバックします。トランザクションの状態は、ended、rollback only、または prepared のいずれかです。トランザクション ID を取得するには、show transactions コマンドを実行し、XID をこのコマンドに貼り付けます。



**prefetch=none** および **maxRedelivery=number** の設定で、キューに送信されたメッセージは、ループでメッセージを受信し、XA が準備段階の後に XA のロールバックを行う EMS アプリケーションでは、*number* に指定した回数で受信されません。

## updatecrl

```
updatecrl
```

サーバーの失効した証明書のリスト (CRL) を直ちに更新します。

## whoami

`whoami`

現在ログインしているユーザーを表示する `show user` コマンドのエイリアスです。

## 第 7 章

# 設定ファイルの使用

この章では、TIBCO Enterprise Message Service の設定について説明します。

### トピック

---

- [「設定ファイルの場所」](#)、176 ページ
- [「設定の仕組み」](#)、176 ページ
- [「tibemsd.conf」](#)、177 ページ
- [「その他の設定ファイルの使用」](#)、228 ページ

## 設定ファイルの場所

---

インストールプロセスにより、設定ファイルは以下の 2 か所のディレクトリに置かれます。

- `config-file-directory/cfmgmt/ems/data/` には、インストールをすばやくテストするのに適した設定ファイルのサブセットがあります。`config-file-directory` は、設定ディレクトリステップのインストールプロセス時に指定されます。
- `EMS_HOME/samples/config/` には、サンプル設定ファイルの完全なセットがあります。実装する場合は、このディレクトリから実稼動設定ディレクトリにファイルをコピーし、コピーしたファイルを編集することをお勧めします。

実稼動設定ディレクトリを選択する場合、信頼性と災害復旧の必要に応じて、定期的にバックアップが行われるファイルシステムを使用することをお勧めします。EMS サーバーは、設定ディレクトリに対して読み取りおよび書き込み権限を持っている必要があります。

## 設定の仕組み

---

**設定ファイル** EMS サーバーは、起動時に一度だけ設定ファイルを読み取りますが、それ以降の設定ファイルへの変更は無視します。設定ファイルを変更した場合、EMS 管理ツールから `shutdown` コマンドを実行してサーバーをシャットダウンし、101 ページの「[EMS サーバーの実行](#)」の説明どおりにサーバーを再起動します。

**管理要求** `tibemsadmin`（コマンドラインツール）、Java または .NET の管理 API、または TIBCO Administrator™（別の TIBCO 製品）を使用して、管理要求を使ってサーバー設定を変更することもできます。

サーバーは、管理要求を検証して受け入れるときに、変更内容を適切な設定ファイルに書き込みます（設定ファイルに手動で加えられた変更を上書きします）。このポリシーにより、サーバーの再起動時（フォールトトレランス機能が稼動している状況や、ハードウェア障害の後など）に、設定ファイルが最新状態に保たれます。

EMS を再インストールまたは更新すると、`bin/` および `samples/config/` の各ディレクトリにあるファイルは上書きされます。実装を設定する際に、これらのディレクトリは使用しないでください。



## tibemsd.conf

主要設定ファイルは、EMS サーバーの特性を制御します。このファイルは通常、`tibemsd.conf` という名前ですが、サーバーの起動時に別の名前を指定することもできます。サーバーの起動についての詳細は、101 ページの「[EMS サーバーの実行](#)」を参照してください。

`tibemsd.conf` ファイルのサンプルは、`config-file-directory/cfmgmt/ems/data/` ディレクトリにあります。ここで、`config-file-directory` は TIBCO Enterprise Message Service のインストール時に指定されます。この設定ファイルはテキストエディタを使って編集できます。この設定ファイルの一部の設定アイテムは管理ツールを使用しても変更できますが、ほとんどの設定パラメータはファイルを編集することにより変更する必要があります（そうしないと、サーバーは、これらのパラメータに対する変更を受け入れません）。管理ツールの使用についての詳細は、117 ページの第 6 章、「[EMS 管理ツールの使用](#)」を参照してください。

いくつかのパラメータはブール値を取ります。パラメータの記述には、特定の値セットがひとつ指定されています（たとえば、`enable` と `disable` など）。ただし、ブール値を取るすべてのパラメータは、次の値を持つことができます。

- `enable`、`enabled`、`true`、`yes`、`on`
- `disable`、`disabled`、`false`、`no`、`off`

複数の要素を取るパラメータには、要素が二重引用符で囲まれていない限り、要素間にスペースを含めることはできません。パラメータの行の長さは、256,000 文字以下に制限されます。

次の表は、`tibemsd.conf` のパラメータをカテゴリ別にまとめたものです。以降のセクションでは、各パラメータについてより詳しく説明します。

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<b>グローバルなシステムのパラメータ</b>		
<code>authorization</code>	サーバー認可を有効または無効にする。	189
<code>clock_sync_interval</code>	EMS サーバーの UTC 時間をクライアントに定期的に送信する。	189
<code>compliant_queue_ack</code>	クライアントが、ルーティングされているキューから受信したメッセージの確認応答に成功したら、そのメッセージを再配信しないことを保証する。	189

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<code>disconnect_non_acking_consumers</code>	コンシューマ内の確認応答されていない保留メッセージのサイズと数を確認する。	190
<code>flow_control</code>	デスティネーションのフロー制御を有効または無効にする。	190
<code>listen</code>	クライアントからの接続をサーバーがリスンするポートを指定する。	191
<code>max_msg_field_print_size</code>	トレーシングメッセージの文字列フィールドのサイズを制限する。	191
<code>max_msg_print_size</code>	トレースされたメッセージの出力されたメッセージのサイズを制限する。	191
<code>npsend_check_mode</code>	プロデューサからの <code>NON_PERSISTENT</code> メッセージの受信確認をサーバーがいつ提供するかを指定する。	192
<code>password</code>	<code>authorization</code> が有効になっている他のルーティング先サーバーで認証を受けるために使用されるパスワード。	192
<code>processor_ids</code>	ネットワーク I/O トラフィックで使用するプロセッサを指定する。	193
<code>routing</code>	このサーバーのルーティング機能を有効または無効にする。	194
<code>selector_logical_operator_limit</code>	セレクトタの評価時にサーバーが確認する演算子の数を制限する。	194
<code>server</code>	サーバーの名前です。	194
<code>startup_abort_list</code>	初期化シーケンス中にサーバーが終了する条件を指定する。	195
<code>user_auth</code>	EMS サーバーへのアクセスを試みるユーザーの認証に使用される認証情報のソースを指定する。	196
<code>xa_default_timeout</code>	XA トランザクションの TX タイムアウトを指定する。	196

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<b>ストレージファイルのパラメータ</b>		
<code>store</code>	サーバーがデータを保存するディレクトリを指定する。	196
<b>接続とメモリのパラメータ</b>		
<code>destination_backlog_swapout</code>	メッセージスワップが有効になる前にサーバーに保存されるデスティネーションあたりのメッセージの最大数を指定する。	197
<code>handshake_timeout</code>	EMS サーバーが SSL 接続の完了を待機する時間を指定する。	197
<code>max_client_msg_size</code>	受信メッセージの最大サイズを設定する。	197
<code>max_connections</code>	サーバーに同時に接続できるクライアントの最大数を指定する。	197
<code>max_msg_memory</code>	サーバーがメッセージに使用できるメモリの最大量を指定する。	198
<code>msg_pool_block_size</code>	メッセージの保存用に、サーバーによって事前に割り当てられるプールのサイズを指定する。	198
<code>msg_swapping</code>	メッセージスワップを有効または無効にする。	199
<code>reserve_memory</code>	緊急時に使用するために予約しておくメモリの容量を指定する。	199
<code>socket_send_buffer_size</code>	EMS サーバーへの接続時にクライアントによって使用される、送信バッファのサイズを設定する。	200
<code>socket_receive_buffer_size</code>	EMS サーバーへの接続時にクライアントによって使用される、受信バッファのサイズを設定する。	200

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<b>ネットワーク接続障害の検出パラメータ</b>		
<code>client_heartbeat_server</code>	クライアントがサーバーにハートビートを送信する間隔を指定する。	201
<code>server_timeout_client_connection</code>	クライアント接続を打ち切る前に、サーバーがクライアントハートビートを待機する時間を指定する。	201
<code>server_heartbeat_server</code>	このサーバーが別のサーバーにハートビートを送信する間隔を指定する。	202
<code>server_timeout_server_connection</code>	別のサーバーへの接続を打ち切る前に、このサーバーが接続先サーバーからのハートビートを待機する時間を指定する。	202
<code>server_heartbeat_client</code>	このサーバーが、そのすべてのクライアントにハートビートを送信する間隔を指定する。	202
<code>client_timeout_server_connection</code>	接続を打ち切る前に、クライアントがサーバーからのハートビートを待機する時間を指定する。	202
<b>フォールトトレランスのパラメータ</b>		
<code>ft_active</code>	アクティブなサーバーの URL を指定する。	203
<code>ft_heartbeat</code>	稼動中であることを示すために、アクティブなサーバーがハートビート信号をバックアップサーバーに送信する間隔を指定する。	203
<code>ft_activation</code>	アクティブなサーバーに障害が発生したと見なす前に、バックアップサーバーがハートビート信号を待機する間隔の最大値を指定する。	203
<code>ft_reconnect_timeout</code>	プライマリサーバーの役割を持つサーバーがフェイルオーバー状況にあると判断した後、クライアントの再接続をバックアップサーバーが待機する時間の最大値を指定する。	204

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<code>ft_ssl_auth_only</code>	サーバーが、フォールトトレランスサーバーが認証時にのみ SSL の使用を要求することを許可するかどうかを指定する。	204
<code>ft_ssl_identity</code>	サーバーのデジタル証明書を指定する。	204
<code>ft_ssl_issuer</code>	サーバーの証明書チェーンメンバーを指定する。	204
<code>ft_ssl_private_key</code>	サーバーの秘密鍵を指定する。	205
<code>ft_ssl_password</code>	秘密鍵のパスワードを指定する。	205
<code>ft_ssl_trusted</code>	信頼する証明書のリストを指定する。	205
<code>ft_ssl_rand_egd</code>	インストール済みの EGD (Entropy Gathering Daemon) のパスを指定する。	205
<code>ft_ssl_verify_host</code>	フォールトトレランスサーバーに、別のサーバーの証明書を検証させるかどうかを指定する。	206
<code>ft_ssl_verify_hostname</code>	フォールトトレランスサーバーに、別のサーバーの証明書の CN フィールドが指定された名前であることを検証させるかどうかを指定する。	206
<code>ft_ssl_expected_hostname</code>	フォールトトレランスサーバーの CN フィールドにサーバーが指定することを予期される名前を指定する。	206
<code>ft_ssl_ciphers</code>	サーバーで使用される暗号化スイートを指定する。	206
<b>メッセージ追跡のパラメータ</b>		
<code>track_message_ids</code>	メッセージ ID によるメッセージ追跡を有効または無効にする。	207
<code>track_correlation_ids</code>	相関 ID によるメッセージ追跡を有効または無効にする。	207

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<b>マルチキャストパラメータ</b>		
<code>multicast</code>	EMS サーバーのマルチキャストを有効または無効にする。	207
<code>multicast_channels</code>	マルチキャストチャネルが定義されている設定ファイルを指定する。	207
<code>multicast_daemon_default</code>	EMS クライアントからの接続をマルチキャストデーモンがリスンするデフォルトポートを指定する。	208
<code>multicast_statistics_interval</code>	マルチキャストの統計情報が各チャネルに生成される間隔を秒単位で指定する。	208
<b>TIBCO Rendezvous パラメータ</b>		
<code>tibrv_transports</code>	<code>transports.conf</code> ファイルで定義されている TIBCO Rendezvous トランスポートを有効または無効にする。	209
<b>TIBCO SmartSockets パラメータ</b>		
<code>module_path</code>	TIBCO SmartSockets ライブラリファイルが含まれるディレクトリを指定する。	209
<code>tibss_transports</code>	<code>transports.conf</code> ファイルで定義されている TIBCO SmartSockets トランスポートを有効または無効にする。	209
<code>tibss_config_dir</code>	SmartSockets 設定ファイルとメッセージファイルのディレクトリを指定する。	210
<b>ログファイルパラメータのトレース</b>		
<code>logfile</code>	サーバーログファイルの名前と場所を指定する。	210
<code>log_trace</code>	<code>logfile</code> パラメータによって指定されているファイルのトレースオプションを指定する。	210
<code>logfile_max_size</code>	ログファイルの最大サイズを指定する。最大サイズに達すると、ログファイルはバックアップにコピーされ、空にされる。	211

表 31 tibemspd.conf パラメータ

パラメータ	説明	参照ページ
<code>console_trace</code>	<code>stderr</code> に出力するトレースオプションを指定する。	211
<code>client_trace</code>	接続の開始および終了、メッセージアクティビティ、トランザクションアクティビティに関する、クライアントによるトレース出力の生成を有効または無効にする。	211
<code>trace_client_host</code>	接続に関連するトレースステートメントが、ホストを識別する方法を指定する (ホスト名、IP アドレス、またはその両方)。	212
<b>統計情報収集のパラメータ</b>		
<code>server_rate_interval</code>	サーバーの全般統計情報の平均値を求める間隔を指定する。	212
<code>statistics</code>	プロデューサ、コンシューマ、デスティネーション、およびルートを収集する統計情報を有効または無効にします。	213
<code>rate_interval</code>	ルート、デスティネーション、プロデューサ、コンシューマに関する統計情報の平均値を求める間隔を指定する。	213
<code>detailed_statistics</code>	オブジェクトに対して詳細統計情報の追跡を行うかどうかを指定します。	213
<code>statistics_cleanup_interval</code>	デスティネーションにアクティビティがない場合、サーバーが詳細統計情報を保持する期間を指定する。	213
<code>max_stat_memory</code>	詳細統計情報の収集に使用するメモリの最大容量を指定します。	214
<b>SSL サーバーのパラメータ</b>		
<code>ssl_dh_size</code>	Diffie-Hellman 鍵のサイズを指定する。	214
<code>ssl_server_ciphers</code>	サーバーで使用される暗号化スイートを指定する。	214

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<code>ssl_require_client_cert</code>	デジタル証明書を持つクライアントからの SSL 接続のみをサーバーが受け入れるかどうかを指定する。	214
<code>ssl_use_cert_username</code>	クライアントのユーザー名を、常にクライアントのデジタル証明書の CN フィールドから抽出するかどうかを指定する。	215
<code>ssl_cert_user_specname</code>	デジタル証明書からユーザー名を取得する必要があるクライアントを識別するために使用される、特別なユーザー名を指定する。	215
<code>ssl_server_identity</code>	サーバーのデジタル証明書を指定する。	216
<code>ssl_server_key</code>	サーバーの秘密鍵を指定する。	216
<code>ssl_password</code>	秘密鍵のパスワードを指定する。	216
<code>ssl_server_issuer</code>	サーバーの証明書チェーンメンバーを指定する。	217
<code>ssl_server_trusted</code>	クライアント証明書の発行者としてサーバーが信頼する、CA ルート証明書のリストを指定する。	217
<code>ssl_rand_egd</code>	インストール済みの EGD (Entropy Gathering Daemon) のパスを指定する。	217
<code>ssl_crl_path</code>	失効した証明書のリスト (CRL : Certificate Revocation List) ファイルへのパスを指定する。	217
<code>ssl_crl_update_interval</code>	サーバーが CRL を更新する間隔を指定する。	218
<code>ssl_auth_only</code>	サーバーが、クライアントが認証時にのみ SSL の使用を要求することを許可するかどうかを指定する。	218
<code>fips140-2</code>	FIPS コンプライアンスのサーバーを有効化する。	218



表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<b>LDAP パラメータ</b>		
<code>ldap_url</code>	外部のディレクトリサーバーの URL を指定する。	218
<code>ldap_principal</code>	LDAP 管理者の識別名 (DN : Distinguished Name) を指定する。	219
<code>ldap_credential</code>	<code>ldap_principal</code> プロパティで定義されているユーザーに関連付けられているパスワードを指定する。	219
<code>ldap_cache_enabled</code>	LDAP データのキャッシュを有効または無効にする。	219
<code>ldap_cache_ttl</code>	キャッシュされた LDAP データが更新される前に、保持される最大時間を指定する。	219
<code>ldap_conn_type</code>	サーバーによる LDAP 情報の取得に使用される接続の種類を指定する。	219
<code>ldap_tls_cacert_file</code>	EMS サーバーが LDAP サーバーの証明書に署名するときに信頼する CA 証明書が含まれているファイルを指定する。	220
<code>ldap_tls_cacert_dir</code>	確認チェーンにふたつ以上の CA 証明書がある場合、このパラメータを使用して、CA 証明書が含まれているディレクトリを指定する。	220
<code>ldap_tls_cipher_suite</code>	セキュアな LDAP 接続の暗号化に使用される暗号スイートを指定する。	220
<code>ldap_tls_rand_file</code>	暗号化用のランダムなデータが含まれているファイルを指定する。	220
<code>ldap_tls_cert_file</code>	EMS サーバーを LDAP サーバーに対して識別する証明書が含まれているファイルを指定する。	221
<code>ldap_tls_key_file</code>	LDAP サーバーがクライアントを認証するのに必要な秘密鍵が含まれているファイルを指定する。	221

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<code>ldap_user_class</code>	ユーザーを格納する LDAP オブジェクトクラスの名前を指定する。	<a href="#">221</a>
<code>ldap_user_attribute</code>	ユーザーの名前を保持するユーザーオブジェクトクラスの属性の名前を指定する。	<a href="#">221</a>
<code>ldap_user_base_dn</code>	ユーザーを格納する LDAP ツリーのベース識別名 (DN : Distinguished Name) を指定する。	<a href="#">221</a>
<code>ldap_user_scope</code>	ベース DN からどのくらい下位の階層までユーザーを検索するかを指定する。	<a href="#">222</a>
<code>ldap_user_filter</code>	指定されたユーザー名の検索に使用する LDAP 検索フィルタを指定する。	<a href="#">222</a>
<code>ldap_all_users_filter</code>	特定のユーザーベース DN を持つすべてのユーザーの検索に使用する、LDAP 検索フィルタを指定する。	<a href="#">222</a>
<code>ldap_group_base_dn</code>	グループを格納する LDAP ツリーのベース識別名 (DN : Distinguished Name) を指定する。	<a href="#">222</a>
<code>ldap_group_scope</code>	ベース DN からどのくらい下位の階層までグループを検索するかを指定する。	<a href="#">222</a>
<code>ldap_group_filter</code>	指定されたグループ名を持つグループの検索に使用する LDAP 検索フィルタを指定する。	<a href="#">223</a>
<code>ldap_all_groups_filter</code>	特定のグループベース DN を持つすべてのグループの検索に使用する、LDAP 検索フィルタを指定する。	<a href="#">223</a>
<code>ldap_static_group_class</code>	静的グループを格納する LDAP オブジェクトクラスの名前を指定する。	<a href="#">223</a>
<code>ldap_static_group_attribute</code>	グループの名前を保持する静的グループオブジェクトクラスの属性の名前を指定する。	<a href="#">223</a>

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
<code>ldap_static_member_attribute</code>	グループメンバーの識別名 (DN : Distinguished Name) を指定する LDAP 静的グループオブジェクトの属性を指定する。	<a href="#">224</a>
<code>ldap_dynamic_group_class</code>	動的グループを格納する LDAP オブジェクトクラスの名前を指定する。	<a href="#">224</a>
<code>ldap_dynamic_group_attribute</code>	グループの名前を保持する動的オブジェクトクラスの属性の名前を指定する。	<a href="#">224</a>
<code>ldap_dynamic_member_url_attribute</code>	動的グループメンバーの URL を指定する、動的 LDAP グループオブジェクトの属性を指定する。	<a href="#">224</a>
<b>拡張可能なセキュリティパラメータ</b>		
<code>jaas_classpath</code>	JAAS LoginModule によって使用される JAR ファイルと依存クラスを含める。	<a href="#">225</a>
<code>jaas_config_file</code>	JAAS 設定ファイルの場所を指定する。このファイルは、カスタム認証 LoginModule を実行するために使用されます。	<a href="#">225</a>
<code>jaas_login_timeout</code>	サーバーが JAAS 認証モジュールの実行と応答を待機する時間をミリ秒単位で指定する。	<a href="#">225</a>
<code>jaci_classpath</code>	JACI カスタムアクセス権限モジュールによって使用される JAR ファイルと依存クラスを含める。	<a href="#">225</a>
<code>jaci_class</code>	拡張可能なアクセス権限インタフェースを実装するクラスの名前を指定する。	<a href="#">226</a>
<code>jaci_timeout</code>	サーバーが JACI アクセス権限モジュールの実行と応答を待機する時間をミリ秒単位で指定する。	<a href="#">226</a>

表 31 tibemsd.conf パラメータ

パラメータ	説明	参照ページ
JVM パラメータ		
jre_library	EMS サーバーの JVM を有効化する。	226
jre_option	起動時に JVM にコマンドラインオプションを渡す。	227

## グローバルなシステムのパラメータ

### authorization

`authorization = enabled | disabled`

サーバー認可を有効または無効にする。

デフォルトでは、**authorization** は無効です。セキュアなデスティネーションに対しても、ユーザー認証情報とアクセス権限を、サーバーで確認する必要がある場合、このパラメータを「**enabled**」に設定する必要があります。

詳細は、265 ページの「[アクセス制御の有効化](#)」を参照してください。

例：

`authorization = enabled`

これらのパラメータについての詳細は、255 ページの第 8 章、「[認証とアクセス権限](#)」を参照してください。

### clock\_sync\_interval

`clock_sync_interval = seconds`

EMS サーバーの UTC（協定世界時）時間をクライアントに定期的に送信します。これにより、EMS クライアントはオフセットを更新できます。

秒単位で指定された時間で、**clock sync** コマンドがサーバーからそのクライアントに送信される間隔を特定します。

省略されたり、ゼロが指定されたりしている場合、EMS サーバーは、EMS クライアントがサーバーに接続しているときに、オフセット時間のみを送信します。**clock\_sync\_interval** が -1 の場合は、クライアントがサーバーに接続しているときでもオフセットは送信されません。クライアントは時間の値をサーバー時間に一致するように調整することができません。

### compliant\_queue\_ack

`compliant_queue_ack = enable | disable`

クライアントが、ルーティングされているキューから受信したメッセージの確認応答に成功したら、そのメッセージを再配信しないことを保証します。これは、応答をクライアントに送信する前に、キューのホーム EMS サーバーが、メッセージが正常に確認応答されるまで待機することにより行われます。

デフォルトでは、**compliant\_queue\_ack** は有効です。**compliant\_queue\_ack** を有効にするとオーバーヘッドが発生するため、パフォーマンスを低下させたくない場合はこの機能を無効にしてください。**compliant\_queue\_ack** が無効に設定されていて、メッセージが再配信される場合、メッセージの **JMSRedelivered** インジケータが設定されます。

## disconnect\_non\_acking\_consumers

```
disconnect_non_acking_consumers = enabled | disabled
```

このパラメータは **maxbytes** および **maxmsgs** デスティネーションプロパティと連携して動作します。コンシューマがメッセージをコンシュームするものの、確認応答していない状況では、メッセージは確認されるまでサーバーに保持されます。この場合は、設定の制限を超えてサーバーをプッシュできます。

このパラメータが有効な場合、**disconnect\_non\_acking\_consumers** によりサーバーは、コンシューマに送信する保留中のメッセージ数およびサイズを確認します。**maxbytes** または **maxmsgs** の制限に到達し、コンシューマがそのメッセージに確認応答しなかった場合、サーバーはコンシューマに送信されるメッセージを破棄し、コンシューマの接続を解除します。これにより、メッセージに確認応答せずにメッセージをコンシュームするアプリケーションからサーバーが保護されます。

このプロパティを有効化する前に、**maxbytes** および **maxmsgs** の制限が、**prefetch** 設定、トランザクションのサイズ（受信が処理された場合）、または、クライアントまたは明示的なクライアント確認応答モードを使用して確認応答されたメッセージの数を参照して、設定されていることを確認します。そうしないと、メッセージに確認応答する前にサーバーがコンシューマの接続を解除する可能性があります。

ルートを実装する場合は、すべてのルーティング先サーバーで同一の **disconnect\_non\_acking\_consumers** 設定を使用する必要があります。また、**maxbytes** または **maxmsgs** がグローバルデスティネーション用に設定されている場合、すべてのサーバーに同じ設定を適用する必要があります。ルートの接続を解除すると適切に動作しているアプリケーションに影響する可能性があるため、サーバーはルーティング先コンシューマを破棄または接続解除しません。サーバーはローカルコンシューマを破棄および接続解除します。これに応じて関連する他のサーバーがそれらのサーバーのリモートコンシューマのメッセージを認識し、破棄します。

デフォルトでは、このパラメータは無効化されています。

## flow\_control

```
flow_control = enable | disable
```

デスティネーションのフロー制御を有効または無効にするかを指定します。

デフォルトでは、フロー制御は無効です。フロー制御が有効になっている場合、各デスティネーションの **flowControl** プロパティは、デスティネーションの保留中のメッセージを保存する最大サイズ（目標値）を指定します。

フロー制御についての詳細は、83 ページの「**フロー制御**」を参照してください。

## listen

```
listen=protocol://servername:port
```

クライアントからの接続をサーバーがリスンするポートを指定する。

例：

```
listen=tcp://localhost:7222
```

SSL を有効にしている場合、以下のようになります。

```
listen=ssl://localhost:7222
```

複数のインタフェースを持つコンピュータでは、複数の `listen` エントリを使用することができます。例：

```
listen=tcp://localhost:7222
```

```
listen=tcp://localhost:7224
```

`localhost` が指定されていない場合、または `servername` が存在しない場合、サーバーは使用可能なすべてのインタフェースを使用します。例：

```
listen=tcp://7222
```

```
listen=ssl://7243
```

ホスト名の代わりに、IP アドレスを使用できます。例：

```
listen=tcp://192.168.10.107:7222
```

IPv6 アドレスを指定する場合は、指定するアドレスを角かっこで囲みます。例：

```
listen=tcp://[2001:cafe::107]:7222
```

## max\_msg\_field\_print\_size

```
max_msg_field_print_size = size [KB|MB|GB]
```

トレーシングメッセージの文字列フィールドのサイズを制限する。文字列フィールドが `size` より大きい場合、トレーシングメッセージでこのフィールドは切り捨てられます。

32 ビット符号付整数を `KB`、`MB`、または `GB` として指定します。最小許容サイズは、1 KB です。デフォルトでは、このフィールドの制限は 1 KB です。

## max\_msg\_print\_size

```
max_msg_print_size = size [KB|MB|GB]
```

トレースされたメッセージの出力されたメッセージのサイズを制限する。メッセージが「`size`」より大きい場合、そのメッセージは切り捨てられます。

32 ビット符号付整数を `KB`、`MB`、または `GB` として指定します。最小許容サイズは、8 KB です。デフォルトでは、このフィールドの制限は 8 KB です。

## npsend\_check\_mode

`npsend_check_mode = [always | never | temp_dest | auth | temp_auth]`  
 プロデューサからの **NON\_PERSISTENT** メッセージの受信確認をサーバーがいつ提供するかを指定する。

`npsend_check_mode` パラメータは、**NON\_PERSISTENT** 配信モードと非トランザクションセッションを使用してメッセージを送信するプロデューサにのみに適用されます。

メッセージの確認機能はパフォーマンスへの影響が大きいので、必要な場合にのみ有効にしてください。プロデューサが、サーバーに確認の **NON\_PERSISTENT** メッセージを送信させる状況は、以下のとおりです。

- **authorization** が有効になっている場合に、プロデューサのメッセージ送信アクセス権限がサーバーによって拒否されたとき。
- メッセージをテンポラリデスティネーションに送信する場合に、すでに破棄されたテンポラリデスティネーションにメッセージが送信されたとき。
- メッセージがキュー / トピックのしきい値を超過した場合（トピックに **rejectIncoming** ポリシーが必要）。
- メッセージのブリッジに失敗した場合。
- サーバーのメモリが不足しているか、またはそれ以外のサーバーエラーが発生している場合。

利用可能な `npsend_check_mode` パラメータモードは以下のとおりです。

- **default**（モードを指定しない）：EMS 4.3 およびそれ以前のバージョンと同じ動作。つまり、サーバーが **NON\_PERSISTENT** メッセージの確認を実行するのは、**authorization** が有効になっている場合のみ。
- **always**：サーバーは **NON\_PERSISTENT** メッセージの確認を常に実行する。
- **never**：サーバーは **NON\_PERSISTENT** メッセージの確認を常に実行しない。
- **temp\_dest**：サーバーは、メッセージがテンポラリデスティネーションに送信される場合のみ、**NON\_PERSISTENT** メッセージの確認を実行する。
- **auth**：サーバーは、接続の作成時に **authorization** が有効になっていた場合のみ、**NON\_PERSISTENT** メッセージの確認を実行する。
- **temp\_auth**：サーバーは、メッセージがテンポラリデスティネーションに送信される場合、または接続の作成時に **authorization** が有効になっていた場合のみ、**NON\_PERSISTENT** メッセージの確認を実行する。

## password

`password = password`

**authorization** が有効になっている他のルーティング先 **server** にログインするときに使用されるパスワードを指定します。詳細は、519 ページの「**ルーティングと認可**」を参照してください。



## processor\_ids

```
processor_ids = processor-id1,processor-id2,...
```

このパラメータを設定すると、リストで指定されているプロセッサ ID があるため、EMS server は多数のネットワーク I/O スレッドとして起動されます。各ネットワーク I/O スレッドは、特定のプロセッサ ID にバインドされています。つまり、スレッドはプロセッサ上でのみ実行できます。



デフォルトの動作で十分なスループットが得られる場合は、このパラメータは使用しません。

*processor-id* は 0 で始まり、利用可能なプロセッサ数から 1 を差し引いた数までの整数として指定します。たとえば、四つのプロセッサがある場合、有効なプロセッサ ID は 0、1、2、および 3 です。ID は任意の順序で指定できます。

次の場合、起動時に、パラメータが解析され、server の起動は拒否されます (startup\_abort\_list パラメータの指定の有無には関係ありません)。

1. リストの形式が正しくない。つまり、数値でない要素などの無効な値を含む場合。
2. マシンで使用可能なプロセッサ数より大きいプロセッサ ID が指定されている場合。たとえば、四つのプロセッサのみを備えるマシンで、プロセッサ ID 4 が指定されている場合などです。(四つのプロセッサを備えるマシンの有効な ID は、0、1、2、および 3 です)。
3. server は特定のプロセッサ ID にネットワーク I/O スレッドにバインドすることができません。これは、プロセッサ ID が無効化されたり、tibemsd プロセスがシステム管理者によって、このプロセッサ ID を含まないプロセッサのセットに制限された場合に発生する可能性があります。



ハイパースレディングを使用するべきではありません。

たとえば、24 のプロセッサ、ふたつのダイを備え、最初のダイのプロセッサ ID が 0 ~ 5 と 12 ~ 17、および 2 番目のダイのプロセッサが 6 ~ 11 および 18 ~ 23 であるマシンについて検討します。この例では、0 ~ 5 の範囲か 6 ~ 11 の範囲のいずれかで、プロセッサ ID を指定する必要があります。

リストのプロセッサ ID を 0 ~ 12 に指定すると、ふたつのネットワーク I/O スレッドが同一のプロセッサ (またはコア) にバインドされているため、スラッシングが発生します。また、最適なパフォーマンスを得るためには、プロセッサ ID は同じダイ上のものである必要があります。

このパラメータは stores.conf パラメータの [processor\\_id](#) と併用できます。詳細は、115 ページの「[パフォーマンス調整](#)」を参照してください。

## routing

```
routing = enabled | disable
```

このサーバーのルーティング機能を有効または無効にします。

例：

```
routing = enabled
```

ルーティングについての詳細は、497 ページの第 20 章、「ルートの使用」を参照してください。

## selector\_logical\_operator\_limit

```
selector_logical_operator_limit = number
```

セレクトタの評価時にサーバーが確認する演算子の数を制限します。

サーバーは、指定された *number* (数) の *false* 条件に達するまで演算子进行评估します。指定の数に到達すると、サーバーは再帰的な評価をそれ以上行わないようにするために以降の評価を停止します。多数の OR 条件が含まれるなど、非常に長い *selector* 句の場合は、再帰的にセレクトタの評価が実行され、EMS サーバーのスタックオーバーフローが発生する可能性があります。

*number* には、任意の正の整数を指定できます。デフォルトの値は、5000 です。ゼロは特別な値で、制限がないことを示します。

たとえば、*selector\_logical\_operator\_limit* = 10 でセレクトタが次のようになっている場合を考えます。

```
a=1 or b=2 or c=3 or d=4 or e=5 or f=6 or g=7 or h=8 or i=9 or j=10  
or k=11 or l=12 or m=13 or n=14 or o=15 or p=16 or q=17 or r=18 or  
s=19 or t=20 or u=21 or v=22 or w=23 or x=24 or y=25 or z=26
```

最初の 10 個の条件が *false* である場合は、サーバーはそれ以降の評価を行いません。

## server

```
server = serverName
```

サーバーの名前です。

サーバー名は 64 文字以内で指定する必要があります。

## startup\_abort\_list

```
startup_abort_list=[SSL,TRANSPORTS,CONFIG_FILES,CONFIG_ERRORS,
DB_FILES,MULTICAST]
```

初期化シーケンス中にサーバーが終了する条件を指定します。

各条件の任意の組み合わせをコンマで区切ったリストで指定することができます。リストには、要素が二重引用符で囲まれていない限り、要素間にスペースを含めることはできません。スペースが含まれているが、引用符で囲まれていない場合、サーバーはスペースの後に指定されている条件を無視します。

リストに指定されていない条件は、サーバーによって無視されます。デフォルトでは、リストは空です。

指定できる条件は以下のとおりです。

- **SSL** : SSL の初期化に失敗した場合、サーバーは終了する。
- **TRANSPORTS** : 設定ファイルで指定されたとおりに作成されなかったトランスポートがある場合、サーバーは終了する。
- **CONFIG\_FILES** : `tibemsd.conf` に指定された設定ファイルのいずれかが存在しない場合、サーバーは終了する。
- **CONFIG\_ERRORS** : 設定ファイルの読み取り中に何らかのエラーを検出した場合、サーバーは終了する。

JSON 設定を使用して `tibemsd` を起動するとき、不明なパラメータはサイレントに無視されます。たとえば、`tibemsd.json` ファイルに廃止されたパラメータが含まれていても、エラーはスローされません。

- **DB\_FILES** : サーバーのひとつ以上のストアが見つからない場合、サーバーは終了する。ストアにはデフォルトのストアファイル、および `stores.conf` 設定ファイルで設定されている任意のファイルストアまたはデータベースストアが含まれます。

`DB_FILES` が `startup_abort_list` に含まれていない場合、サーバーはストアを見つけることができず、欠落しているファイルまたはデータベースを作成します。最良の結果を得るためには、サーバーの初回の起動時は、サーバーがファイルを作成できるように `DB_FILES` を含めないようにします。初回の起動または主要なストア設定の変更（新しいストアの追加など）の後には、サーバーの再起動時に、すべての設定済みのファイルが存在する場合に限りサーバーが起動するようにリストに `DB_FILES` を含めます。

- **MULTICAST** : マルチキャストメッセージを送信できないことをサーバーが検出した場合、サーバーは終了する。

`MULTICAST` が `startup_abort_list` になく、マルチキャストの初期化が失敗した場合は、マルチキャスト対応のトピックでコンシューマを作成するアプリケーションは、TCP を介してメッセージを受信します。これは、TCP 接続を介してメッセージが送信されているときに、ネットワークがマルチキャストに割り当てられている帯域幅を処理できない場合に検討することが重要です。

## user\_auth

```
user_auth = [local, ldap, jaas]
```

ユーザー認証情報のソースを指定します。

このパラメータは、以下の値をひとつ以上指定できます（複数指定する場合は、コンマで区切ります）。

- **local** : ローカル EMS サーバーのユーザー設定からユーザーの認証情報を取得する。
- **ldap** : LDAP ディレクトリサーバーからユーザーの認証情報を取得する (LDAP 固有の設定パラメータを参照してください)。
- **jaas** : カスタム認証モジュールからユーザーの認証情報を取得する (282 ページの「[拡張可能な認証](#)」を参照してください)。

ユーザーが認証を試みるたびに、サーバーは、対応する認証情報を取得するために、このパラメータで指定された場所を指定された順番で検索します。EMS サーバーは、指定されたソースのいずれかを使って認証に成功した場合、その認証を受け入れます。



**user\_auth** 設定はデフォルトの管理者の認証には影響しません。サーバーは常に、ローカル設定ファイルから **admin** ユーザーを認証します。詳細は、120 ページの「[管理者へのパスワードの割り当て](#)」を参照してください。

## xa\_default\_timeout

```
xa_default_timeout = seconds
```

XA トランザクションのデフォルトの TX タイムアウトを秒単位で指定します。デフォルトは 0 であり、タイムアウトを指定しません。

デフォルトのタイムアウト設定は動的に変更することはできません。ただし、API を使用して、個々の XA リソースごとの他のトランザクションタイムアウトを指定することはできます。

## ストレージファイルのパラメータ

ここで説明するパラメータでは、ファイルベースのストアと **mstore** を設定します。データベースストアについての詳細は、[第 10 章](#)、「[データベースストアの使用](#)」を参照してください。

### store

```
store = directory
```

サーバーがデータファイルを保存するディレクトリです。例：

```
store = /usr/tmp
```

## 接続とメモリのパラメータ

### destination\_backlog\_swapout

`destination_backlog_swapout = number`

メッセージスワップが有効になる前にサーバーのメモリに保存できるメッセージ数を指定します。指定した制限は各デスティネーションに適用されます。たとえば、制限が 10,000 で三つのキューがある場合、サーバーのメモリにスワップされていないメッセージを 30,000 個まで保存できます。

*number* には任意の正の値を指定できます。`destination_backlog_swapout` が 0 の場合、サーバーは即座にメッセージのスワップを試みます。

デフォルトでは、各デスティネーションの制限は 1024 メッセージです。

### handshake\_timeout

`handshake_timeout = seconds`

EMS サーバーが SSL 接続の完了を待機する時間（秒単位）を指定します。*seconds* には、任意の正の整数を指定できます。デフォルトの値は、3 秒です。

タイムアウトに達すると、EMS サーバーは SSL 接続を閉じて他のクライアントに対するサービスを続行します。

### max\_client\_msg\_size

`max_client_msg_size = size [KB|MB|GB]`

受信メッセージの最大許容サイズを指定します。

このパラメータを設定すると、サーバーは指定したサイズ制限を超える受信メッセージを拒否します。

整数を KB、MB、または GB 単位で指定します。最大値は 2 GB です。

省略するか、ゼロを指定した場合、EMS サーバーは任意のサイズのメッセージを受信して処理を試みます。

### max\_connections

`max_connections = number`

同時に接続できるクライアント接続の最大数を指定します。

ゼロに設定すると、同時接続数に制限がなくなります。

## max\_msg\_memory

```
max_msg_memory = size [KB|MB|GB]
```

サーバーがメッセージに使用できるメモリの最大量です。

サーバーでメッセージに使用されるメモリ量を、このパラメータを使って制限することで、サーバーのメモリ使用量がシステムのメモリ容量を超えないようにできます。

`msg_swapping` が有効になっている場合、メッセージがこのしきい値を超過すると、サーバーはプロセスメモリからディスクへのメッセージのスワップを開始します。スワップすることで、サーバーは、着信メッセージのためにプロセスメモリを解放し、しきい値を超えたメッセージボリュームを処理できるようになります。

サーバーがメッセージをディスクにスワップした場合でも、スワップされたメッセージごとに、小さなレコードがメモリに常駐します。すべてのメッセージがディスクにスワップされた後、メモリ内のレコードだけでメモリ制限を超えてしまった場合は、サーバーに新しい着信メッセージを受け取るための空き容量はありません。サーバーは新しいメッセージの受け入れを停止し、メッセージプロデューサによる `send` 呼び出しはエラーになります。この状況は、多くの場合、このパラメータに設定されている値が小さすぎるか、さもなければ非常に大量のメッセージが存在することを示しています。

単位としては、KB、MB、または GB を指定できます。最小値は 8 MB です。0（ゼロ）のデフォルト値は、制限がないことを意味します。

例：

```
max_msg_memory = 512MB
```

## msg\_pool\_block\_size

```
msg_pool_block_size size
```



このパラメータを使用する前に、TIBCO サポート担当にご相談ください。

`malloc` および `free` に付随するオーバーヘッドコストを減らすために、サーバーはメッセージのストレージ用プールを事前に割り当てます。このパラメータは、これらのプールの動作を決定します。パフォーマンスは、オペレーティングシステムプラットフォームと使用パターンによって異なります。

`size` 引数は、ブロックまたはプールが収容できる内部メッセージ構造体のおよその個数を決定します（バイト数ではありません）。

`msg_pool_block_size` は、「拡張可能な」プールを割り当てるようサーバーに指示します。サーバーがプールを使い切るたびに、追加のストレージが利用可能な限り、サーバーはここで指定されているサイズを単位としてプールを拡張します。値は、32 ～ 65536 の範囲で指定します。

このパラメータを指定しなかった場合、デフォルトは `msg_pool_block_size 128` です。

## msg\_swapping

```
msg_swapping = enable | disable
```

このパラメータは、スワップ機能を有効または無効にします（スワップ機能については、前述の `max_msg_memory` にも説明があります）。

明示的に `disabled` を設定した場合を除き、デフォルト値は、`enabled` です。

## reserve\_memory

```
reserve_memory = size
```

`reserve_memory` にゼロ以外が指定されている場合、緊急時に使用するためのメモリブロックをデーモンが割り当てておくことで、メモリ不足の状況下で EMS サーバーが不安定になるのを防ぎます。デーモンプロセスがメモリリソースを使い切ってしまうと、デーモンプロセスは、クライアントとルートが新しいメッセージを作成することを禁止します。さらに、デーモンプロセスは、このメモリのブロックを解放することで、コンシューマが処理を続行できるようにします（処理を続行することによって、結果的にメモリが解放されます）。

EMS サーバーは、サーバーで保留中のメッセージ数が割り当てエラーの発生時に保留中だったメッセージ数の 10% まで減ったときに、その予約メモリの再割り当てを試みます。メモリが正常に再割り当てされると、サーバーは新しいメッセージの受信を開始します。

`reserve_memory` パラメータは、EMS サーバーがメモリ不足になった場合にのみトリガーされる、対応メカニズムの一種です。予約メモリの解放が、EMS サーバーによりトリガーされた場合の適切な対応は次のとおりです。まず、大部分のメッセージをコンシュームさせてクリアし、それから EMS サーバーを停止してから再起動します。この操作により、オペレーティングシステムが、EMS サーバーによって消費されていた仮想メモリリソースを再び利用できるようになります。トレースオプション `MEMORY` を使用して、サーバーがメッセージを受信していないときに行っている処理を表示することもできます。

`size` を MB 単位で指定します。ゼロ以外を指定した場合、最小のブロックは 16MB です。このパラメータを指定しなかった場合、デフォルト値はゼロです。



EMS サーバーが極端に多くのメッセージを貯めこまないようにするために、各種の制限を設定することができます。メッセージを貯めこみすぎると、EMS サーバーのメモリ不足を招く恐れがあります。この制限には、グローバルパラメータの `max_msg_memory` や、デスティネーションプロパティの `maxbytes` などがあります。これらの制限を使用して、`reserve_memory` メカニズムがトリガーされないようにしてください。

## socket\_send\_buffer\_size

```
socket_send_buffer_size = size [KB|MB|GB]
```

EMS サーバーへの接続時にクライアントによって使用される、送信バッファのサイズ（バイト単位）を設定します。

*size* には次のいずれかを指定できます。

- 512 より大きい任意の数。
- 0（デフォルトのバッファサイズを使用）。
- -1（指定したバッファの呼び出しをスキップ）。
- オプションで、単位として KB、MB、または GB を指定する。単位を指定しない場合、サイズはバイト単位で扱われます。

省略するか、ゼロを指定した場合、デフォルトのバッファサイズが使用されます。



Linux プラットフォームでは、パラメータを省略するかゼロを指定すると、EMS サーバーはその値をスキップします。この場合、Linux の自動調整によってバッファが制御されます。

## socket\_receive\_buffer\_size

```
socket_receive_buffer_size = size [KB|MB|GB]
```

EMS サーバーへの接続時にクライアントによって使用される、受信バッファのサイズ（バイト単位）を設定します。

*size* には次のいずれかを指定できます。

- 512 より大きい任意の数。
- 0（デフォルトのバッファサイズを使用）。
- -1（指定したバッファの呼び出しをスキップ）。
- オプションで、単位として KB、MB、または GB を指定する。単位を指定しない場合、サイズはバイト単位で扱われます。

省略するか、ゼロを指定した場合、デフォルトのバッファサイズが使用されます。



Linux プラットフォームでは、パラメータを省略するかゼロを指定すると、EMS サーバーはその値をスキップします。この場合、Linux の自動調整によってバッファが制御されます。



## ネットワーク接続障害の検出パラメータ

この機能を使用して、サーバーとクライアントは、ネットワーク接続の障害をすばやく検出することができます。これらのパラメータが指定されていない場合、またはこの機能が無効になっている場合、tibemspd は、オペレーティングシステムから通知されたときにだけ接続を切断します。

### client\_heartbeat\_server

```
client_heartbeat_server = interval
```

サーバーからクライアントへの接続では、このパラメータで指定した間隔（秒単位）でクライアントがサーバーにハートビートを送信します。

client\_heartbeat\_server パラメータは、server\_timeout\_client\_connection が設定されている場合に指定する必要があります。client\_heartbeat\_server の間隔は、server\_timeout\_client\_connection の制限の 3 分の 1 以下にする必要があります。

また、この設定によって接続時にガベージコレクションが確実に実行されます。コレクションは、受信メッセージとハートビートによってトリガーされます。メッセージのサイズが大きく異なる場合、またはメッセージトラフィックが不安定な場合、このパラメータを使用して確実にコレクションを実行できます。

省略するか、ゼロを指定した場合、client\_heartbeat\_server は無効になります。

### server\_timeout\_client\_connection

```
server_timeout_client_connection = limit
```

サーバーからクライアントへの接続では、このパラメータで指定した期間（秒単位）が過ぎてもサーバーがハートビートを受け取らなかった場合、サーバーは接続を切断します。

この期間には、client\_heartbeat\_server で指定したハートビート間隔の約 3 倍の値を設定することをお勧めします。



server\_timeout\_client\_connection が指定されていて client\_heartbeat\_server パラメータを設定しない場合、起動時に設定エラーが発生します。CONFIG\_ERRORS が [startup\\_abort\\_list](#) に含まれる場合、サーバーは起動しません。含まれない場合、エラーが表示されますがサーバーは起動します。また、クライアントは server\_timeout\_client\_connection 秒後に切断されます。

ゼロは特別な値で、サーバー側のハートビート検出を無効にします（ただし、クライアントはハートビートを送信し続けます）。

## server\_heartbeat\_server

```
server_heartbeat_server = interval
```

サーバーからサーバーへの接続では、このパラメータで指定した間隔（秒単位）でサーバーがハートビートを送信します。ふたつのサーバーは、ルート経由か、フォールトトレランスペアとして接続されます。

## server\_timeout\_server\_connection

```
server_timeout_server_connection = limit
```

サーバからサーバへの接続では、このパラメータで指定した期間（秒単位）が過ぎてもサーバーがハートビートを受け取らなかった場合、サーバーは接続を切断します。このパラメータは、他のルートからの接続と、バックアップサーバーへの接続にのみ適用されます。

この期間には、接続先サーバーのハートビート間隔の約 3.5 倍の値を設定することをお勧めします。接続先サーバーまたはネットワークの負荷が高い場合、あるいはクライアントプログラムが非常に大きなメッセージを送信する場合は、3.5 倍より大きな倍数を設定することをお勧めします。



フォールトトレランス設定では、`server_timeout_server_connection` パラメータは、スイッチオーバー後のバックアップサーバーに何の影響も与えません。バックアップサーバーは、`ft_activation` パラメータで設定されているタイムアウトが経過した後にのみアクティブ化されます。

## server\_heartbeat\_client

```
server_heartbeat_client = interval
```

サーバーからクライアントへの接続では、このパラメータで指定した間隔（秒単位）でサーバーがすべてのクライアントに対してハートビートを送信します。

省略するか、ゼロを指定した場合のデフォルトは 5 秒です。



このパラメータは、リリース 4.4 の新機能です。この機能は、一方のエントリがリリース 4.0 より前のリリースのものである場合、無効になります。

## client\_timeout\_server\_connection

```
client_timeout_server_connection = limit
```

サーバからクライアントへの接続では、このパラメータで指定した期間が過ぎてもクライアントがハートビートを受け取らなかった場合、クライアントは接続を切断します。

この期間には、ハートビート間隔の約 3.5 倍の値を設定することをお勧めします。

ゼロは特別な値で、クライアントでのハートビート検出を無効にします（ただし、

サーバーはハートビートを送信し続けます)。



このパラメータは、リリース 4.4 の新機能です。この機能は、一方のエントリがリリース 4.0 より前のリリースのものである場合、無効になります。

## フォールトトレランスのパラメータ

これらのパラメータについての詳細は、473 ページの第 19 章、「フォールトトレランス」を参照してください。

先頭に `ft_ssl` のプリフィクスが付いたフォールトトレランスのパラメータは、フォールトトレランスサーバーのペア間のセキュアな接続を確立するために使用されます。このプロセスについての詳細は、488 ページの「SSL」を参照してください。

### ft\_active

`ft_active = URL`

アクティブなサーバーの URL を指定する。アクティブなサーバーにこのサーバーが接続できた場合、このサーバーはバックアップサーバーとして機能します。アクティブなサーバーにこのサーバーが接続できなかった場合、このサーバーがアクティブなサーバーになります。

### ft\_heartbeat

`ft_heartbeat = seconds`

稼働中であることを示すために、アクティブなサーバーがハートビート信号をバックアップサーバーに送信する間隔（秒単位）を指定します。デフォルトは 3 秒です。

### ft\_activation

`ft_activation = seconds`

アクティブなサーバーで障害が発生したことを示すアクティブ化間隔（ハートビート信号間の間隔の最大値）を指定します。秒単位で設定し、デフォルトは 10 秒です。この間隔には、ハートビート間隔の少なくとも 2 倍の値を設定する必要があります。

例：

`ft_activation = 60`



`ft_activation` パラメータはフォールトトレランスのスイッチオーバー処理後に、バックアップサーバーによってのみ使用されます。アクティブなサーバーは `server_timeout_server_connection` を使用して、障害が発生したサーバーを検出します。

**ft\_reconnect\_timeout**

```
ft_reconnect_timeout = seconds
```

バックアップサーバーが、フェイルオーバー状況にあるプライマリサーバーの役割を引き継いだ後、クライアントの再接続を待機する時間（秒単位）を指定します。クライアントが、この時間内に再接続しなかった場合、サーバーは共有ステートファイルからそのクライアントのステートを削除します。[ft\\_reconnect\\_timeout](#) 時間は、サーバーが共有ステートを完全に復元したときに開始します。従って、この値には、ストアファイルを復元するのに必要な時間は含まれていません。

このパラメータのデフォルト値は **60** です。

**ft\_ssl\_auth\_only**

```
ft_ssl_auth_only = enable | disable
```

**enabled** の場合、フォールトトレランスサーバーが認証時にのみ SSL の使用を要求することを許可します（認証時に SSL が必要なのはユーザーのパスワードを保護するためです）。この機能についての詳細は、470 ページの「[認証にのみ SSL を使用する](#)」を参照してください。

**disabled** に設定すると、サーバーはこの機能の要求を無視します。このパラメータを指定しなかった場合、デフォルト値は **disabled** です。

**ft\_ssl\_identity**

```
ft_ssl_identity = pathname
```

サポートされている形式の証明書を含むファイルへのパスを指定します。サポートされている形式は、PEM、DER、または PKCS#12 です。

このデジタル証明書のファイルの種類についての詳細は、457 ページの「[証明書と鍵のファイル名](#)」を参照してください。

**ft\_ssl\_issuer**

```
ft_ssl_issuer = chain_member
```

サーバーの証明書チェーンメンバーを指定します。CA ルート証明書を含む、チェーン全体を指定します。サーバーはチェーン内の証明書を、このパラメータで指定されている順番で読み取ります。

証明書は、PEM、DER、PKCS# 7、または PKCS#12 形式のいずれかでなければなりません。このデジタル証明書のファイルの種類についての詳細は、457 ページの「[証明書と鍵のファイル名](#)」を参照してください。

## ft\_ssl\_private\_key

```
ft_ssl_private_key = key
```

サーバーの秘密鍵を指定します。秘密鍵が `ft_ssl_identity` 内のデジタル証明書に含まれている場合、このパラメータは必要ありません。

このパラメータは、PEM、DER、および PKCS#12 形式の秘密鍵をサポートしています。

このパラメータに実際の鍵を指定することも、鍵が含まれているファイルへのパスを指定することもできます。このデジタル証明書のファイルの種類についての詳細は、457 ページの「[証明書と鍵のファイル名](#)」を参照してください。

## ft\_ssl\_password

```
ft_ssl_password = password
```

秘密鍵または秘密鍵のパスワードを指定します。

パスワードは、`tibemsadmin` ツールを使用して設定できます。このツールでパスワードを設定した場合、パスワードは設定ファイル内で暗号化されます。`tibemsadmin` を使ったパスワードの設定についての詳細は、117 ページの第 6 章、「[EMS 管理ツールの使用](#)」を参照してください。

## ft\_ssl\_trusted

```
ft_ssl_trusted = trusted_certificates
```

信頼する証明書のリストを指定します。このパラメータは、クライアント証明書の発行元として信頼する CA の証明書を設定します。

証明書は、PEM、DER、または PKCS#7 形式のいずれかである必要があります。実際の証明書を指定することも、証明書チェーンが含まれているファイルへのパスを指定することもできます。

このデジタル証明書のファイルの種類についての詳細は、457 ページの「[証明書と鍵のファイル名](#)」を参照してください。

## ft\_ssl\_rand\_egd

```
ft_ssl_rand_egd = pathname
```

EGD (Entropy Gathering Daemon) がインストールされている場合、その場所のパスを指定します。このデーモンは、EMS サーバーで使用される乱数の生成に使用されます。

## ft\_ssl\_verify\_host

```
ft_ssl_verify_host = enabled | disabled
```

フォールトトレランスサーバーに、別のサーバーの証明書を検証させるかどうかを指定する。このパラメータの値は、**enabled** または **disabled** です。デフォルトは **enabled** で、サーバーがもう一方のサーバーの証明書を検証する必要があることを意味します。

このパラメータが **disabled** に設定されている場合は、サーバーはもう一方のフォールトトレランスサーバーとのセキュアな接続を確立しますが、そのサーバーの ID 証明書は検証しません。

## ft\_ssl\_verify\_hostname

```
ft_ssl_verify_hostname = enabled | disabled
```

フォールトトレランスサーバーに、別のサーバーの証明書の CN フィールドが指定された名前であることを検証させるかどうかを指定する。このパラメータの値は、**enabled** および **disabled** です。デフォルトは **enabled** で、フォールトトレランスサーバーが接続先のホストの名前、または **ft\_ssl\_expected\_hostname** パラメータに指定されている名前を、サーバーの証明書の値に照らし合わせて検証する必要があることを意味します。名前が一致しない場合、接続は拒否されます。

このパラメータが **disabled** に設定されている場合は、フォールトトレランスサーバーは他方のサーバーとのセキュアな接続を確立しますが、そのサーバーの身元は検証しません。

## ft\_ssl\_expected\_hostname

```
ft_ssl_expected_hostname = serverName
```

フォールトトレランスサーバーの CN フィールドにサーバーが指定することを予期される名前を指定する。このパラメータが設定されていない場合は、必要な名前は、サーバーのホスト名になります。

このパラメータは、**ft\_ssl\_verify\_hostname** パラメータが **enabled** に設定されている場合に使用されます。

## ft\_ssl\_ciphers

```
ft_ssl_ciphers = cipherSuite
```

サーバーによって使用される暗号スイートを指定します。複数のスイートを指定する場合、各スイートをコロン (:) で区切ります。このパラメータには、暗号スイート用の **OpenSSL** 名を使用することも、より長くて分かりやすい名前を使用することもできます。

EMS で利用可能な暗号スイート、および暗号スイートで使用する **OpenSSL** 名と長い名前についての詳細は、464 ページの「[暗号スイートの指定方法](#)」を参照してください。

## メッセージ追跡のパラメータ

### track\_message\_ids

```
track_message_ids = enabled | disabled
```

メッセージ ID でメッセージを追跡します。デフォルトは「disabled」です。

このパラメータを有効にした場合、管理ツールで `show message messageID` コマンドを使用することによって、メッセージを表示できます。

### track\_correlation\_ids

```
track_correlation_ids = enabled | disabled
```

相関 ID でメッセージを追跡します。デフォルトでは無効です。

このパラメータを有効にした場合、管理ツールで `show messages correlationID` コマンドを使用することによって、メッセージを表示できます。

## マルチキャストパラメータ

マルチキャストについての詳細は、357 ページの [第 13 章](#)、「[マルチキャストの使用方法](#)」を参照してください。

### multicast

```
multicast = enabled | disabled
```

EMS サーバーのマルチキャストを有効または無効にする。例：

```
multicast = enabled
```

デフォルトでは、この機能は無効です。

### multicast\_channels

```
multicast_channels = file
```

マルチキャストチャンネルが定義されている設定ファイルを指定する。

例：

```
multicast_channels = mychannels.conf
```

このパラメータを指定しなかった場合、EMS サーバーは `channels.conf` ファイル内のチャンネル定義を検索します。

## multicast\_daemon\_default

```
multicast_daemon_default = tcp-port
```

EMS クライアントがマルチキャスト デーモン への接続に使用する TCP ポートを指定します。例：

```
multicast_daemon_default = 9999
```

このパラメータは、EMS クライアントがマルチキャスト デーモン への接続に使用する TCP ポートを決定します。すべてのクライアントを一元的に設定するため、サーバーにこのパラメータが提供されます。EMS クライアントの動作を決定しますが、マルチキャスト デーモン には影響しません。マルチキャスト デーモン は、クライアントが接続に使用するポートと同じポートでクライアントをリスンする必要があります。マルチキャスト デーモン が `multicast_daemon_default` で指定されたポートと同じポートをリスンしていない場合、クライアントはデーモンに接続できず、エラーが発生します。

マルチキャスト デーモン がリスンする TCP ポートを変更するには、デーモンでコマンドラインの引数 `-listen` を使用します。詳細は、364 ページの「[コマンドラインオプション](#)」を参照してください。

このパラメータを指定しなかった場合、デフォルトのポートは 7444 です。

## multicast\_statistics\_interval

```
multicast_statistics_interval = seconds
```

マルチキャストの統計情報が各チャネルの監視トピック

`$sys.monitor.multicast.stats` にパブリッシュされる間隔を秒単位で指定します。5 秒より短い間隔はサポートされていません。

例：

```
multicast_statistics_interval = 90
```

マルチキャストの統計情報を無効にするには、`multicast_statistics_interval` を 0（ゼロ）に設定します。

このパラメータを指定しなかった場合、デフォルト値は 0（無効）です。



## TIBCO Rendezvous パラメータ

389 ページの第 15 章、「TIBCO Rendezvous との相互運用」も参照してください。

### tibrv\_transports

`tibrv_transports = enabled | disabled`

`transports.conf` で定義されている TIBCO Rendezvous トランスポートを有効または無効にするかを指定します。

明示的にこのパラメータを `enabled` に設定しない限り、デフォルト値は `disabled` です。したがって、すべてのトランスポートは無効であり、外部のシステムとメッセージを送受信することはできません。

## TIBCO SmartSockets パラメータ

413 ページの第 16 章、「TIBCO SmartSockets との相互運用」も参照してください。

### module\_path

`module_path = SmartSockets-shared-library-directory`

`SmartSockets-shared-library-directory` には、SmartSockets のライブラリファイルが含まれるディレクトリの絶対パスを指定します。例：

`module_path = c:\tibco\ss\bin\i86_w32`

該当する共有ライブラリがインストールされているディレクトリについては、SmartSockets のマニュアルを参照してください。



`module_path` パラメータは、AIX プラットフォームのインストールで IBM JVM を読み込むためにも使用されます。libjvm.so とその依存ライブラリが含まれるディレクトリを指定します。

複数のディレクトリを指定できます (SmartSockets と JVM 両方のライブラリを読み込む場合など)。パスを区切るには、UNIX プラットフォームではコロン (:)、Windows プラットフォームではセミコロン (;) を使用します。

### tibss\_transports

`tibss_transports = enabled | disabled`

`transports.conf` で定義されている TIBCO SmartSockets トランスポートを有効または無効にするかを指定します。

明示的にこのパラメータを `enabled` に設定しない限り、デフォルト値は `disabled` です。したがって、すべてのトランスポートは無効であり、外部のシステムとメッセージを送受信することはできません。

## tibss\_config\_dir

```
tibss_config_dir = pathname
```

SmartSockets 設定ファイルとメッセージファイルのディレクトリを指定します。

- `tal_ss.cat` は、メッセージに必要なファイル。このファイルが存在しない場合、`tibemsd` は警告メッセージを出力する。
- `tibems_ss.cm` は、SmartSockets RTclient 設定オプションのオプションファイル。

このパラメータが指定されていない場合、`tibemsd` は、これらのファイルを求めて現在の作業ディレクトリを検索します。

これらのファイルについての詳細は、『TIBCO SmartSockets User's Guide』を参照してください。

## ログファイルパラメータのトレース

これらのパラメータについての詳細は、433 ページの [第 17 章、「サーバーアクティビティの監視」](#) を参照してください。

## logfile

```
logfile = pathname
```

サーバーログファイルの名前と場所を指定する。

*pathname* に空白が含まれる場合は、二重引用符で囲む必要があります。

## log\_trace

```
log_trace = traceOptions
```

`logfile` パラメータによって定義されたファイルのトレースの基本設定を設定します。`logfile` が設定されていない場合、ここで指定した値は反映されません。

このパラメータの値には、トレースオプションをコンマで区切って指定します。トレースオプションのリストとその意味については、436 ページの [表 72、「サーバーのトレースオプション」](#) を参照してください。

トレースオプションは、次の三つの形式で指定できます。

- 符号なし：プリフィクス文字が指定されていないトレースオプション。既存のすべてのトレースオプションを置換する。
- `+:+` で始まるトレースオプション。指定したオプションを現在のトレースオプションセットに追加する。
- `-:-` で始まるトレースオプション。指定したオプションを、現在のトレースオプションセットから削除する。

次のコード例では、アクセス制御違反に関するメッセージのみを表示するようにトレースログを設定します。

```
log_trace=ACL
```

次のコード例では、SSL メッセージに加えて、すべてのデフォルトのトレースメッセージを表示するように設定します。ただし、ADMIN メッセージを除きます。

```
log_trace=DEFAULT,-ADMIN,+SSL
```

## logfile\_max\_size

```
logfile_max_size = size [KB|MB|GB]
```

ログファイルのローテーションが行われる前に、推奨の最大ログファイルサイズを指定します。ゼロを指定すると、制限は設定されません。単位には、KB、MB、またはGBを使用します（単位を指定しない場合、ファイルサイズはバイト単位で扱われます）。

サーバーは、現在のログファイルのサイズを定期的に確認します。指定のサイズよりも大きくなると、ログファイルは、バックアップコピーが作成された後、空にされます。その後、サーバーは空のログファイルへの書き込みを開始し、指定されたサイズに再び達するまで、書き込みを続けます。

バックアップログファイルは、順番に名前が付けられ、現在のログと同じディレクトリに格納されます。

## console\_trace

```
console_trace = traceOptions
```

stderr に出力するトレースオプションを設定します。これらのパラメータに指定可能な値は、log\_trace の場合と同じです。ただし、コンソールのトレースは、ログファイルとは独立して行われます。

logfile が定義されている場合は、次のように指定することによって、コンソール出力を停止できます。

```
console_trace=-DEFAULT
```

重要なエラーメッセージ（およびその他一部メッセージ）は、このトレース設定より優先され、常に出力されます。

次のコード例では、TIBCO Rendezvous アドバイザリメッセージを受け取ったときに、トレースメッセージをコンソールに送信します。

```
console_trace=RVADV
```

## client\_trace

```
client_trace = {enabled|disabled} [target=location]
               [user|connid|clientid=value]
```

管理者は、接続または接続のグループをトレースすることができます。このプロパティが enabled の場合、サーバーは各クライアントに、接続のオープンとクローズ、メッセージアクティビティ、およびトランザクションアクティビティのトレース出力を生成するように指示します。このタイプのトレースでは、クライアントプログラムを再起動する必要はありません。

各クライアントは、*location* にトレース出力を送信します。*location* は、`stderr`（デフォルト）または `stdout` です。



また、C、Java、および .NET ライブラリでそれぞれ `tibems_SetTraceFile`、`Tibjms.setTraceFile`、および `Tibems.SetTraceFile` を使用することで、クライアントによるトレースをファイルに出力することもできます。

デフォルトの動作では、すべての接続がトレースされます。`user`、`connid` または `clientid` を指定し、特定の接続を選択してトレースすることもできます。*value* には、必要に応じて、ユーザー名または ID を指定できます。

管理ツールを使用してこのパラメータを設定した場合、設定ファイル `tibemsd.conf` 内の値は変更されません。したがって、サーバーの再起動後、この値は存続しません。存続させるには、設定ファイルに値を設定する必要があります。

### trace\_client\_host

```
trace_client_host = [hostname|address|both|both_with_port]
```

接続に関連するトレースステートメントは、ホストをホスト名、IP アドレス、またはその両方により識別できます。このパラメータを指定しなかった場合、デフォルトは `hostname` です。`both_with_port` オプションでは、IP アドレスとホスト名に加えて、ホストで使用される一時的なポートが表示されます。

## 統計情報収集のパラメータ

これらのパラメータについての詳細は、433 ページの [第 17 章、「サーバーアクティビティの監視」](#) を参照してください。

### server\_rate\_interval

```
server_rate_interval = seconds
```

サーバーの全般統計情報の平均値を求める間隔（秒単位）を設定します。このパラメータには、ゼロより大きい任意の正の整数を設定できます。

サーバーの全般統計情報は常に収集されるため、このパラメータにゼロを設定することはできません。デフォルトではこのパラメータは 1 に設定されています。

このパラメータを設定すると、指定した間隔で、メッセージの送信率とメッセージサイズの平均値が得られます。

## statistics

`statistics = enabled | disabled`

プロデューサ、コンシューマ、デスティネーション、およびルートを集める統計情報を有効または無効にします。デフォルトではこのパラメータは **disabled** に設定されています。

統計情報の収集を無効にすると、各オブジェクトの統計情報で合計値がゼロにリセットされます。

## rate\_interval

`rate_interval = seconds`

ルート、デスティネーション、プロデューサ、コンシューマの統計情報の平均値を求める間隔（秒単位）を設定します。デフォルトではこのパラメータは 3 秒に設定されています。このパラメータにゼロを設定すると、平均計算が無効になります。

## detailed\_statistics

`detailed_statistics = NONE | [PRODUCERS, CONSUMERS, ROUTES, CHANNELS]`

オブジェクトに対して詳細統計情報の追跡を行うかどうかを指定します。詳細統計情報の追跡は、ルート、チャネル、デスティネーションを指定しないプロデューサ、ワイルドカードのデスティネーションを指定するコンシューマにのみ適しています。詳細統計情報の追跡が有効になっている場合、各デスティネーションの統計情報がそのオブジェクトのために記録されます。

このパラメータに **NONE** を設定すると、詳細統計情報の追跡が無効になります。各オブジェクトのトラックを有効にする場合、**PRODUCERS**、**CONSUMERS**、**ROUTES**、または **CHANNELS** を任意の組み合わせで指定できます。複数の種類の詳細追跡を指定する場合、各アイテムをコンマで区切ります。

例：

`detailed_statistics = NONE`

詳細統計情報の追跡をオフにします。

`detailed_statistics = PRODUCERS, ROUTES`

プロデューサとルートの詳細統計情報を収集するように指定します。

## statistics\_cleanup\_interval

`statistics_cleanup_interval = seconds`

デスティネーションにアクティビティがない場合、サーバーが詳細統計情報を保管する期間（秒単位）を指定します。このパラメータは、詳細統計情報の追跡に使用されるメモリの最大容量を制御する場合に役立ちます。指定された間隔に達すると、アクティビティのないデスティネーションの統計情報は削除されます。

## max\_stat\_memory

```
max_stat_memory = size [KB|MB|GB]
```

詳細統計情報の収集に使用するメモリの最大容量を指定します。単位を指定しない場合、ファイルサイズの単位にはバイトが使用されます。単位を指定する場合は、KB、MB、または GB を使用できます。

最大メモリ制限に達すると、サーバーは詳細統計情報の収集を中止します。統計情報が削除され、メモリが利用可能になると、サーバーは詳細統計情報の収集を再開します。

## SSL サーバーのパラメータ

これらのパラメータについての詳細は、453 ページの [第 18 章、「SSL プロトコルの使用」](#) を参照してください。

### ssl\_dh\_size

```
ssl_dh_size = [512 | 768 | 1024 | 2048]
```

Diffie-Hellman キーのサイズを指定します。512、768、1024、または 2048 ビットを指定できます。デフォルト値は 1024 です。

この鍵は、エクスポート可能な暗号スイートには使用されません。

### ssl\_server\_ciphers

```
ssl_server_ciphers = cipherSuites
```

サーバーによって使用される暗号スイートを指定します。複数のスイートを指定する場合、各スイートをコロン (:) で区切ります。このパラメータは、OpenSSL 暗号文字列構文に従う必要があります。

たとえば、次の設定を使って、2 種類の暗号スイートを有効にできます。

```
ssl_server_ciphers = RC4-MD5:RC4-SHA
```

EMS で利用可能な暗号化スイート、およびこのパラメータで暗号スイートを指定する場合の構文についての詳細は、464 ページの [「暗号スイートの指定方法」](#) を参照してください。

### ssl\_require\_client\_cert

```
ssl_require_client_cert = enable | disable
```

このパラメータが **enable** に設定されている場合、サーバーは、デジタル証明書を持つクライアントからの SSL 接続だけを受け入れます。証明書のないクライアントからの接続は拒否されます。

このパラメータが **disable** に設定されている場合は、デジタル証明書を持たないクライアントからの接続も受け入れられます。

このパラメータが **enable** と **disable** のどちらに設定されているかにかかわらず、デジタル証明書を持つクライアントは常に、**ssl\_server\_trusted** パラメータに指定されている証明書のリストと照らし合わせて認証されます。

## ssl\_use\_cert\_username

**ssl\_use\_cert\_username** = **enable** | **disable**

このパラメータが **enable** に設定されている場合は、デジタル証明書が指定されていれば、クライアントのユーザー名はクライアントのデジタル証明書の CN フィールドから常に抽出されます。接続ファクトリまたは API 呼び出しで異なるユーザー名が指定された場合、そのユーザー名は破棄されます。CN からのユーザー名のみが使用されます。

CN フィールドには、ユーザー名、電子メールアドレス、または Web アドレスのいずれかが指定されています。



**ssl\_use\_cert\_username** が有効な場合、CN によって指定されたユーザー名が唯一の有効なユーザー名になります。API 呼び出しに関連付けられたアクセス権限など、異なるユーザー名に関連付けられた権限は無視されます。

## ssl\_cert\_user\_specname

**ssl\_cert\_user\_specname** = *username*

このパラメータは、ユーザー名を指定するようにクライアントで求められたときに、特別なユーザー名を指定することで、クライアントのデジタル証明書からクライアントのユーザー名を取得するようにする必要があるときに便利です。

たとえば、ログイン時に、すべてのクライアントにユーザー名を指定させるように設定するとします。この場合は、**ssl\_use\_cert\_username** パラメータを **disable** に設定します。**username** はユーザーによって入力され、デジタル証明書からは取得されません。一方、あるユーザー名について、そのユーザー名でログインするクライアントに証明書から名前を取得させるように設定することもできます。この方法の代表的な使用例は、**anonymous** です。**anonymous** としてログインするすべてのクライアントは、クライアント自身のデジタル証明書からユーザー名を取得します。

このパラメータによって指定される値は、ユーザー名をデジタル証明書から取得するときに、クライアントがログインに使用するユーザー名です。このパラメータの代表的な使用例は、**anonymous** です。

また、このパラメータの値は、**ssl\_use\_cert\_username** が **enable** に設定されている場合は無視されます。このような場合には、すべてのクライアントユーザー名が証明書から取得されるためです。このパラメータは、証明書を持たないユーザーには影響しません。

## ssl\_server\_identity

```
ssl_server_identity = certificate
```

PEM、DER、または PKCS#12 形式のサーバーのデジタル証明書を指定します。サポートされている形式の証明書を含むファイルへのパスを指定することができます。

SSL ポートが `listen` パラメータにリストされている場合、このパラメータを指定する必要があります。

PEM と PKCS#12 形式では、デジタル証明書に秘密鍵を含めることができます。これらの形式を使用し、デジタル証明書に秘密鍵が含まれている場合は、`ssl_server_key` の設定はオプションです。

例：

```
ssl_server_identity = certs/server.cert.pem
```

## ssl\_server\_key

```
ssl_server_key = private_key
```

サーバーの秘密鍵を指定します。秘密鍵が `ssl_server_identity` 内のデジタル証明書に含まれている場合、このパラメータは必要ありません。

このパラメータは、PEM、DER、および PKCS#12 形式の秘密鍵をサポートしています。

秘密鍵が含まれているファイルへのパスを指定する必要があります。

## ssl\_password

```
ssl_password = password
```

秘密鍵または秘密鍵のパスワードを指定します。

このパスワードはオプションで、`tibemsd` の起動時にコマンドラインで指定できます。

SSL が有効になっている場合、このパラメータまたはコマンドラインを使ってパスワードが指定されていないと、`tibemsd` は起動時にパスワードの入力を求めます。

パスワードは、`tibemsadmin` ツールを使用して設定できます。このツールでパスワードを設定した場合、パスワードは設定ファイル内で暗号化されます。`tibemsadmin` を使ったパスワードの設定についての詳細は、117 ページの [第 6 章、「EMS 管理ツールの使用」](#) を参照してください。



接続ファクトリには（セキュリティ上の理由により）`ssl_password` が含まれないため、EMS サーバーはユーザー認証用の「接続の作成」呼び出しで指定されたパスワードを使用します。接続の作成パスワードが `ssl_password` と異なる場合、接続の作成は失敗します。



## ssl\_server\_issuer

```
ssl_server_issuer = chain_member
```

サーバーの証明書チェーンメンバーを指定します。サーバーはチェーン内の証明書を、このパラメータで指定されている順番で読み取ります。

同じ証明書が証明書チェーン内の複数の箇所に登場する場合があります。

証明書は、PEM、DER、PKCS# 7、または PKCS#12 形式のいずれかでなければなりません。

このデジタル証明書のファイルの種類についての詳細は、457 ページの「[証明書と鍵のファイル名](#)」を参照してください。

## ssl\_server\_trusted

```
ssl_server_trusted = certificates
```

クライアント証明書の発行者として、サーバーが信頼する CA ルート証明書のリストを指定します。

CA ルート証明書を指定します。中間 CA 証明書は指定できません。

証明書は、PEM、DER、または PKCS#7 形式のいずれかである必要があります。実際の証明書を指定することも、証明書チェーンが含まれているファイルへのパスを指定することもできます。

例：

```
ssl_server_trusted = certs¥CA1_root.pem  
ssl_server_trusted = certs¥CA2_root.pem
```

このデジタル証明書のファイルの種類についての詳細は、457 ページの「[証明書と鍵のファイル名](#)」を参照してください。

## ssl\_rand\_egd

```
ssl_rand_egd = pathname
```

EGD (Entropy Gathering Daemon) がインストールされている場合、その場所のパスを指定します。このデーモンは、C クライアントと EMS サーバーで使用される乱数の生成に使用されます。Java クライアントはこのパラメータを使用しません。

## ssl\_crl\_path

```
ssl_crl_path = pathname
```

サーバーの失効した証明書のリスト (CRL: Certificate Revocation List) をアクティブ化するには、このパラメータに NULL 値以外の値を指定します。

サーバーは、このディレクトリから CRL ファイルを読み取ります。ディレクトリには CRL ファイルのみが含まれている必要があります。**pathname** ディレクトリ内に他のファイルがある場合、SSL の初期化は失敗します。

## ssl\_crl\_update\_interval

```
ssl_crl_update_interval = hours
```

サーバーが CRL を自動更新する間隔（時間単位）を指定します。

このパラメータを指定しなかった場合、デフォルトは 24 時間です。

## ssl\_auth\_only

```
ssl_auth_only = enable | disable
```

**enabled** の場合、クライアントが認証時にのみ SSL の使用を要求することを許可します（認証時に SSL が必要なのはユーザーのパスワードを保護するためです）。この機能についての詳細は、470 ページの「[認証にのみ SSL を使用する](#)」を参照してください。

**disabled** に設定すると、サーバーはこの機能を要求するクライアントを無視します。このパラメータを指定しなかった場合、デフォルト値は **disabled** です。

## fips140-2

```
fips140-2 = true | false
```

**true** の場合、EMS サーバーを FIPS 140-2 準拠モードで実行できます。**false** の場合、またはこのパラメータを指定しなかった場合、サーバーは FIPS に準拠しません。詳細は、471 ページの「[FIPS 準拠の有効化](#)」を参照してください。

## LDAP パラメータ

これらのパラメータについての詳細は、255 ページの第 8 章、「[認証とアクセス権限](#)」を参照してください。

### ldap\_url

外部のディレクトリサーバーの URL を指定します。以下の形式を使用できます。

```
LDAP://host:tcp_port
```

または

```
LDAPS://host:ssl_port
```

例：

```
LDAP://myLdapServer:1855
```

## ldap\_principal

`ldap_principal = DN`

EMS サーバーが LDAP サーバーへのバインドに使用する LDAP ユーザーの識別名 (DN: Distinguished Name) を指定します。このユーザーは、グループユーザーをバインドおよび参照できる権限を持っている必要がありますが、管理権限を持っている必要はありません。

例 :

`ldap_principal = "cn=Manager"`

## ldap\_credential

`ldap_credential = password`

`ldap_principal` プロパティに定義されているユーザーに関連付けられているパスワードを指定します。値を指定する必要がある、空の文字列にすることはできません。

## ldap\_cache\_enabled

`ldap_cache_enabled = enable | disable`

LDAP データのキャッシュを有効にします。

## ldap\_cache\_ttl

`ldap_cache_ttl = seconds`

キャッシュされた LDAP データが更新される前に、保持される最大時間 (秒単位) を指定します。

## ldap\_conn\_type

`ldap_conn_type = [ldaps | startTLS]`

サーバーによる LDAP 情報の取得に使用される接続の種類を指定する。

- このパラメータが指定されていない場合、LDAP 接続は TCP (セキュアでない) を使用する。下位互換性を維持する目的で、デフォルトでこのように設定されている。
- `ldaps` : LDAP 接続に SSL を使用する (セキュア)。
- `startTLS` : LDAP プロトコルのバージョン 3 で導入された `startTLS` を使用する (セキュア)。

## ldap\_tls\_cacert\_file

```
ldap_tls_cacert_file = pathname
```

このファイルには、EMS サーバーが LDAP サーバーの証明書に署名するときに信頼する CA 証明書が含まれています。

セキュアな接続を確立するには、`ldap_tls_cacert_file` を指定する必要があります。オプションで、`ldap_tls_cacert_file` と同時に `ldap_tls_cacert_dir` を使用することで、個々の CA 証明書が追加されたディレクトリを指定できます。

## ldap\_tls\_cacert\_dir

```
ldap_tls_cacert_dir = pathname
```

確認チェーンにふたつ以上の CA 証明書がある場合、サーバーはこのディレクトリをスキャンして CA 証明書を検索します。

セキュアな接続を確立するには、`ldap_tls_cacert_file` も指定する必要があります。`ldap_tls_cacert_dir` はオプションのパラメータで、`ldap_tls_cacert_file` と同時に使用することで、個々の CA 証明書が追加されたディレクトリを指定できます。

## ldap\_tls\_cipher\_suite

```
ldap_tls_cipher_suite = cipher_suite
```

オプション。セキュアな LDAP 接続の暗号化に使用される暗号スイートを指定できます。

このパラメータは、OpenSSL 暗号文字列構文に従う必要があります。詳細は、464 ページの「[暗号スイートの指定方法](#)」を参照してください。スイートの指定には短い名前を使用する必要があります。たとえば、`SSL_RSA_WITH_DES_CBC_SHA` ではなく `DES-CBC-SHA` を使用します。長い名前を使用すると、クライアントへの接続時に認可エラーが発生します。

実際の暗号化名に加えて、暗号化の品質を指定することもできます。たとえば、以下のように指定します。

- HIGH
- HIGH:MEDIUM

## ldap\_tls\_rand\_file

```
ldap_tls_rand_file = pathname
```

オペレーティングシステムにランダムデータ機能が搭載されていない場合、このファイルが暗号化用のランダムデータのソースとなります。

**ldap\_tls\_cert\_file**

```
ldap_tls_cert_file = pathname
```

LDAP サーバーがクライアント認証を必要とする場合、このファイル内の証明書を使用して EMS サーバーを識別します。

**ldap\_tls\_key\_file**

```
ldap_tls_key_file = pathname
```

LDAP サーバーがクライアント認証を必要とする場合、このファイル内の秘密鍵を使用します。

サーバーをリモートで起動する場合は、この秘密鍵ファイルをパスワード暗号化しないことをお勧めします。

これらのパラメータについての詳細は、255 ページの [第 8 章、「認証とアクセス権限」](#) を参照してください。

**ldap\_user\_class**

```
ldap_user_class = class_name
```

ユーザーを格納する LDAP オブジェクトクラスの名前を指定します。

例：

```
ldap_user_class = person
```

**ldap\_user\_attribute**

```
ldap_user_attribute = attribute
```

ユーザーオブジェクトクラスにおいて、ユーザーの名前を保持している属性の名前を指定します。

例：

```
ldap_user_attribute = uid
```

**ldap\_user\_base\_dn**

```
ldap_user_base_dn = DN
```

ユーザーを格納している LDAP ツリーのベース識別名 (DN : Distinguished Name) を指定します。

例：

```
ldap_user_base_dn = "ou=People,dc=Corp"
```

## ldap\_user\_scope

```
ldap_user_scope = onelevel | subtree
```

ベース DN からどのくらい下位の階層までユーザーを検索するかを指定する。このパラメータには、**onelevel** および **subtree** を指定できます。**onelevel** を指定すると、DN より 1 レベル下の階層のみが検索されます。**subtree** を指定すると、すべてのサブツリーが検索されます。

例：

```
ldap_user_scope = subtree
```

## ldap\_user\_filter

```
ldap_user_filter = filter
```

指定されたユーザー名の検索に使用する LDAP 検索フィルタを指定します。**%s** は、フィルタ中でユーザー名のプレースホルダとして使用します。例：

```
uid=%s
```

LDAP 検索の完全な構文は、RFC 2254 と RFC 2251 で定義されています。

このパラメータを指定しない場合、デフォルトの検索フィルタはユーザーオブジェクトクラスとユーザー名属性に基づいて生成されます。

## ldap\_all\_users\_filter

```
ldap_all_users_filter = filter
```

特定のユーザーベース DN を持つすべてのユーザーの検索に使用する、オプションの LDAP 検索フィルタを指定します。

このパラメータを指定しない場合、デフォルトの検索フィルタはユーザーオブジェクトクラスとユーザー名属性に基づいて生成されます。

これらのパラメータについての詳細は、255 ページの [第 8 章、「認証とアクセス権限」](#) を参照してください。

## ldap\_group\_base\_dn

```
ldap_group_base_dn = DN
```

グループを格納している LDAP ツリーのベース識別名 (DN : Distinguished Name) を指定します。

例：

```
ldap_group_base_dn = "ou=Groups,dc=Corp"
```

## ldap\_group\_scope

```
ldap_group_scope = onelevel | subtree
```

ベース DN からどのくらい下位の階層までグループを検索するかを指定する。このパラメータには、**onelevel** および **subtree** を指定できます。**onelevel** を指定

すると、DN より 1 レベル下の階層のみが検索されます。`subtree` を指定すると、すべてのサブツリーが検索されます。

例：

```
ldap_group_scope = subtree
```

## ldap\_group\_filter

```
ldap_group_filter = filter
```

指定されたグループ名を持つグループの検索に使用するオプションの LDAP 検索フィルタを指定します。`%s` は、フィルタ中でグループ名のプレースホルダとして使用します。

LDAP 検索の完全な構文は、RFC 2254 と RFC 2251 で定義されています。

このパラメータを指定しない場合、デフォルトの検索フィルタはグループオブジェクトクラスとグループ属性に基づいて生成されます。

例：

```
ldap_group_filter =
"(|(&(cn=%s)(objectClass=groupofUniqueNames))(&(cn=%s)
(objectClass=groupOfURLs)))"
```

## ldap\_all\_groups\_filter

```
ldap_all_groups_filter = filter
```

特定のグループベース DN を持つすべてのグループの検索に使用される、オプションの LDAP 検索フィルタを指定します。

このパラメータを指定しない場合、デフォルトの検索フィルタはグループオブジェクトクラスとグループ属性に基づいて生成されます。

## ldap\_static\_group\_class

```
ldap_static_group_class = name
```

静的グループを格納する LDAP オブジェクトクラスの名前を指定します。

例：

```
ldap_static_group_class = groupofuniquenames
```

## ldap\_static\_group\_attribute

```
ldap_static_group_attribute = class
```

グループの名前を保持する静的グループオブジェクトクラスの属性の名前を指定します。

例：

```
ldap_static_group_attribute = cn
```

**ldap\_static\_member\_attribute**

```
ldap_static_member_attribute = attribute
```

グループメンバーの識別名（DN：Distinguished Name）を指定する、LDAP 静的グループオブジェクトの属性を指定します。

例：

```
ldap_static_member_attribute = uniquemember
```

**ldap\_dynamic\_group\_class**

```
ldap_dynamic_group_class = class
```

動的グループを格納する LDAP オブジェクトクラスの名前を指定します。

例：

```
ldap_dynamic_group_class = groupofURLs
```

**ldap\_dynamic\_group\_attribute**

```
ldap_dynamic_group_attribute = attribute
```

グループの名前を保持する、動的グループオブジェクトクラスの属性の名前を指定します。例：

```
ldap_dynamic_group_attribute = cn
```

**ldap\_dynamic\_member\_url\_attribute**

```
ldap_dynamic_member_url_attribute = attribute
```

動的グループメンバーの URL を指定する、動的 LDAP グループオブジェクトの属性です。

例：

```
ldap_dynamic_member_url_attribute = memberURL
```

**拡張可能なセキュリティパラメータ**

拡張可能なセキュリティ機能では、サーバーの独自の認証およびアクセス権限モジュールを記述できます。この機能についての詳細は、279 ページの[第 9 章、「拡張可能なセキュリティ」](#)を参照してください。



## jaas\_classpath

```
jaas_classpath = classpath
```

JAAS LoginModule によって使用される JAR ファイルと依存クラスを含める。このパラメータは、認証で拡張可能なセキュリティ機能を有効にするために必要です。

例：

```
jaas_classpath = ../usr/local/custom/user_jaas_plugin.jar
```

## jaas\_config\_file

```
jaas_config_file = file-name
```

JAAS 設定ファイルの場所を指定します。このファイルは、EMS サーバーでカスタム認証 LoginModule を実行するために使用されます。詳細は、284 ページの「[EMS サーバーでの LoginModule の読み込み](#)」を参照してください。

このパラメータは、認証で拡張可能なセキュリティ機能を有効にするために必要です。

例：

```
jaas_config_file = jaas.conf
```

## jaas\_login\_timeout

```
jaas_login_timeout = milliseconds
```

EMS サーバーが JAAS 認証モジュールの実行と応答を待機する時間をミリ秒単位で指定します。このタイムアウトは、サーバーが LoginModule にユーザー名とパスワードを渡すたびに使用されます。モジュールが応答を返さない場合、サーバーは認証を拒否します。

このパラメータはオプションです。このパラメータを指定しなかった場合、デフォルトのタイムアウトは 500 ミリ秒です。

例：

```
jaas_login_timeout = 250
```

## jaci\_classpath

```
jaci_classpath = classpath
```

JACI カスタムアクセス権限モジュールによって使用される JAR ファイルと依存クラスを含める。このパラメータは、アクセス権限の付与で拡張可能なセキュリティ機能を有効にするために必要です。

例：

```
jaci_classpath = ../usr/local/custom/user_jaci_plugin.jar
```

## jaci\_class

```
jaci_class = class-name
```

拡張可能なアクセス権限インタフェースを実装するクラスの名前を指定する。このクラスは、Java アクセス制御インタフェース (JACI: Java Access Control Interface) を使用して記述する必要があります。JACI を使用してアクセス権限を付与するカスタムアプリケーションを記述する方法についての詳細は、290 ページの「[アクセス権限モジュールの作成](#)」を参照してください。

例：

```
jaci_class = com.userco.auth.CustomAuthorizer
```

## jaci\_timeout

```
jaci_timeout = milliseconds
```

EMS サーバーが JACI アクセス権限モジュールの実行と応答を待機する時間をミリ秒単位で指定します。このタイムアウトは、サーバーがアクセス権限モジュールにデスティネーション、ユーザー名、およびアクションを渡すたびに使用されます。モジュールが応答を返さない場合、サーバーは認可を拒否します。

このパラメータはオプションです。このパラメータを指定しなかった場合、デフォルトのタイムアウトは 500 ミリ秒です。

例：

```
jaci_timeout = 250
```

## JVM パラメータ

これらのパラメータは、EMS サーバーで Java 仮想マシン (JVM: Java Virtual Machine) を有効にして設定します。EMS での JVM の動作についての詳細は、292 ページの「[JVM の有効化](#)」を参照してください。

## jre\_library

```
jre_library = path
```

EMS サーバーで JVM を有効にします。*path* には、JRE と同時にインストールされた *jvm.dll* 共有ライブラリファイルの絶対パスを指定します。このパラメータを指定しなかった場合、JVM はデフォルトで無効になります。

*path* に空白が含まれる場合は、パスを引用符で囲む必要があります。

例：

```
jre_library = "C:¥Program Files¥Java¥jdk1.6.0_04¥jre¥bin¥server¥jvm.dll"
```

## jre\_option

```
jre_option = JVMoption
```

起動時に JVM にコマンドラインオプションを渡す。jre\_option パラメータは、Java システムプロパティの定義に使用できます。Java システムプロパティは、拡張可能なセキュリティモジュールなどの JVM で実行するアプリケーションで使用されます。

複数の jre\_option エントリを使用して、JVM に複数のオプションを渡すことができます。JVMoption の値には、Sun Microsystems によって定義されたほとんどの JVM オプションを指定できます。

たとえば、次の設定では JVM の最大ヒープサイズを 256 MB に制限します。

```
jre_option = -Xmx256m
```

## その他の設定ファイルの使用

主要設定ファイル以外にも、各種の設定ファイルがあり、多様な用途に使用できます。

表 32 設定ファイル

設定ファイル	説明	参照 ページ
<code>acl.conf</code>	EMS アクセス制御リストを定義する。	229
<code>bridges.conf</code>	デスティネーション間のブリッジを定義する。	230
<code>channels.conf</code>	マルチキャストメッセージがブロードキャストされるマルチキャストチャンネルを定義する。	231
<code>durables.conf</code>	静的持続サブスクライバを定義する。	234
<code>factories.conf</code>	EMS サーバーに JNDI 名として格納されている、接続ファクトリを定義する。	235
<code>groups.conf</code>	EMS グループを定義する。	239
<code>jaas.conf</code>	LoginModule を探して読み込む。	240
<code>queues.conf</code>	EMS キューを定義する。	240
<code>routes.conf</code>	この EMS サーバーと別の EMS サーバー間のルートを定義する。	241
<code>stores.conf</code>	EMS サーバーでメッセージを保存する、ストアファイルまたはデータベースの場所を定義する。	243
<code>tibrvcn.conf</code>	<code>tibrvcn</code> トランSPORTにメッセージをEXPORTするトピックによって使用される、TIBCO Rendezvous 保証メッセージ (RVCN: Rendezvous Certified Messaging) リスナーを定義する。	247
<code>topics.conf</code>	EMS トピックを定義する。	247
<code>transports.conf</code>	Rendezvous や SmartSockets などの外部のメッセージサービスに対してメッセージのインポートとEXPORTを行う際に EMS が使用するトランSPORTを定義する。	248
<code>users.conf</code>	EMS ユーザーを定義する。	252

これらの設定ファイルは手作業で編集できますが、一部のファイルは管理ツールまたは管理 API を使って変更することもできます。管理ツールの使用についての詳細は、117 ページの第 6 章、「[EMS 管理ツールの使用](#)」を参照してください。

以降のセクションでは、各設定ファイルについて説明します。

## acl.conf

このファイルは、すべてのユーザーとグループについて、トピックとキューに対するすべてのアクセス権限を定義します。ファイルは、以下の形式で記述します。

```
TOPIC=topic USER=user PERM=permissions
TOPIC=topic GROUP=group PERM=permissions
QUEUE=queue USER=user PERM=permissions
QUEUE=queue GROUP=group PERM=permissions
ADMIN USER=user PERM=permissions
ADMIN GROUP=group PERM=permissions
```

表 33 ACL パラメータ

パラメータ名	説明
TOPIC	アクセス権限を追加するトピックの名前。
QUEUE	アクセス権限を追加するキューの名前。
ADMIN	管理者のアクセス権限を追加するよう指定する。
USER	アクセス権限を追加するユーザーの名前。
GROUP	アクセス権限を追加するグループの名前。all というデスティネーションは、すべてのユーザーが含まれている定義済みのグループを指定します。
PERM	追加するアクセス権限。  キューに割り当て可能なアクセス権限は、send、receive、および browse です。トピックに割り当て可能なアクセス権限は、publish、subscribe、durable、および use_durable です。all と指定すると、すべての利用可能なアクセス権限が指定されます。このアクセス権限についての詳細は、276 ページの「 <a href="#">アクセス権限の確認</a> 」および 77 ページの「 <a href="#">アクセス権限の継承</a> 」を参照してください。  管理アクセス権限は、管理アクティビティを実行できるユーザーに付与されます。管理アクセス権限についての詳細は、257 ページの「 <a href="#">管理者のアクセス権限</a> 」を参照してください。

例

```
ADMIN USER=sys-admins PERM=all
TOPIC=foo USER=user2 PERM=publish,subscribe
TOPIC=foo GROUP=group1 PERM=subscribe
```

bridges.conf

このファイルは、デスティネーション間のブリッジを定義します。デスティネーションブリッジについての詳細は、78 ページの「[デスティネーションブリッジ](#)」を参照してください。

ファイルは、以下の形式で記述します。

```
[destinationType:destinationName] # mandatory -- include brackets
destinationType=destinationToBridgeTo1 [selector="msg-selector"]
destinationType=destinationToBridgeTo2 [selector="msg-selector"]
...
```

*destination-name* には、任意の特定のデスティネーションを指定するか、または複数のデスティネーションに一致するワイルドカードパターンを指定します。

表 34 ブリッジのパラメータ

パラメータ名	説明
<i>destinationType</i>	デスティネーションのタイプ。 <code>topic</code> または <code>queue</code> を指定します。
<i>destinationName</i>	デスティネーションの名前。
<i>destinationToBridgeTo</i>	ブリッジを作成するひとつ以上のデスティネーションの名前。
<i>selector</i>	オプションのプロパティ。ブリッジされたデスティネーションが受信するメッセージを制限するメッセージセクタを指定します。  メッセージセクタの構文についての詳細は、「説明」の「メッセージセクタ」セクションまたは、TIBCO Enterprise Message Service Java API リファレンスの <code>Message</code> クラスを参照してください。

例

```
[topic:myTopic1]
  topic=myTopic2
  queue=myQueue1
```

channels.conf

このファイルは、マルチキャスト対応のトピックにパブリッシュされたメッセージがブロードキャストされるマルチキャストチャネルを定義します。このファイルで定義された各チャネルには固有の名前が付けられ、異なるマルチキャストアドレス、マルチキャストポート、およびプロパティ値を指定できます。

ファイルは、以下の形式で記述します。

```
[multicast-channel-name]
  address = multicast-group-address : multicast-port
  [ttl = hops]
  [priority = priority]
  [maxrate = size [KB|MB|GB]]
  [maxtime = seconds]
  [interface = ip-address]
```

表 35 チャネルのパラメータ

パラメータ名	説明
[multicast-channel-name]	<p>[multicast-channel-name] は、このマルチキャストチャネルを識別する名前です。</p> <p>角かっこ [ ] は、multicast-channel-name がオプションであることを示しているわけではありません。名前は角かっこで囲まれている必要があります。</p>
address	<p>メッセージが送信される場所を指定します。</p> <ul style="list-style-type: none"><li>multicast-group-address には、メッセージが送信されるマルチキャストグループの IP アドレスを指定します。アドレスは 224.0.0.0 ～ 239.255.255.255 で指定する必要があります。</li><li>multicast-port には、メッセージが送信されるマルチキャストポートのデスティネーションを指定します。マルチキャストポートは 1 ～ 65535 で指定する必要があります。</li></ul> <p>たとえば、次の設定では、チャネルを介したメッセージは IP アドレスが 234.5.6.7 でマルチキャストポートが 99 の場所へ送信されます。</p> <p>address = 234.5.6.7:99</p>

表 35 チャンネルのパラメータ

パラメータ名	説明
<code>ttl</code>	<p>メッセージがサーバーとマルチキャスト デーモン 間で移動できる最大ホップ数を指定します。</p> <p>サーバーとマルチキャスト デーモン 間のホップ数は、その間にあるルーター数に 1 を加えた数です。たとえば、サーバーとマルチキャスト デーモン が同じサブネット内にある場合、その間のホップ数は 1 です。サーバーとマルチキャスト デーモン が 1 台のルーターで分離されている場合、その間のホップ数は 2 です。したがって、<code>ttl</code> の値が 1 の場合はマルチキャストデータがローカルサブネット内にとどまり、<code>ttl</code> の値が 2 の場合はメッセージが 1 台のルーターを経由して次のサブネットに移動できます。</p> <p>このパラメータを指定しなかった場合、デフォルトのネットワークホップの最大許容数は 16 です。</p>
<code>priority</code>	<p>帯域幅が割り当てられるときのチャンネルの送信優先度を指定します。<code>priority</code> は数値順位で指定します。最も高い優先度は -5 で、最も低い優先度は 5 です。</p> <p>このパラメータを指定しなかった場合、デフォルトの優先度は 0（ゼロ）です。</p>
<code>maxrate</code>	<p>メッセージのチャンネルを介した最大送信速度を指定します。単位としては、KB または MB を指定できます</p> <p>このパラメータを指定しなかった場合、デフォルト値は 12.5MB です。</p>



表 35 チャンネルのパラメータ

パラメータ名	説明
maxtime	<p>サーバーが送信済みメッセージを再送信用に保持する最大時間を秒単位で指定します。マルチキャストデーモンがメッセージの紛失を検出して EMS サーバーに否定応答を送信した場合、メッセージが再送信されます。</p> <p>maxtime を長くすると、サーバーでが使用されるメモリ量が増加します。チャンネルで使用されるメモリの最大容量は、<math>\text{maxrate} \times \text{maxtime}</math> です。たとえば、maxrate に 10MB を指定して maxtime に 10 秒を指定した場合、再送信するためにサーバーが 100 MB のデータをバッファする可能性があります。</p> <p>このパラメータを指定しなかった場合、メッセージは 35 秒間保持されます。</p>
interface	<p>サーバーがこのチャンネルでマルチキャストトラフィックを送信するために使用する IP アドレスを指定します。</p> <p>この IP アドレスは、マルチキャスト対応のインタフェースである必要があります。UNIX システムでは、ifconfig UNIX コマンドを実行することで、IP インタフェースがマルチキャスト対応であるかどうかを確認できます。</p> <p>このパラメータを指定しなかった場合、デフォルト値は 0.0.0.0 です。この場合、EMS サーバーはシステムのデフォルトインタフェースを使用します。</p>

## 例

```
[channel-1]
address=234.5.6.7:99
maxrate=10MB
maxtime=10
ttl=4
```

```
[channel-2]
address=234.5.3.9:99
maxrate=15MB
maxtime=10
ttl=3
```

durables.conf

このファイルは、静的持続サブスクライバを定義します。  
このファイルは、以下のいずれかの形式で記述された行により構成されます。

```
topic-name durable-name
[route]
[clientid=id]
[nolocal]
[selector="msg-selector"]
```

表 36 持続サブスクライバのパラメータ

パラメータ名	説明
topic-name	持続サブスクライバのトピック。
durable-name	持続サブスクライバの名前。
route	指定されている場合、サブスクライバは別のサーバーであり、durable-name はそのサーバーの名前です。  このプロパティを指定した場合、別のプロパティを指定することはできません。
clientid=id	サブスクライバ接続のためのクライアント ID。
nolocal	このパラメータを指定した場合、サブスクライバは、サブスクライバ自身の接続からパブリッシュされたメッセージを受信しません。
selector="string"	このパラメータを指定した場合、セレクトは、持続サブスクライバが受信するメッセージのセットを限定します。  メッセージセレクトの構文についての詳細は、「説明」の「メッセージセレクト」セクションまたは、TIBCO Enterprise Message Service Java API リファレンスの Message クラスを参照してください。

例

```
topic1 dName1
topic2 dName2 clientid=myId,nolocal
topic3 dName3 selector="urgency in ( ' high' , ' medium' )"
topic4 Paris route
```

- 指定の競合
- 持続サブスクライバ間の競合が検出された場合、サーバーは最初の指定を維持し、警告を出力します。次の例を参照してください。
- このファイル内の静的な指定は、クライアントによって動的に作成された新しい持続サブスクリプションよりも優先される。

- クライアントによって動的に作成された既存の持続サブスクリプションは、管理者によって新しく定義された静的な持続サブスクリプションより優先される。
- 静的な持続サブスクリバは、(同じトピックと持続名から) 動的なアンサブスクリバを試みるクライアントより優先される。

競合は、ワイルドカードによって生じることもあります。たとえば、クライアントがトピック `foo.*` の持続サブスクリバを動的に作成し、後から管理者がトピック `foo.1` の静的な持続サブスクリバを定義しようと試みる場合、サーバーはこの競合を検出し、管理者に警告します。

**設定**      このファイル内の持続サブスクリプションを設定するには、`tibemsadmin` ツールの `create durable` コマンドの使用をお勧めします。詳細は、[125 ページの「create durable」](#)を参照してください。

同じトピックと名前を持つ既存の動的な持続サブスクリプションが `create durable` コマンドによって検出された場合、そのサブスクリプションは静的サブスクリプションに昇格され、ファイル `durables.conf` に指定が書き込まれます。

factories.conf

このファイルは、内部 JNDI 名の接続ファクトリを定義します。  
このファイルは、以下の形式で記述されたファクトリ定義により構成されます。

```
[factory-name] # mandatory -- square brackets included
  type = generic|xageneric|topic|queue|xatopic|xqueue|
  url = url-string
  metric = connections | byte_rate
  clientID = client-id
  [connect_attempt_count|connect_attempt_delay|
  connect_attempt_timeout|reconnect_attempt_count|
  reconnect_attempt_delay|reconnect_attempt_timeout = value]
  [ssl-prop = value]*
```

表 37 接続ファクトリのパラメータ

パラメータ名	説明
<b>必須パラメータ</b>	
これらのパラメータは必須です。パラメータに指定された値は、API 呼び出しを使用して無効化することはできません。	
[factory-name]	[factory-name] は、接続ファクトリの名前です。  角かっこ [] は、factory-name がオプションであることを示しているわけではありません。名前は角かっこで囲まれている必要があります。

表 37 接続ファクトリのパラメータ

パラメータ名	説明
type	<p>接続ファクトリのタイプ。指定できる値は次のとおりです。</p> <ul style="list-style-type: none"><li>• generic：汎用的な接続（JMS 1.1）</li><li>• xageneric：汎用的な XA 接続（JMS 1.1）</li><li>• topic：トピック接続（JMS 1.0.2b）</li><li>• queue：キュー接続（JMS 1.0.2b）</li><li>• xatopic：XA トピック接続（JMS 1.0.2b）</li><li>• xaqueue：XA キュー接続（JMS 1.0.2b）</li></ul>
url	<p>この文字列はこのファクトリの接続の作成先のサーバーを指定します。</p> <ul style="list-style-type: none"><li>• 単一の URL は唯一のサーバーを指定する。例： tcp://host1:8222</li><li>• コンマで区切られた URL のペアは、フォールトトレランスサーバーのペアを指定する。例： tcp://host1:8222,tcp://backup1:8222</li><li>• 縦棒で区切られた URL のセットは、サーバー間で負荷分散を行うことを指定する。例： tcp://a:8222 tcp://b:8222 tcp://c:8222</li><li>• 負荷分散とフォールトトレランスは組み合わせることができる。例： tcp://a1:8222,tcp://a2:8222 tcp://b1:8222,tcp://b2:8222</li></ul> <p>この例では、2 台のサーバー（a と b）と、それらのサーバーのフォールトトレランスバックアップを定義しています。クライアントプログラムはプライマリ a サーバーとプライマリ b サーバーの負荷を確認し、負荷の少ない方に接続します。いずれかのプライマリサーバーに接続できない場合、クライアントはそのセカンダリサーバーへの接続を試みます。たとえば、b1 に接続できない場合は b2 に接続します。</p> <p>接続 URL は 1000 文字以下にする必要があります。</p> <p>注意事項については、239 ページの「<a href="#">負荷分散</a>」を参照してください。</p>

表 37 接続ファクトリのパラメータ

パラメータ名	説明
<b>オプションのパラメータ</b>	
これらのパラメータはオプションです。パラメータの値は、API 呼び出しを使用して無効化できます。	
<b>metric</b>	<p>ファクトリは、この測定基準を使用して、サーバーのグループ間の負荷を分散します。</p> <ul style="list-style-type: none"> <li>• <b>connections</b> : クライアント接続数が最も少ないサーバーに接続する。</li> <li>• <b>byte_rate</b> : バイト転送率が最も低いサーバーに接続する。バイト転送率は、受信データと送信データの両方が含まれている統計情報。</li> </ul> <p>このパラメータを指定しなかった場合、デフォルトの測定基準は <b>connections</b> です。</p> <p>注意事項については、239 ページの「<a href="#">負荷分散</a>」を参照してください。</p>
<b>clientID</b>	このクライアント ID 文字列は、ファクトリが作成する接続に関連付けられます。クライアント ID は 255 文字以下にする必要があります。
<b>connect_attempt_count</b>	クライアントプログラムは、EMS サーバーへの最初の接続が確立されるまで、サーバーへの接続を試みます（フォールトトレランス設定では、リスト内の URL を順に繰り返します）。このプロパティは、繰り返しの最大回数を決定します。このパラメータを指定しなかった場合、デフォルトでは 2 が設定されます。
<b>connect_attempt_delay</b>	サーバーへの最初の接続を試みる（フォールトトレランス設定では、リスト内の URL を繰り返す）ときに、クライアントは、次の試行を行う前にこのパラメータで指定した間隔（ミリ秒単位）でスリープします。このパラメータを指定しなかった場合、デフォルトでは 500 ミリ秒に設定されます。
<b>connect_attempt_timeout</b>	EMS サーバーへの接続を試みる場合、このパラメータで接続タイムアウト期間を設定し、指定した期間（ミリ秒）が経過した後で接続の試みを中断することができます。
<b>reconnect_attempt_count</b>	サーバーへの接続が切断された後、複数のサーバー URL が設定されているクライアントプログラムは再接続を試みようとし、EMS サーバーとの接続が再確立されるまで、リスト内の URL を繰り返します。このプロパティは、繰り返しの最大回数を決定します。このパラメータを指定しなかった場合、デフォルトでは 4 が設定されます。

表 37 接続ファクトリのパラメータ

パラメータ名	説明
reconnect_attempt_delay	再接続を試みるとき、クライアントは、次に URL リストを繰り返すまで、このパラメータで設定された間隔（ミリ秒単位）でスリープします。このパラメータを指定しなかった場合、デフォルトでは 500 ミリ秒に設定されます。
reconnect_attempt_timeout	EMS サーバーへの再接続を試みる場合、このパラメータで接続タイムアウト期間を設定し、指定した期間（ミリ秒）が経過した後で接続の試みを中断することができます。
multicast_daemon	<p>クライアントがマルチキャスト デモン への接続を確立するときに使用する TCP ポートを指定します。</p> <p>このパラメータは EMS クライアントの動作を決定しますが、マルチキャスト デモンには影響しません。マルチキャスト デモンは、クライアントが接続に使用するポートと同じポートでクライアントをリスンする必要があります。マルチキャスト デモンがリスンする TCP ポートを変更するには、デモンでコマンドラインの引数 <code>-listen</code> を使用します。詳細は、364 ページの「<a href="#">コマンドラインオプション</a>」を参照してください。</p> <p>マルチキャストについての詳細は、<a href="#">第 13 章</a>、「<a href="#">マルチキャストの使用方法</a>」を参照してください。</p>
multicast_enabled	<p>接続ファクトリのマルチキャストを無効にします。</p> <p>デフォルトでは、接続ファクトリが接続されている EMS サーバーでマルチキャストが有効になっている場合、接続ファクトリは常にマルチキャスト対応です。クライアントがマルチキャスト対応サーバーからマルチキャストを介してメッセージを受信しない場合、このプロパティを無効に設定できます。</p> <ul style="list-style-type: none"><li>• <code>enabled</code> : ファクトリでマルチキャストが有効。</li><li>• <code>disabled</code> : ファクトリでマルチキャストが無効。</li></ul> <p>マルチキャストについての詳細は、357 ページの<a href="#">第 13 章</a>、「<a href="#">マルチキャストの使用方法</a>」を参照してください。</p>
ssl-prop	<p>このファクトリが作成する接続の SSL プロパティ。</p> <p>SSL についての詳細は、453 ページの<a href="#">第 18 章</a>、「<a href="#">SSL プロトコルの使用</a>」を参照してください。</p>

例

```
[north_america]
type = topic
url = tcp://localhost:7222,tcp://server2:7222
clientID = "Sample Client ID"
ssl_verify_host = disabled
```

**設定**      このファイル内の接続ファクトリを設定するには、tibemsadmin ツールの使用をお勧めします。詳細は、125 ページの「create factory」を参照してください。

負荷分散



**持続サブスクリバが使用されている状況では、負荷分散を指定しないでください。**

持続サブスクリバを作成するクライアントプログラムが、負荷分散接続ファクトリを使用するサーバー A に接続している場合、サーバー A は持続サブスクリバを作成してサポートします。クライアントプログラムが終了して再起動し、今度はサーバー B に接続するとします。サーバー B は新しい持続サブスクリバを作成してサポートしますが、サーバー A の保留中のメッセージは、クライアントがサーバー A に再接続するまで保留されたままになります。



**アプリケーションで厳密なメッセージの順序付けが必要とされる場合、負荷分散を指定しないでください。**

負荷分散は、複数のサーバーから選択されるという性質上、厳密な順序付けを妨げるからです。

groups.conf

このファイルは、すべてのグループを定義します。ファイルは、以下の形式で記述します。

```
group-name1:"description"
    user-name1
    user-name2
group-name2:"description"
    user-name1
    user-name2
```

表 38 グループのパラメータ

パラメータ名	説明
group-name	グループの名前。グループ名は 127 文字以下にする必要があります。
description	グループについて説明する文字列。
user-name	グループに属する 1 人以上のユーザー。

例

```
administrators: "TIBCO Enterprise Message Service administrators"
    admin
    Bob
```

## jaas.conf

このファイルは、TIBCO Enterprise Message Service サーバーが JAAS LoginModule を使用するように指示します。jaas.conf ファイルについての詳細は、284 ページの「[EMS サーバーでの LoginModule の読み込み](#)」を参照してください。

## queues.conf

このファイルは、すべてのキューを定義します。ファイルは、以下の形式で記述します。

```
[jndi-name1, jndi-name2, ...]queue-name property1, property2, ...
```



JNDI 名の指定はオプションですが、指定する場合は JNDI 名を角かっこ [ ] で囲む必要があります。JNDI 名の設定についての詳細は、125 ページの「[create jndiname](#)」を参照してください。

たとえば、次のように入力します。

```
test store=mystore,secure,prefetch=2
```

アプリケーションは、このファイルに記載されているキュー、またはこのファイルに記載されているキューと名前が一致するキューのみを作成できます (acl.conf のエントリによって別途許可されていない場合)。たとえば、このファイルに foo.\* というキューが記載されている場合、アプリケーションは foo.bar および foo.baz というキューを作成できます。

キューのプロパティは、一致する名前を持つすべての静的および動的キューに継承されます。たとえば、test.\* に対して secure プロパティが設定されている場合、test.1 と test.foo も secure になります。キューに割り当て可能なプロパティについての詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

キューのプロパティの継承についての詳細は、73 ページの「[ワイルドカード\\*と>](#)」と 76 ページの「[プロパティの継承](#)」を参照してください。



サンプルファイルでは、ファイルの先頭に > というワイルドカードを指定して、アプリケーションが任意の名前で有効なキューを作成できるようにしています。キューの設定ファイルの先頭に > を付けると、キューの作成時に名前の一致が要求されません。

キュー名の制限とルールについては、52 ページの「[デスティネーション名の構文](#)」を参照してください。



routes.conf

このファイルは、この TIBCO Enterprise Message Service サーバーと、他の TIBCO Enterprise Message Service サーバー間のルートを実行します。



ルートは、管理ツール (117 ページの「第 6 章」を参照)、または管理 API (オンラインマニュアルの「com.tibco.tibjms.admin.RouteInfo」を参照) を使用して、管理者権限でのみ設定可能です。routes.conf ファイルを直接編集するとエラーが発生します。

ファイルは、以下の形式で記述します。

```
[route-name] # mandatory -- square brackets included.  
  url=url-string  
  zone_name=zone_name  
  zone_type=zone_type  
  [selector]*  
  [ssl-prop = value]*
```

表 39 ルートのパラメータ

パラメータ名	説明
[route-name]	[route-name] は、パッシブサーバーの名前 (ルートのもう一方の端)。ルートの名前になります。角かっこ [ ] は、route-name がオプションであることを示しているわけではありません。名前は角かっこで囲まれている必要があります。
url	メッセージのルート元またはルート先となるサーバーの URL。
zone_name	<p>ルートが属するルーティングゾーンの名称。指定しなかった場合、デフォルト値は default_mhop_zone となる (このデフォルト値は、4.0 より前のリリースの設定ファイルに下位互換性を提供する)。</p> <p>このパラメータはルートの作成時に設定できますが、後から変更することはできません。</p> <p>詳細は、以下のセクションを参照してください。</p> <ul style="list-style-type: none"><li>502 ページの「ゾーン」</li><li>506 ページの「ルートとゾーンの設定」</li></ul>

表 39 ルートのパラメータ

パラメータ名	説明
zone_type	<p>ゾーンのタイプ。<b>1hop</b> または <b>mhop</b> を指定します。指定しなかった場合、デフォルト値は <b>mhop</b> です。</p> <p>このパラメータはルートの作成時に設定できますが、後から変更することはできません。</p> <p><b>routes.conf</b> ファイルに指定されているゾーンのタイプが、プライマリサーバーとバックアップサーバーの共有ステートを保持している <b>\$sys.meta</b> ファイル内にすでに作成されているゾーンのタイプと一致しなかった場合、EMS サーバーは起動を拒否します。</p>
selector	<p>ルートに沿ってトピックの経路を制御するトピックセクタ (<b>incoming_topic</b> および <b>outgoing_topic</b> のパラメータ)。</p> <p>構文とセマンティクスについての詳細は、513 ページの「<a href="#">トピックメッセージのルーティングに使用するセクタ</a>」を参照してください。</p>
ssl-prop	<p>このルートの SSL プロパティ。</p> <p>SSL についての詳細は、453 ページの第 18 章、「<a href="#">SSL プロトコルの使用</a>」を参照してください。</p>

例

```
[test_route_2]
url = tcp://server2:7222
ssl_verify_host = disabled
```

stores.conf

このファイルは、EMS サーバーでメッセージまたはメタデータ（デフォルトの `$sys.meta` 定義が無効化されている場合）を保存する、ストアファイル、`mstore`、またはデータベースの場所を定義します。`stores.conf` ファイルにはひとつ以上のストアを設定できます。

設定される各ストアは、ファイルベースのストア、`mstore`、またはデータベースストアのいずれかです。ここでは、ファイルベースのストアと `mstore` のパラメータについて説明します。データベースストアのパラメータについては、[第 10 章、「データベースストアの使用」](#) を参照してください。

ファイルは、以下の形式で記述します。

```
[store_name] # mandatory -- square brackets included
  type=file
  file=name
  file_destination_defrag=size
  [file_crc=true|false]
  [file_minimum=value]
  [file_truncate=value]
  [mode=async|sync]
  [processor_id = processor id]

[store_name]
  type=mstore
  file=name
  [processor_id = processor-id]
  [scan_iter_interval=time msec|sec|min|hour|day]
  [scan_target_interval=time msec|sec|min|hour|day]
```

表 40 ストアファイルのパラメータ

パラメータ名	説明
[store_name]	[store_name] は、このストアファイル設定を識別する名前です。  角かっこ [ ] は、store_name がオプションであることを示しているわけではありません。名前は角かっこで囲まれている必要があります。
type	ストアタイプを識別します。このパラメータはすべてのストアタイプで必須です。type には次のいずれかを指定できます。 <ul style="list-style-type: none"><li>file : ファイルベースのストア</li><li>mstore : mstore</li><li>dbstore : データベースストア</li></ul> データベースストアの設定に使用するパラメータについての詳細は、297 ページの「 <a href="#">stores.conf の設定</a> 」を参照してください。

表 40 ストアファイルのパラメータ

パラメータ名	説明
file	<p>このストアファイルの作成時に使用するファイル名を指定します。このパラメータは、<code>file</code> タイプと <code>mstore</code> タイプで必須です。たとえば、<code>mystore.db</code> などです。</p> <p>このファイルの場所は、絶対パス名または相対パス名で指定できます。パスの区切り文字を指定しなかった場合、<code>tibemsd.conf</code> ファイルの <code>store</code> パラメータによって指定されている場所があればその場所にファイルが保存されます。</p>
processor_id	<p>このパラメータを指定した場合、EMS サーバーは指定したプロセスにこのストアのストレージスレッドをバインドします。</p> <p>デフォルトの動作で十分なスループットが得られる場合は、このパラメータは使用しません。ストアにプロセッサ ID が指定されていない場合、そのストアは特定のプロセッサにはバインドされません。</p> <p><code>processor-id</code> は整数で指定します。プロセッサ ID の番号は 0 で始まり、利用可能なプロセッサ数から 1 を差し引いた数まで続きます。たとえば、四つのプロセッサがある場合、利用可能なプロセッサ ID は 0、1、2、3 です。</p> <p>このパラメータには、<code>tibemsd.conf</code> の <code>processor_ids</code> パラメータと同様の要件、制限、および利点があります。使用上のガイドラインについては、115 ページの「パフォーマンス調整」を参照してください。</p>

ファイルベースのストアのパラメータ	
file_destination_defrag	<p>デスティネーションデフラグ機能で使用する最大バッチサイズを指定します。</p> <p>デスティネーションデフラグは、新しいメッセージ用に連続した領域を保持することで、ストアファイルのパフォーマンスを向上させます。さらに、サーバーの読み取りパフォーマンスも向上させます。永続的な保留メッセージがキューに蓄積され始めると、メッセージはディスクに再書き込みされるバッチにグループ化されます。メッセージは隣接して書き込まれるため、サーバーが後でメッセージをコンシューマに送信するときに、より効率的にメッセージを読み取ることができます。</p> <p><code>size</code> は、バイト、KB、MB、または GB 単位で指定します。</p> <p><code>size</code> は、ストアが参照するディスクで利用可能なサイズに設定する必要があります。たとえば、2MB に設定した場合、ディスクが 2 MB のバッチを効率的に書き込むことができます必要があります。</p> <p><code>file_destination_defrag</code> を指定しなかった場合、またはゼロに指定した場合、デスティネーションデフラグ機能は無効になります。</p>

表 40 ストアファイルのパラメータ

パラメータ名	説明
<code>file_crc</code>	<p>ストアファイルの読み取り時に、EMS サーバーが CRC を使用してデータの整合性を検証するかどうかを指定します。</p> <p>このパラメータを指定しなかった場合、デフォルトは <code>true</code> です。</p>
<code>file_minimum</code>	<p>ストアファイルのディスク領域を事前に割り当てます。事前割り当ては、サーバーが初めてストアファイルを作成したときに行われます。</p> <p>単位としては、MB または GB を指定できます。ゼロは特別な値で、事前割り当ての最小値を指定しないことを示します。これ以外の場合、4MB より大きい値を指定する必要があります。</p> <p>例：</p> <pre>file_minimum = 32MB</pre> <p><code>file_truncate</code> が <code>true</code> に設定されている場合、<code>file_minimum</code> パラメータは設定されたサイズよりも小さいファイルを EMS サーバーが切り捨てることを防ぎます。</p> <p>このパラメータを指定しなかった場合、デフォルトの事前割り当て最小値はありません。</p>
<code>file_truncate</code>	<p>EMS サーバーが不定期にストアファイルの切り捨てを行い、未使用のディスク領域を解放するかどうかを指定します。</p> <p><code>file_truncate</code> が <code>true</code> に設定されている場合、ストアファイルは切り捨てられる可能性があります。ただし、<code>file_minimum</code> で設定されているサイズより小さくなることはありません。</p> <p>このパラメータを指定しなかった場合、デフォルトは <code>true</code> で、サーバーは定期的にストアファイルの切り捨てを行います。</p>
<code>mode</code>	<p><code>mode</code> では、メッセージをストアファイルに同期と非同期のどちらで書き込むのかを指定します。モードには次のいずれかを指定します。</p> <ul style="list-style-type: none"> <li>• <code>async</code> : サーバーは非同期 I/O 呼び出しを使用してこのファイルにメッセージを保存します。</li> <li>• <code>sync</code> : サーバーは同期 I/O 呼び出しを使用してこのファイルにメッセージを保存します。</li> </ul> <p>このパラメータを指定しなかった場合、デフォルトは <code>async</code> です。</p>

表 40 ストアファイルのパラメータ

パラメータ名	説明
mstore のパラメータ	
scan_iter_interval	<p>各ストアスキャンの間隔の時間を指定します。EMS サーバーは、ここで指定した時間間隔で <b>mstore</b> の新しいセクションのスキャンを開始します。</p> <p><i>time</i> は、<b>msec</b> (ミリ秒)、<b>sec</b> (秒)、<b>min</b> (分)、<b>hour</b> (時間)、または <b>day</b> (日数) 単位で指定します。例：</p> <pre>scan_iter_interval=100msec</pre> <p>デフォルトでは、<b>mstore</b> は 10 秒間隔でストアを調査します。</p> <p>詳細は、32 ページの「<a href="#">mstore の間隔について</a>」を参照してください。</p>
scan_target_interval	<p><b>mstore</b> のフルスキャンの完了にかかるおおよその時間を制御します。</p> <p><i>time</i> は、<b>msec</b> (ミリ秒)、<b>sec</b> (秒)、<b>min</b> (分)、<b>hour</b> (時間)、または <b>day</b> (日数) 単位で指定します。例：</p> <pre>scan_target_interval=12hour</pre> <p>デフォルトでは、スキャン間隔は 24 時間です。</p> <p>詳細は、<a href="#">mstore の間隔について</a>を参照してください。</p>

例

```
[my_sync]
type = file
file = /var/local/tibems/my_sync.db
file_destination_defrag=2MB
file_crc = true
file_minimum = 10MB
file_truncate = true
mode = sync
```

例

```
[mstore1]
type = mstore
file = /var/local/tibems/mstore1.db
scan_iter_interval=100msec
scan_target_interval=12hour
```

## tibrvcn.conf

このファイルは、**tibrvcn** トランスポートにメッセージをエクスポートするトピックによって使用される **TIBCO Rendezvous 保証メッセージ (RVCM : Rendezvous Certified Messaging)** リスナーを定義します。サーバーは起動時にこれらのリスナーを事前に登録しておくことで、**tibrvcn** トランスポート経由で送信されるすべてのメッセージ（最初にパブリッシュされるメッセージを含む）を保証します。メッセージをエクスポートする前にサーバーが **RVCM** リスナーを事前に登録しなかった場合、最初のメッセージがパブリッシュされたときにリスナーが作成されますが、最初のメッセージは保証されません。

ファイルは、以下の形式で記述します。

```
transport listenerName subjectName
```

表 41 RVCM リスナーのパラメータ

パラメータ名	説明
<i>transport</i>	この RVCM リスナーのトランスポートの名前。
<i>listenerName</i>	トピックメッセージのエクスポート先となる RVCM リスナーの名前。
<i>subjectName</i>	メッセージのパブリッシュ先となる RVCM サブジェクトの名前。 <b>export</b> プロパティを指定するトピック名と同じ名前です。

### 例

```
RVCM01 listener1 foo.bar
RVCM01 listener2 foo.bar.bar
```

## topics.conf

このファイルは、すべてのトピックを定義します。ファイルは、以下の形式で記述します。

```
[jndi-name1, jndi-name2, ...]topic-name property1, property2, ...
```



JNDI 名の指定はオプションですが、指定する場合は JNDI 名を角かっこ [ ] で囲む必要があります。JNDI 名の設定についての詳細は、[125 ページ](#)の「**create jndiname**」を参照してください。

たとえば、次のように入力します。

```
business.inventory global, import="RV01,RV02", export="RV03",
maxbytes=1MB
```

アプリケーションは、このファイルに記載されているトピック、またはこのファイルに記載されているトピックと名前が一致するトピックのみを作成できます ([acl.conf](#) のエントリによって別途許可されていない場合)。たとえば、このファイルに `foo.*` というトピックが記載されている場合、アプリケーションは `foo.bar` および `foo.baz` というトピックを作成できます。

トピックのプロパティは、一致する名前を持つすべての静的および動的トピックによって継承されます。たとえば、`test.*` に対して `secure` プロパティが設定されている場合、`test.1` と `test.foo` も `secure` になります。トピックに割り当て可能なプロパティについての詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

トピックのプロパティの継承についての詳細は、73 ページの「[ワイルドカード \\* と >](#)」と 76 ページの「[プロパティの継承](#)」を参照してください。

トピック名の制限とルールについては、52 ページの「[デスティネーション名の構文](#)」を参照してください。

transports.conf

このファイルは、TIBCO Rendezvous や TIBCO SmartSockets などの外部メッセージサービスに対してメッセージのインポートとエクスポートを行うためのトランスポートを定義します。

ファイルは、以下の形式で記述します。

```
[transport_name] # mandatory -- square brackets included
  type = tibrv | tibrvcn | tibss # mandatory
  [topic_import_dm = TIBEMS_PERSISTENT |
                      TIBEMS_NON_PERSISTENT |
                      TIBEMS_RELIABLE]
  [queue_import_dm = TIBEMS_PERSISTENT |
                    TIBEMS_NON_PERSISTENT |
                    TIBEMS_RELIABLE]
  [export_headers = true | false]
  [export_properties = true | false]
  transport-specific-parameters
```

表 42 トランスポートのパラメータ

パラメータ名	説明
[transport_name]	トランスポートの名前。角かっこ [ ] は <i>transport_name</i> がオプションであることを示しているわけではありません。名前は角かっこで囲まれている必要があります。



表 42 トランスポートのパラメータ

パラメータ名	説明
type	<p>トランスポートのタイプ。</p> <ul style="list-style-type: none"> <li>• <code>tibrv</code> は TIBCO Rendezvous トランスポートを設定する。</li> <li>• <code>tibrvcn</code> は、TIBCO Rendezvous 保証メッセージトランスポートを設定する。</li> <li>• <code>tibss</code> は TIBCO SmartSockets トランスポートを設定する。</li> </ul> <p>各トランスポートには、追加の <i>transport-specific-parameters</i> が含まれます。</p>
topic_import_dm queue_import_dm	<p>EMS 送信クライアントは、各メッセージで <code>JMSDeliveryMode</code> ヘッダフィールドを設定できません。Rendezvous クライアントはこのヘッダを設定できません。代わりに、これらのふたつのパラメータを使用して、<code>tibemsd</code> がこのトランスポート上にインポートする、すべてのトピックメッセージとキューメッセージの配信モードを決定します。</p> <p><code>TIBEMS_PERSISTENT</code>   <code>TIBEMS_NON_PERSISTENT</code>   <code>TIBEMS_RELIABLE</code></p> <p>このパラメータを指定しなかった場合、デフォルトは <code>TIBEMS_NON_PERSISTENT</code> です。</p>
export_headers	<p><code>true</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> ヘッダフィールドを設定します。</p> <p><code>false</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> ヘッダフィールドを設定しません。</p> <p>このパラメータを指定しなかった場合、デフォルト値は <code>true</code> です。</p>
export_properties	<p><code>true</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> プロパティを設定します。</p> <p><code>false</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> プロパティを設定しません。</p> <p>このパラメータを指定しなかった場合、デフォルト値は <code>true</code> です。</p>
<i>transport-specific-parameters</i>	<p>トランスポート固有のパラメータを参照してください。</p>



`transports.conf` ファイルに複数の TIBCO Rendezvous トランスポートが設定されていて、EMS サーバーが最後のエントリに基づいたトランスポートの作成に失敗した場合、サーバーは、すべてのエントリを順に検索して、さらにトランスポートの作成を試みます。

## トランスポート固有のパラメータ

`type = tibrv` の場合、拡張の構文は以下のとおりです。

```
[service = service]
[network = network]
[daemon = daemon]
[temp_destination_timeout = seconds]
[rv_queue_policy = [TIBRVQUEUE_DISCARD_NONE |
                    TIBRVQUEUE_DISCARD_FIRST |
                    TIBRVQUEUE_DISCARD_LAST]:max_msgs:qty_discard]
```

詳細は、393 ページの「[Rendezvous パラメータ](#)」を参照してください。

`type = tibrvcn` の場合、拡張の構文は以下のとおりです。

```
rv_tport = name # mandatory
[cm_name = name]
[ledger_file = file-name]
[sync_ledger = true | false]
[request_old = true | false]
[explicit_config_only = true | false]
[default_ttl = seconds]
[rv_queue_policy = [TIBRVQUEUE_DISCARD_NONE |
                    TIBRVQUEUE_DISCARD_FIRST |
                    TIBRVQUEUE_DISCARD_LAST]:max_msgs:qty_discard]
```

詳細は、394 ページの「[Rendezvous 保証メッセージ \(RVCN\) のパラメータ](#)」を参照してください。

`type = tibss` の場合、拡張の構文は以下のとおりです。

```
[username = name]
[password = password]
[server_names = single_or_list_of_servers]
[project = name]
[delivery_mode = best_effort | gmd_all | gmd_some | ordered]
[lb_mode = none | round_robin | weighted | sorted]
[override_lb_mode = enable | disable]
[gmd_file_delete = enable | disable]
[import_ss_headers = none | type_num | all]
[preserve_gmd = always | receivers | never]
```

詳細は、416 ページの「[SmartSockets パラメータ](#)」を参照してください。

**例**

```
[RV01]
type = tibrv
topic_import_dm = TIBEMS_RELIABLE
queue_import_dm = TIBEMS_PERSISTENT
service = 7780
network = lan0
daemon = tcp:host5:7885
```

```
[RVCM01]
  type = tibrvc
  export_headers = true
  export_properties = true
  rv_tport = RV02
  cm_name = RVCMTrans1
  ledger_file = ledgerFile.store
  sync_ledger = true
  request_old = true
  default_ttl = 600

[SS01]
  type = tibss
  server_names = tcp:rtHost2A:5555, ssl:rtHost2B:5571
  username = emsServer6
  password = myPasswd
  project = mfg_process_control
  override_lb_mode = enable
  delivery_mode = gmd_some

[RV02]
  type = tibrv
  topic_import_dm = TIBEMS_PERSISTENT
  queue_import_dm = TIBEMS_PERSISTENT
  service = 7780
  network = lan0
  daemon = tcp:host5:7885
  rv_queue_policy = TIBRVQUEUE_DISCARD_LAST:10000:100
```

users.conf

このファイルは、すべてのユーザーを定義します。ファイルは、以下の形式で記述します。

```
username:password:"description"
```

表 43 ユーザーのパラメータ

パラメータ名	説明
username	ユーザーの名前。ユーザー名は 127 文字以下にする必要があります。

表 43 ユーザーのパラメータ

パラメータ名	説明
<i>password</i>	<p>新しいユーザーを作成する場合、このアイテムは空欄にします。例：</p> <pre>bob::"Bob Smith"</pre> <p><b>administrator</b> という、定義済みのユーザーがあります。</p> <p>ユーザーのパスワードはこの設定ファイルでは入力されず、管理ツールを使用して設定するまでブランクのままになるため、「セキュア」ではありません。詳細は、120 ページの「<a href="#">管理者へのパスワードの割り当て</a>」を参照してください。また、ユーザーを作成し、API 呼び出しを使用してパスワードを割り当てることもできます。使用する言語の API リファレンスを参照してください。</p>
<i>description</i>	ユーザーについて説明する文字列。

**例**

```
admin::"Administrator"
Bob::"Bob Smith"
Bill::"Bill Jones"
```

サーバーが起動されてパスワードが割り当てられると、ファイルは次のようになります。

```
admin:$1$urmKVgq78:"Administrator"
Bob:$2$sldfkj;lsafd:"Bob Smith"
Bill:$3$tyavmwq92:"Bill Jones"
```



## 第 8 章

# 認証とアクセス権限

ユーザーを作成してパスワードを割り当て、EMS サーバーへのアクセスを制御することができます。EMS サーバーへのアクセス制御に、LDAP サーバーなどの外部ディレクトリを使用するように EMS サーバーを設定することもできます。

また、ユーザーとグループにアクセス権限を割り当てることにより、デスティネーションに対して実行可能なアクションを制御することもできます。

この章では、EMS における認証とアクセス権限について説明します。

### トピック

---

- [「EMS のアクセス制御」](#)、256 ページ
- [「管理者のアクセス権限」](#)、257 ページ
- [「アクセス制御の有効化」](#)、265 ページ
- [「ユーザーとグループ」](#)、267 ページ
- [「ユーザーのアクセス権限」](#)、273 ページ
- [「アクセス権限の確認」](#)、276 ページ

## EMS のアクセス制御

---

EMS は、管理者とユーザーという、2 種類の基本的なアクセスレベルをサポートしています。

管理者のアクセス権限を付与されたユーザーは、管理者としてログインし、ユーザー、デスティネーション、接続、ファクトリに対して、作成、削除、ステータスの表示を行うことができます。必要なアクセス権限を持つ管理者は、ユーザーの作成、パスワードの割り当て、アクセス権限の設定の各操作により、ユーザーの EMS サーバーに対するアクセスを制御することができます。

次に、ユーザー、グループ、アクセス権限を管理者が設定する際の基本的な手順と、手順の各ステップに関する詳細な情報について説明します。

1. システムに対するアクセス制御を有効にします。265 ページの「[アクセス制御の有効化](#)」を参照してください。
2. アクセス制御が必要なデスティネーションを決定し、それらのデスティネーションに対してアクセス制御を有効にします。266 ページの「[デスティネーション制御](#)」を参照してください。
3. 管理アクセス権限が必要なユーザーを決定し、管理者がアクションをグローバルに実行できるのか、それとも特定のサブセットについてのみアクションの実行を許可するのかを決定します。詳細は、257 ページの「[管理者のアクセス権限](#)」を参照してください。
4. システムに対する権限を与えるユーザーの名前を決定し、これらのユーザーに対してユーザー名とパスワードを作成します。267 ページの「[ユーザーとグループ](#)」を参照してください。
5. グループを作成し、ユーザーをグループに割り当てます（オプション）。267 ページの「[ユーザーとグループ](#)」を参照してください。
6. ユーザーおよびグループの情報を保存に使用する外部のディレクトリを有効にします（オプション）。269 ページの「[外部ディレクトリの設定](#)」を参照してください。
7. セキュアである必要があるデスティネーションに対する特定のアクセス権限をユーザー（またはグループ）に付与することにより、アクセス制御リストを作成します。273 ページの「[ユーザーのアクセス権限](#)」を参照してください。



## 管理者のアクセス権限

管理者とは、EMS サーバーを管理できる、特別なユーザーのクラスのことです。管理者は、ユーザー、デスティネーション、ルート、ファクトリおよびその他のアイテムを作成、修正、削除することができます。一般に、管理者は、管理ツールや API を使用するために、管理アクティビティを実行するアクセス権限が付与されていなければなりません。管理者には、ユーザーの作成やすべてのキューを表示するなどの、各種のグローバルアクセス権限を付与することができます。また、管理者には、特定のデスティネーションに対してのみ、キューのページや、特定のトピックのプロパティの表示などの操作に必要な権限を付与することもできます。



管理者のアクセス権限は、管理者が、管理ツールまたは API を使う場合にのみ、サーバー内で表示したり変更したりできるアイテムを制御します。管理者コマンドは、各設定ファイル ([tibemsd.conf](#)、[acl.conf](#)、[routes.conf](#) など) 内にエントリを作成します。

特定のシステム管理者だけが設定ファイルの表示や修正を行うことができるように、設定ファイルへのアクセスを制御する必要があります。ユーザーが設定ファイルを表示または変更できるのであれば、ユーザーが設定ファイルを手動で編集してしまうと、そのユーザーが管理可能なデスティネーションを制御するためのアクセス権限を強制することができなくなります。

サーバーの設定ファイルへのアクセスを制御するには、使用しているオペレーティングシステムの機能を利用します。

管理者の作成には、管理ツールまたは管理 API を使用するか、または、直接、設定ファイルを編集する必要があります。

### 定義済みの管理者ユーザーとグループ

任意の管理者アクションを実行するために、**admin** という名前のユーザーが、特別に定義されています。**admin** に対しては、アクセス権限の付与や取り消しを行うことはできません。インストールを完了したら、ただちに、**admin** にパスワードを割り当てる必要があります。**admin** パスワードの変更についての詳細は、120 ページの「[tibemsadmin の最初の起動](#)」を参照してください。

また、システム管理者ユーザーのための特別なグループとして、**\$admin** というグループがあります。ユーザーは、このグループにメンバーとして追加されると、**admin** ユーザーと同じアクセス権限が与えられます。**\$admin** グループのメンバーであるユーザーに対して、管理者のアクセス権限を付与したり取り消したりすることはできません。**\$admin** グループには、システム全体を管理する管理者だけを割り当ててください。

## 管理アクセス権限の付与と取り消し

`tibemsadmin` の `grant` および `revoke` コマンドを使用するか、または Java や .NET の管理 API を使用すると、ユーザーに管理者のアクセス権限を付与したり取り消したりすることができます。グローバル管理者のアクセス権限、または特定のデスティネーションのアクセス権限のどちらを付与することもできます。グローバル管理者のアクセス権限の全リストについては、259 ページの「[グローバルな管理者アクセス権限](#)」を参照してください。デスティネーションの管理者のアクセス権限についての詳細は、262 ページの「[デスティネーションレベルのアクセス権限](#)」を参照してください。

グローバルおよびデスティネーションレベルのアクセス権限は、異なる管理者コマンドを使用して、個別に付与したり取り消したりすることができます。`grant` および `revoke` コマンドの構文については、122 ページの「[コマンドのリスト](#)」を参照してください。

あるユーザーにグローバルおよびデスティネーションレベルの両方の管理者のアクセス権限がある場合、そのユーザーが実行できるアクションは、そのユーザーに付与されているすべてのグローバルな管理者のアクセス権限と、デスティネーションレベルの管理者のアクセス権限の組み合わせによって決まります。たとえば、管理者に `view-destination` アクセス権限が付与されている場合、管理者は、すべてのデスティネーションに関する情報を表示することができます。特定のデスティネーションに対する表示アクセス権限がその管理者に付与されていないなくても問題ありません。

`admin` ユーザーまたは `$admin` グループ内のすべてのユーザーは、任意のユーザーに対する管理者のアクセス権限を付与したり、取り消したりできます。その他のすべてのユーザーは、`change-admin-acl` アクセス権限、および `view-user` または `view-group` が付与されていない限り、他のユーザーに対して管理者のアクセス権限を付与したり、取り消したりすることはできません。

ユーザーに `change-admin-acl` アクセス権限が付与されていても、そのユーザーが付与または取り消しできるのは、そのユーザーに付与されているアクセス権限だけです。たとえば、ユーザー BOB は `$admin` グループのメンバーではなく、`change-admin-acl` および `view-user` アクセス権限のみが付与されている場合、BOB は `view-user` または `change-admin-acl` 以外の管理者アクセス権限を他のユーザーに付与することはできません。

ユーザーには、そのユーザーが属しているグループに付与されている管理者アクセス権限のすべてが付与されます。管理者グループを作成し、作成したグループに管理者アクセス権限を付与してから、それらの管理者グループにユーザーを追加することができます。ユーザーは、そのユーザーが属しているグループに付与されているアクセス権限によって許可されている、任意の管理アクションを実行することができます。

ワイルドカードデスティネーションとして、ユーザーまたはグループに付与されたデスティネーションレベルのアクセス権限は、親デスティネーションに一致するすべての子デスティネーションに継承されます。

保護アクセス権限が設定されている場合、管理者は、管理者と同じ保護アクセス権限を持つほかのユーザーに対してのみ、アクセス権限を付与または取り消します。保護アクセス権限についての詳細は、263 ページの「[保護アクセス権限](#)」を参照してください。

## 管理者のアクセス権限の設定

管理者は、アクセス権限を与えられているアクションのみを実行できます。管理者がアクションを実行する場合、その管理者に付与されているアクセス権限のセットによる制約を受ける場合があります。

たとえば、管理者に `foo.*` デスティネーションの `view` アクセス権限が付与されているとします。この管理者には、グローバル `view-destination` アクセス権限は付与されていません。この管理者は、親デスティネーション `foo.*` に一致するデスティネーションだけを表示することができます。この管理者に、グローバル `view-acl` アクセス権限が付与されている場合、管理者は `foo.*` の親と一致するデスティネーションのアクセス制御リストだけを表示することができます。他のデスティネーションへのアクセス制御リストは、管理者が `showacl topic` または `showacl queue` コマンドを実行する場合は表示されません。

管理者ユーザーが、アクセス権限を持たずにコマンドの実行を試みると、ユーザーはエラーを受け取ります。または、単に出力が表示されない場合もあります。たとえば、管理者が `showacl queue bar.foo` コマンドを実行した場合、管理者は「Not authorized to execute command」エラーを受け取ります。これは、管理者が `foo.*` に一致するデスティネーション以外のデスティネーションを表示する権限を持っていないためです。



管理者は、`change-user` アクセス権限が付与されていなくても、いつでも自分のパスワードを変更することができます。

管理者は次のコマンドを実行することにより、自分のアクセス権限をいつでも表示できます。

```
showacl username
```

管理者に `view-acl` アクセス権限が付与されていなくても問題ありません。

## グローバルな管理者アクセス権限

特定のアクセス権限が付与される場合、管理者はユーザーの作成やすべてのキューの表示など、グローバルなアクションを実行できます。

表 44 では、グローバルな管理者アクセス権限について説明します。

表 44 グローバルな管理者アクセス権限（1 / 3 シート）

アクセス権限	管理者に許可する操作
all	すべての管理コマンドの実行。
view-all	管理対象の任意のアイテムの表示（例：ユーザー、グループ、トピック）。

表 44 グローバルな管理者アクセス権限（2 / 3 シート）

アクセス権限	管理者に許可する操作
change-acl	ユーザーレベルアクセス権限の付与と取り消し。
change-admin-acl	管理アクセス権限の付与と取り消し。
change-bridge	デスティネーションブリッジの作成および削除。
change-connection	接続の削除。
create-destination	任意のデスティネーションの作成。
modify-destination	任意のデスティネーションの変更。
delete-destination	任意のデスティネーションの削除。
change-durable	持続サブスクライバの削除。
change-factory	ファクトリの作成、削除、変更。
change-group	グループの作成、削除、変更。
change-message	サーバーに保存されているメッセージの削除。
change-route	ルートの作成、削除、変更。
change-server	サーバーパラメータの変更。
change-user	ユーザーの作成、削除、変更。
purge-destination	デスティネーションのページ。
purge-durable	持続サブスクライバのページ。
shutdown	サーバーのシャットダウン。
view-acl	ユーザーレベルアクセス権限の表示。
view-admin-acl	管理アクセス権限の表示。
view-connection	接続、プロデューサ、コンシューマの表示。
view-bridge	デスティネーションブリッジの表示。
view-destination	デスティネーションのプロパティ情報の表示。

表 44 グローバルな管理者アクセス権限（3 / 3 シート）

アクセス権限	管理者に許可する操作
<code>view-durable</code>	持続サブスクライバの表示。  持続サブスクライバを表示するには、 <b><code>view-destination</code></b> アクセス権限も必要（持続サブスクライバの情報には、サブスクライブ先のデスティネーションの情報も含まれるため）。
<code>view-factory</code>	ファクトリの表示。
<code>view-group</code>	すべてのグループの表示。  このアクセス権限を付与すると、暗黙的に <b><code>view-user</code></b> も付与される。
<code>view-message</code>	サーバーに保存されているメッセージの表示。
<code>view-route</code>	ルートの表示。
<code>view-server</code>	サーバーの設定と情報の表示。
<code>view-user</code>	任意のユーザーの表示。



アイテムにどのような種類の変更を加える場合でも、ユーザーがそのアイテムを表示できることが必要です。このため、権限の作成、変更、削除、変更、ページの各アクセス権限を付与すると、関連するアイテムを表示するのに必要なアクセス権限も暗黙的に付与されます。

特定のユーザーに、アイテムの表示のみを許可する場合は、表示アクセス権限を付与すると便利です。ユーザーがすでにアイテムを変更するアクセス権限を持っている場合、さらに表示アクセス権限を付与する必要はありません。

グローバルアクセス権限は、その他のすべてのアクセス権限とともに、**`acl.conf`** ファイルに保存されます。このファイル内のグローバルアクセス権限は、以下の構文で記述されています。

```
ADMIN USER=<username> PERM=<permission>
```

または

```
ADMIN GROUP=<groupname> PERM=<permission>
```

たとえば、BOB というユーザーにグローバルな管理者のアクセス権限 `view-user` が付与され、`sys-admins` グループに `change-acl` アクセス権限が付与されている場合、次のエントリが `acl.conf` ファイルに追加されます。

```
ADMIN USER=BOB PERM=view-user
ADMIN GROUP=sys-admins PERM=change-acl
```

デスティネーションレベルのアクセス権限

管理者に対して、デスティネーションごとにアクセス権限を付与することができます。デスティネーションレベルのアクセス権限は、特定のデスティネーションにおいてユーザーが実行可能な管理機能を制御します。ユーザーに付与されているグローバル管理アクセス権限は、すべてのデスティネーションレベルのアクセス権限に優先します。

デスティネーションレベルの管理アクセス権限の一般的な用途は、ユーザーの異なるグループに、ワイルドカードデスティネーションに対するアクセス権限を指定することです。これにより、ユーザーのグループが管理制御可能な特定のデスティネーションを指定できます。たとえば、あるグループにすべての `ACCOUNTING.*` トピックの制御を許可し、別のグループにすべての `PAYROLL.*` キューの制御を許可できます。

表 45 では、デスティネーションレベルの管理アクセス権限について説明します。

表 45 デスティネーションレベルの管理アクセス権限

アクセス権限	管理者に許可する操作
view	このデスティネーションの情報の表示。
create	指定されたデスティネーションの作成。このアクセス権限は、特に、ワイルドカードデスティネーション名との組み合わせ時に有用です。これにより、ユーザーは、指定された親と一致する任意のデスティネーションを作成できます。
delete	デスティネーションの削除。
modify	このデスティネーションのプロパティの変更。
purge	デスティネーションがキューの場合、このキューをパージする。または、デスティネーションが持続サブスクライバを持つトピックの場合、持続サブスクライバをパージする。



アイテムにどのような種類の変更を加える場合でも、ユーザーがそのアイテムを表示することが必要です。このため、権限の作成、修正、削除、変更、ページのアクセス権限を付与すると、関連するアイテムを表示するためのアクセス権限の付与も暗黙的に付与されます。

特定のユーザーに、アイテムの表示のみを許可する場合は、表示アクセス権限を付与すると便利です。ユーザーがすでにアイテムを変更するアクセス権限を持っている場合、さらに表示アクセス権限を付与する必要はありません。

デスティネーションの管理アクセス権限は、その他のすべてのアクセス権限とともに、**acl.conf** ファイルに保存されます。たとえば、ユーザー BOB がトピック **foo** に対するパブリッシュおよびサブスクライブアクセス権限を持っている場合、BOB には表示アクセス権限も付与されます。**acl** には以下のように記述されます。

```
TOPIC=foo USER=BOB PERM=publish,subscribe,view
```



デスティネーションに対するユーザーと管理者のアクセス権限は両方とも、**acl.conf** ファイルの同じエントリに保存されます。これは、わかりやすさよりも利便性を重視しているためです。ユーザーのアクセス権限は、クライアントアプリケーションがデスティネーションに対して実行できるアクションを指定します（パブリッシュ、サブスクライブ、送信、受信など）。管理者のアクセス権限は、管理ツールまたは管理 API の使用時に、ユーザーがデスティネーションに対して実行できる管理コマンドを指定します。

## 保護アクセス権限

保護アクセス権限を使用すると、ユーザーを管理ドメインにグループ化し、管理者がそのドメイン内でのみアクションを実行できるようにできます。管理者は、同じ保護アクセス権限を持つユーザーに対してのみ、管理作業を実行できます。保護アクセス権限には、**protect1**、**protect2**、**protect3**、および **protect4** の 4 種類があり、四つの管理者グループを作成することができます。保護アクセス権限は、**admin** ユーザーまたは **\$admin** グループ内のユーザーには適用されません。これらのユーザーは、保護アクセス権限に関係なく、任意のユーザーに対して任意のアクションを実行できます。

保護アクセス権限を使用するには、保護アクセス権限のひとつをユーザーのセットに対して付与します（個々のユーザーに付与することも、定義されたグループに付与することもできます）。次に、同じ保護アクセス権限を、これらのユーザーに対してアクションを実行できる管理者に付与します。



たとえば、会社内に営業、財務、製造、システム管理の四つの部門があるとし、各部門には、定義済みのグループと、そのグループに割り当てられたユーザーのセットが設定されています。システム管理部門には、マネージャがひとりと管理者が 3 人いて、それぞれが他の部門のリソース管理を担当しています。システム管理部門のマネージャは、任意の管理アクションを実行することができます。他のシステム管理者は、各自が担当しているグループのメンバーに対してのみ、管理アクションを実行することができます。

マネージャのユーザー名は `mgr` で、その他のシステム管理者のユーザー名は、それぞれ `admin1`、`admin2`、および `admin3` です。次の各コマンドは、この管理構造の例を作成するのに必要な許可を説明しています。

```
add member $admin mgr
grant admin sales protect1
grant admin admin1 protect1,all
grant admin manufacturing protect2
grant admin admin2 protect2,all
grant admin finance protect3
grant admin admin3 protect3,all
```



保護アクセス権限は、`all` 権限に加えて付与できます。この場合、そのユーザーは、同じ保護アクセス権限を持つすべてのユーザーに対して、すべての管理者アクセス権限を持ちます。ただし、ユーザーの `all` アクセス権限を取り消す場合、保護アクセス権限を含むすべてのアクセス権限が、ユーザーのアクセス制御リストから削除されます。

管理者は、異なる保護アクセス権限が設定されたユーザーを表示できますが、アクションを実行できるのは、同じ保護アクセス権限が設定されたユーザーに対してのみです。

たとえば、`admin1` は `sales` グループ内のすべてのユーザーに対してあらゆるアクションを実行でき、また、`manufacturing` または `finance` グループ内のすべてのユーザーを表示できます。ただし、`admin1` は、`manufacturing` または `finance` グループのユーザーに対して、アクセス権限を付与したり、パスワードを変更したり、グループからユーザーを削除したり、その他の管理アクションを実行することはできません。ユーザー `mgr` は、保護アクセス権限にかかわらず、すべてのユーザーに対してあらゆるアクションを実行できます。これは、`mgr` が `$admin` グループのメンバーであるためです。



## アクセス制御の有効化

管理者は、サーバーに対するアクセス制御を有効または無効にできます。また、管理者は、特定のデスティネーションに対するアクセス権限の確認を有効または無効にすることもできます。

### サーバー制御

主要設定ファイル内のサーバー制御プロパティは、サーバーによって管理されているすべてのデスティネーションのアクセス保護を、有効または無効にすることができます。[authorization](#) プロパティは、ユーザー名とパスワードの検証を有効または無効にすることができます。



デフォルト設定は `disabled` です。セキュアな実装では、管理者は明示的に `authorization` を `enabled` に設定する必要があります。

`authorization` が `disabled` に設定されている場合、サーバーは任意の接続要求を承認し、クライアントが、トピックに対するメッセージのパブリッシュなどのためにデスティネーションにアクセスする際にアクセス権限を確認しません。

`authorization` が有効になっている場合、サーバーは、適切な権限を持つ有効なユーザーからの接続のみを承認します。サーバーは、セキュアデスティネーションに関係するクライアント操作のアクセス権限を確認します。

`authorization` を有効にする場合は、[tibemsd.conf](#) を編集します (`authorization` プロパティを `enabled` に設定し、サーバーを再起動します)。または、`tibemsadmin` ツールを使用し、次の `set server` コマンドを使用して、動的に認可を有効にします。

```
set server authorization=enabled
```

認可は、フォールトトレランスサーバーのペア間の接続には影響しません。詳細は、487 ページの「[認可とフォールトトレランスサーバー](#)」を参照してください。

`authorization` が `disabled` になっている場合でも、管理機能を実行するには、管理者は、常に正しい管理ユーザー名とパスワードでログインする必要があります。

## デスティネーション制御

サーバーの `authorization` が有効になっている場合、サーバーは、例外なしにすべての接続のユーザー名とパスワードを確認します。ただし、メッセージの送受信などのデスティネーションに対する操作は、デスティネーションの `secure` プロパティが有効になっていない限り、検証されません。`secure` が有効になっているデスティネーション上のアプリケーションによるすべての操作は、`acl.conf` にリストされたアクセス権限に基づいて、サーバーによって検証されます。`secure` が無効になっているデスティネーションでは、何の制限も受けずに操作が続行されます。



`secure` プロパティは SSL レベルセキュリティとは関係ありません。`secure` プロパティは、基本の認証とアクセス権限の検証のみを制御します。クライアントとサーバー間の通信のセキュリティには影響しません。

デスティネーションの `secure` プロパティが設定されていない場合、認可されたユーザーはトピックまたはキューに対して任意のアクションを実行できます。

デスティネーションプロパティについての詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

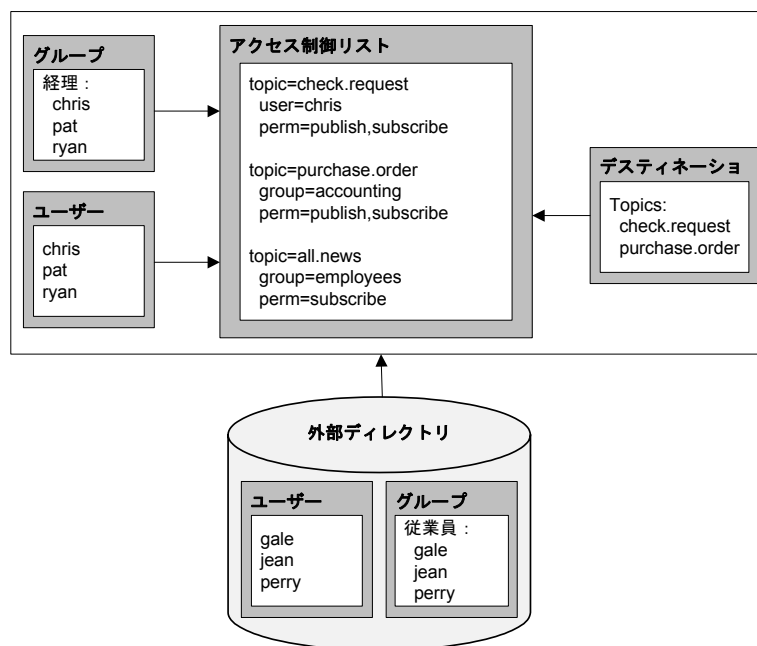
## ユーザーとグループ

ユーザーアクセス権限は、ユーザーが各宛先（トピックとキュー）に対して実行できる操作に適用されます。アクセス権限を使用することにより、どのユーザーがキューに対するメッセージを送信、受信、または参照できるかを制御できます。また、どのユーザーがトピックをパブリッシュまたはサブスクライブでき、どのユーザーがトピックに対する持続サブスクリプションを作成できるかも制御できます。アクセス権限は、サーバーのアクセス制御リストに保存されます。

グループを使用すると、ユーザーのクラスを作成し、よりグローバルなレベルでアクセス権限を制御することができます。個々のユーザーごとにデスティネーションに対するアクセス権限を付与したり、取り消したりする代わりに、グループレベルでデスティネーションに対するアクセスを制御できます。ユーザーは直接付与されたアクセス権限に加え、属する各グループからのアクセス権限を継承します。

図 15 は、ユーザー、グループ、アクセス権限の関係を示しています。

図 15 ユーザー、グループ、およびアクセス権限



外部で設定されたユーザーとグループは、外部ディレクトリを使用して定義および管理されます。ローカルで設定されているユーザーとグループ、およびアクセス制御リストは、設定ファイルを編集する、または、管理ツールまたは管理 API を使用するという、いずれの方法でも設定できます。



アクセス制御と SSL (Secure Sockets Layer) には、いくつかの類似点があります。SSL を使用すると、サーバーはユーザーのデジタル証明書を使用したユーザー認証を要求することができます。ただし、SSL は、デスティネーションレベルのアクセス制御は指定しません。ここで説明する SSL およびアクセス制御機能は、個別にまたは両方一緒に使用することで、システムへのセキュアなアクセスを確保します。SSL についての詳細は、453 ページの第 18 章、「SSL プロトコルの使用」を参照してください。

以降のセクションでは、EMS 内のユーザーとグループについて詳しく説明します。

## ユーザー

ユーザーは、固有の名前による ID であり、サーバーが個々のユーザーを識別するために使用されます。クライアントがログインするときに、接続要求にはユーザー名と、ユーザー名に関連付けられたパスワードが含まれている必要があります。



特別な場合に、サーバーへの匿名アクセスを許可することもできます。この場合、接続要求にユーザー名とパスワードは必要ありません。匿名ログインを許可するようにサーバーを設定するには、パスワードを指定せずに、`anonymous` という名前のユーザーを作成する必要があります。匿名ログインは、`anonymous` というユーザーを設定しないと、許可されません。

匿名でログインしたクライアントは、`anonymous` ユーザーが実行を許可されているアクションのみを実行できます。

唯一の定義済みのユーザー、`admin` は、他のユーザーの作成など、管理作業を実行できます。

ユーザーの作成と削除、およびパスワードの変更を行うには、`users.conf` 設定ファイル内にユーザーを指定するか、または `tibemsadmin` ツールを使用してユーザーを指定するか、あるいは管理 API を使用する方法があります。設定ファイル内のユーザーの指定についての詳細は、252 ページの「`users.conf`」を参照してください。`tibemsadmin` ツールを使ったユーザーの指定についての詳細は、117 ページの第 6 章、「EMS 管理ツールの使用」を参照してください。管理 API についての詳細は、オンラインマニュアルを参照してください。

## グループ

グループを使用すると、ユーザーのクラスを作成できます。グループを使用すると、アクセス制御の管理が非常に簡単になります。これは、グループに対する1回の操作で、多数のユーザーにアクセス権限を付与したり、取り消したりできるからです。各ユーザーは、必要な数だけ異なるグループに所属できます。ユーザーのアクセス権限は、ユーザーが属しているグループのアクセス権限の組み合わせに、ユーザーに直接付与されているアクセス権限を加えたものです。

`groups.conf` 内にグループを指定するか、または `tibemsadmin` ツールを使用してグループを指定するか、あるいは管理 API を使用することによって、グループを作成、削除したり、グループにユーザーを追加したりすることができます。設定ファイル内でのグループの指定についての詳細は、[239 ページの「groups.conf」](#)を参照してください。`tibemsadmin` ツールを使ったグループの指定についての詳細は、[117 ページの第 6 章、「EMS 管理ツールの使用」](#)を参照してください。管理 API についての詳細は、オンラインマニュアルを参照してください。

## 外部ディレクトリの設定

EMS サーバー設定ファイル、または LDAP サーバーなどの外部ディレクトリに、ユーザー認証およびグループ情報を定義することができます。

### 外部ユーザー認証

EMS は、LDAP サーバーなどの外部ディレクトリサーバーに保存されているユーザーも認証するように設定できます。

ユーザーを認証する際、EMS サーバーは `tibemsd.conf` 内のパラメータ `user_auth` を参考にします。ユーザーが EMS サーバーで、認証を受けようと試みるときに、このパラメータは認証情報のソースを指定します。このパラメータは、以下の値をひとつ以上指定できます(複数指定する場合は、コンマで区切ります)。

- `local` : ローカル EMS サーバーのユーザー設定からユーザーの認証情報を取得する。
- `ldap` : LDAP ディレクトリサーバーからユーザーの認証情報を取得する (LDAP 固有の設定パラメータを参照してください)。
- `jaas` : カスタム認証モジュールからユーザーの認証情報を取得する ([282 ページの「拡張可能な認証」](#)を参照してください)。

ユーザーが認証を試みるたびに、サーバーは、対応する認証情報を取得するために、このパラメータで指定された場所を指定された順番で検索します。EMS サーバーは、指定されたソースのいずれかを使って認証に成功した場合、その認証を受け入れます。

## グループ情報

EMS サーバーは、外部ディレクトリに保存されているグループ情報も取得します。静的グループおよび動的グループがサポートされます。このどちらか一方または両方を取得するように EMS サーバーを設定できます。

## 管理コマンドおよび外部ユーザーと外部グループ

ローカル（EMS サーバーのローカル設定ファイル）または外部の LDAP に定義されたユーザーとグループに管理コマンドを実行できます。さらに、異なる場所に定義されているユーザーとグループを組み合わせることもできます。たとえば、LDAP に定義されているユーザーとグループのアクセス権限を付与したり取り消したり、または LDAP で定義されているユーザーをローカルに定義されているグループに追加したりできます。



認証ソースを組み合わせるには、設定パラメータの `user_auth` に `local` と `ldap` の両方が指定されている必要があります。

`show user/s` または `show group/s` コマンドを使用してユーザーとグループを表示する場合、外部ディレクトリに存在しているすべてのユーザーとグループに対して、名前のとりにアスタリスクが表示されます。これらのコマンドを使用して、外部ディレクトリのユーザーとグループを表示できるのは、以下の場合のみです。

- 外部で定義されたユーザーの認証に成功したとき
- 外部で定義されたグループに属するユーザーの認証に成功したとき
- 外部で定義されたユーザーがローカルに定義されたグループに追加されたとき
- トピックまたはキューのアクセス権限が、外部で定義されたユーザーまたはグループに付与されたとき

このため、外部ディレクトリで定義されているすべてのユーザーとグループが、`show user/s` または `show group/s` コマンドの実行時に表示されるとは限りません。これらのコマンドが実行されたとき、前述の条件を満たすユーザーとグループだけが表示されます。

外部で定義されたユーザーとグループと同じ名前のユーザーとグループを作成できます。サーバーの設定に存在するユーザーまたはグループが外部にも定義されている場合、ユーザーのローカル定義が優先されます。`show user/s` または `show group/s` コマンドを実行した場合、ローカルに定義されているユーザーとグループの名前には、アスタリスクは表示されません。

`delete user` または `delete group` コマンドを実行して、ローカルサーバーの設定からユーザーおよびグループを削除することができます。ユーザーまたはグループに割り当てられたアクセス権限は、そのユーザーまたはグループが削除された時点で削除されます。外部で定義されたユーザーまたはグループを削除すると、サーバーのメモリからユーザーまたはグループが削除され、アクセス制御リ

ストに割り当てられているアクセス権限も削除されますが、外部ディレクトリには何の影響もありません。外部で定義されたユーザーは、再びログインできます。このユーザーはサーバーのメモリ内に作成され、ユーザーが属するグループも作成されます。ただし、このユーザーまたはグループのアクセス権限はすべて削除済みであるため、再度、付与する必要があります。

## LDAP ディレクトリサーバーの使用

EMS は、次の外部ディレクトリサーバーに対してテスト済みです。

- Netscape/SunOne iPlanet Directory Server バージョン 5.1
- Windows 2000 Server の機能の一部として出荷されている Microsoft Active Directory

ただし、基本的には、LDAP V2 に準拠している任意の外部ディレクトリサーバーが使用できます。

177 ページの「[tibemspd.conf](#)」の説明には、外部ディレクトリサーバーを設定するための設定パラメータの完全なリストが記載されています。表 46 では、よく利用される LDAP サーバーのデフォルト設定のためのパラメータ設定について説明します。

表 46 よく利用される LDAP サーバーのデフォルト設定 (1 / 2 シート)

外部ディレクトリ サーバー	パラメータ設定
iPlanet	<pre> ldap_principal = cn=Directory Manager  ldap_user_class = Person ldap_user_attribute = uid ldap_user_base_dn = ou=people, o=&lt;your_organization&gt; ldap_user_filter = (&amp;(uid=%s)(objectclass=person))  ldap_group_base_dn = "ou=groups, o=&lt;your_organization&gt; ldap_group_filter = ( (&amp;(cn=%s)(objectclass=groupofUniqueNames))(&amp; (cn=%s)(objectclass=groupOfURLs))) ldap_static_group_class = groupofuniquenames ldap_static_group_attribute = cn ldap_static_member_attribute = uniquemember ldap_dynamic_group_class = groupofURLs ldap_static_group_member_filter = (&amp;(uniquemember=%s)(objectclass=groupofuniquen ames)) ldap_dynamic_group_class = groupofURLs ldap_dynamic_group_attribute = cn ldap_dynamic_member_url_attribute = memberURL </pre>

表 46 よく利用される LDAP サーバーのデフォルト設定 ( 2 / 2 シート)

外部ディレクトリ サーバー	パラメータ設定
Active Directory	<pre>ldap_principal = CN=Administrator, CN=Users, DC=&lt;your_domain&gt;  ldap_user_class = user ldap_user_attribute = cn ldap_user_filter = (&amp;(cn=%s)(objectclass=user))  ldap_group_filter = (&amp;(cn=%s)(objectclass=group)) ldap_static_group_class = group ldap_static_group_attribute = cn ldap_static_member_attribute = member ldap_static_group_member_filter = (&amp;(member=%s)(objectclass=group))</pre>
Open LDAP	<pre>ldap_user_class = person ldap_user_attribute = cn ldap_user_base_dn = ou=people, dc=&lt;your_domain_component&gt;, dc=&lt;your_domain_component&gt; ldap_user_filter = (&amp;(cn=%s)(objectclass=user))  ldap_group_base_dn = ou=groups, dc=&lt;your_domain_component&gt;, dc=&lt;your_domain_component&gt; ldap_group_filter = (&amp;(cn=%s)(objectclass=groupofnames)) ldap_static_group_class = groupofnames ldap_static_group_attribute = cn ldap_static_member_attribute = member ldap_static_group_member_filter = (&amp;(member=%s)(objectclass=groupofnames))</pre>
Novell	<pre>ldap_user_class = person ldap_user_attribute = cn ldap_user_base_dn = ou=people, o=&lt;your_organization&gt; ldap_user_filter = (&amp;(cn=%s)(objectclass=person)) ldap_group_base_dn = ou=groups, o=&lt;your_organization&gt; ldap_group_filter = (&amp;(cn=%s)(objectclass=groupofnames)) ldap_static_group_class = groupofnames ldap_static_group_attribute = cn ldap_static_member_attribute = uniquemember ldap_static_group_member_filter = (&amp;(uniquemember=%s)(objectclass=groupofnames))</pre>



## ユーザーのアクセス権限

デスティネーションに対してユーザーが実行できるアクションは、アクセス制御リストに保存されているユーザーのアクセス権限により決定されます。ユーザーのアクセス権限は、そのユーザーに明示的に付与されたアクセス権限と、そのユーザーがグループに所属していることによって与えられている権限の組み合わせです。

ユーザーアクセス権限を付与する場合は、アクセス権限を付与するユーザーまたはグループ、デスティネーションの名前、および付与するアクセス権限を指定します。アクセス権限の付与は、**authorization** サーバーパラメータとも、関連するデスティネーションの **secure** プロパティとも無関係なアクションです。現在付与されているアクセス権限はアクセス制御ファイルに保存されていますが、サーバーは **authorization** が有効になっている場合に、セキュアデスティネーションに対してのみ、それらの権限を適用します。



外部で定義されたユーザーとグループにアクセス権限を設定する場合、ユーザーとグループの名前では大文字と小文字が区別されます。アクセス権限を設定する場合は、大文字と小文字を正しく指定してください。

ユーザーのアクセス権限は、257 ページの「[管理者のアクセス権限](#)」の説明にあるように、適切なアクセス権限を持つ管理者だけが設定できます。

アクセス権限を割り当てるには、**acl.conf** ファイル内に指定するか、**tibemsadmin** ツールを使用して指定するか、または管理 API を使用して指定します。ユーザーアクセス権限を設定する場合、明示的にデスティネーション名を指定するか、またはワイルドカードデスティネーション名を指定することができます。ワイルドカードデスティネーション名およびアクセス権限についての詳細は、274 ページの「[ユーザーアクセス権限の継承](#)」を参照してください。

[表 47](#) は、キューにアクセスできるようにするためにユーザーに付与されるアクセス権限のリストです。274 ページの「[表 48](#)」は、トピックにアクセスするためのアクセス権限のリストです。

表 47 キューのアクセス権限

名前	説明
receive	キューレシーバを作成するためのアクセス権限
send	キューセNDERを作成するためのアクセス権限
browse	キューブラウザを作成するためのアクセス権限

表 48 トピックのアクセス権限

名前	説明
subscribe	トピックに対する非持続サブスクライバを作成するためのアクセス権限
publish	トピックにパブリッシュするためのアクセス権限
durable	トピックに対する持続サブスクライバを作成、削除、修正するためアクセス権限
use_durable	トピックに対する既存の持続サブスクライバを使用するためのアクセス権限。ただし、持続サブスクライバの作成、削除、修正はできない。

ユーザーパラメータの設定の例

ユーザー bob は、[acl.conf](#) ファイルに、次のアクセス権限が記録されています。

```
USER=bob TOPIC=foo PERM=subscribe,publish
```

このアクセス権限のセットは、bob がトピック foo をサブスクライブしたり、このトピックにメッセージをパブリッシュしたりできますが、持続サブスクライバを foo に対して作成することはできないことを示しています。

bob は、グループ engineering のメンバーであり、このグループの acl ファイルに次のエントリがあるとしています。

```
GROUP=engineering TOPIC=bar PERM=subscribe,publish
```

その場合、bob は、トピック foo および bar に対してパブリッシュおよびサブスクライブすることができます。

ユーザー bob とグループ engineering の両方が [acl.conf](#) ファイルにエントリを持っている場合、bob には、bob 自身が直接持つすべてのアクセス権限セットと、グループ engineering のアクセス権限を組み合わせたアクセス権限が付与されます。

ユーザーアクセス権限の継承

ワイルドカード指定を使用して、トピックまたはキューへのアクセス権限を、ユーザーに付与する場合、指定に一致する、作成されたすべてのトピックおよびキューは、親トピックのアクセス権限と同じアクセス権限を付与されます。複数の親トピックがある場合、ユーザーは、任意の子トピックのすべての親トピックのアクセス権限の和集合を受け取ります。ワイルドカード指定に一致するトピックまたはキューのアクセス権限をユーザーに付与することはできません。

たとえば、Bob というユーザーに、キュー `foo.*` に対する `browse` アクセス権限を付与することができます。ユーザー Bob は、`foo.bar` キューに対する `browse` アクセス権限を受け取ります。さらに、`foo.bar` キューに対する `send` アクセス権限を Bob に付与することもできます。ただし、Bob が継承した、`foo.bar` キューに対する `browse` アクセス権限を削除することはできません。

デスティネーション名に含まれるワイルドカードについての詳細は、73 ページの「[ワイルドカード](#)」を参照してください。

## ユーザーのアクセス権限の取り消し

管理者は、ユーザーがデスティネーションにコンシューマを作成するためのアクセス権限を取り消すことができます。権限がなければ、ユーザーはデスティネーションに新しいコンシューマを作成できません。ただし、デスティネーションの既存のコンシューマは引き続きメッセージを受信します。

取り消すことができるのは、直接付与されたアクセス権限だけです。ユーザーがグループから継承したアクセス権限を取り消すことはできません。また、親トピックから継承したアクセス権限も取り消すことはできません。`tibemsadmin` の `revoke` コマンドは、[acl.conf](#) ファイル内の特定のエントリからしかアイテムを削除できません。また、その他のエントリから継承したアイテムは削除できません。

アクセス権限は、いくつかの方法で取り消すことができます。

- `acl.conf` ファイルのエントリを削除または編集する。
- `tibemsadmin` の `revoke` コマンドを使用する。詳細は、[134 ページ](#) を参照。
- 管理 API を使用する。

## アクセス権限の確認

アクセス権限が適用される場合（つまり、**authorization** 設定プロパティが設定されており、デスティネーションの **secure** プロパティが設定されている場合）、デスティネーションに対する操作をユーザーが実行しようとしたときに、サーバーがアクセス権限を確認します。たとえば、トピックに対するサブスクリプションを作成すると、キューにメッセージが送信されます。アクセス権限は動的に付与されたり、取り消されたりするため、サーバーは、デスティネーションに対して操作が実行されるたびにアクセス権限を確認します（コンシューマやプロデューサーが作成されるときも毎回確認します）。

特定のデスティネーション名（ワイルドカードではない）については、ユーザーが以下のいずれかのアクションを実行するときにアクセス権限が確認されます。

- トピックに対するサブスクリプションを作成する
- キューのコンシューマになろうと試みる
- トピックまたはキューに対してメッセージをパブリッシュまたは送信する
- キューブラウザを作成しようとする

ユーザーは、明示的に適切なアクセス権限を付与されていないデスティネーションに対しては、メッセージを作成することも送信することもできません。デスティネーションに対してメッセージを作成または送信する前に、ユーザーはデスティネーションに対するアクセス権限を付与されている必要があります。

ただし、ワイルドカードトピック名（キューコンシューマはワイルドカードを指定できない）については、ユーザーが非持続サブスクライバを作成する場合、アクセス権限は確認されません。このため、ユーザーは、トピック **foo.\*** またはその任意の子トピックに対する明示的なアクセス権限を持たなくても、**foo.\*** に対するサブスクリプションを作成できます。これにより、管理者は、ユーザーのアプリケーションがサブスクリプションを作成した後に、必要なアクセス権限をユーザーに付与することができます。あいまいなワイルドカードトピックに対するユーザーのサブスクライバを許可し、後から特定のトピックに対するアクセス権限を付与することもできます。ワイルドカードトピックまたはひとつ以上の子トピックに対するアクセス権限が付与されない限り、ユーザーは、ワイルドカードサブスクリプションに基づくメッセージを受信することはできません。



持続サブスクライバを作成する場合、ユーザーは、サブスクライバ先のトピックに対して明示的に設定された **durable** アクセス権限を持っている必要があります。たとえば、トピック **foo.\*** に対する持続サブスクライバを作成するには、ユーザーは、トピック **foo.\*** に対して持続サブスクリプションを作成するための **durable** アクセス権限を付与されている必要があります。あるトピックに対する既存の持続サブスクライバをサブスクライブするには、そのトピックに対して設定された **durable** または **use\_durable** アクセス権限を持っている必要があります。

## アクセス権限確認の例

この例では、ユーザーに対してアクセス権限を付与および取り消す過程をひととおり説明します。併せて、各種の操作を行ったときの結果も説明します。

1. ユーザー **bob** は、トピックをサブスクライブし、これらのトピックに送信されたメッセージを表示する EMS アプリケーションを使用しています。
2. ユーザー **bob** は、**user.\*** に対するサブスクリプションを作成します。このトピックは、各ユーザーの親トピックです。メッセージは、各ユーザーに定期的に送信されます（たとえば、メッセージはトピック **user.bob** に送信されます）。同じアプリケーションが複数のユーザーによって使用されるため、アプリケーションは親トピックに対するサブスクリプションを作成します。
3. ユーザー **bob** は、**corp.news** に対するサブスクリプションを作成します。このトピックへのアクセス権限が **bob** に付与されていないため、この操作は失敗します。
4. トピック **user.bob** に対してメッセージは送信されますが、このトピックへのアクセス権限が **bob** に付与されていないため、アプリケーションはメッセージを受信しません。
5. 管理者は、アプリケーションの日常保守業務の一環として、新しいユーザーへトピックに対するアクセス権限を付与します。管理者は、ユーザー **bob** に、トピック **user.bob** および **corp.\*** に対するサブスクライブアクセス権限を付与します。これらのアクセス権限の付与は動的に行われ、ユーザー **bob** はトピック **user.bob** に送信されたメッセージを受信できるようになり、またトピック **corp.news** に対するサブスクライブも可能になります。
6. 管理者はトピック **user.bob** についてのメッセージを **bob** に送信し、すべての **corp.\*** トピックに対するアクセス権限が付与されたことを通知します。
7. アプリケーションは、トピック **user.bob** についての新しいメッセージを受信し、そのメッセージを表示します。
8. ユーザー **bob** は、トピック **corp.news** のサブスクリプションの作成を試み、成功します。
9. トピック **corp.news** にメッセージが送信されます。ユーザー **bob** のアプリケーションはこのメッセージを受信し、表示します。
10. 管理者は、**bob** が契約社員であり、正社員ではないことに気付いたため、トピック **corp.\*** に対する **bob** のサブスクライブアクセス権限を取り消します。  
  
**corp.news** に対するサブスクリプションは、ユーザー **bob** のアプリケーションにまだ存在しますが、**bob** は **corp.\*** トピックの子に対する新規サブスクリプションを作成することはできません。



## 第 9 章

# 拡張可能なセキュリティ

この章では、カスタムの認証モジュールとアクセス権限モジュールの開発と実装方法の概要を説明します。

### トピック

---

- [「拡張可能なセキュリティの概要」、280 ページ](#)
- [「拡張可能な認証」、282 ページ](#)
- [「拡張可能なアクセス権限」、285 ページ](#)
- [「EMS サーバー内の JVM」、292 ページ](#)

## 拡張可能なセキュリティの概要

---

拡張可能なセキュリティ機能を使用すると、EMS に組み込まれているデフォルトの LDAP サーバーに加えて、独自の認証およびアクセス権限システムを使用して、ユーザーを認証し、パブリッシュ操作やサブスクライブ操作などのアクションの実行を認可できるようになります。認証およびアクセス権限を付与するカスタムアプリケーションを作成すると、システムの設計をより柔軟に行うことができます。

### 拡張可能なセキュリティの機能

拡張可能なセキュリティは、EMS サーバーの Java 仮想マシン (JVM) で実行する独自の認証モジュールおよびアクセス権限モジュールをユーザーが作成できるようにすることで機能します。これらのモジュールは、認証モジュールの場合は Java Authentication and Authorization Service (JAAS) を使用し、アクセス権限モジュールの場合は Java Access Control Interface (JACI) を使用して、サーバーに接続します。

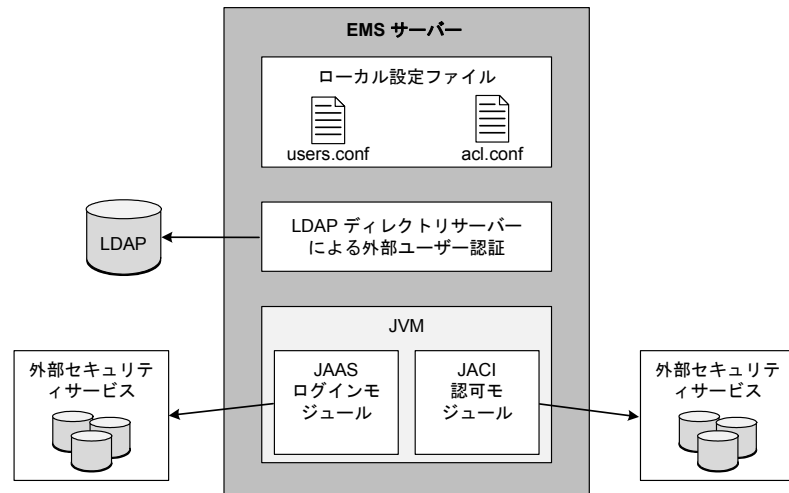
EMS サーバーの起動時に拡張可能なセキュリティ機能が有効になっている場合、サーバーは接続時に各ユーザーを確認して認証し、認可が必要なアクションを実行しようとしているときは、ユーザーのアクセス権限を確認します。

アクセス権限の結果は指定されたタイムアウトまでサーバー内にキャッシュされ、キャッシュされたアクセス権限の期限が切れると、アクセス権限モジュールが再起動されます。その後、サーバーは古いアクセス権限データを新しいデータに置き換えます。

拡張可能な認証と拡張可能なアクセス権限は、`tibemsd.conf` 設定ファイルで有効に設定されます。拡張可能なセキュリティモジュールは、TIBCO Enterprise Message Service フレームワークの外部で機能するシングルサインオン (SSO: Single Sign On) サーバーまたは LDAP ディレクトリなどの外部セキュリティサービスに接続できます。拡張可能なセキュリティモジュールは、LDAP または EMS の `acl.conf` 設定ファイルなど、他の認可およびアクセス権限方式と連携して機能できます。281 ページの「[図 16](#)」には、サーバーで使用できるさまざまなセキュリティ方式が示されています。



図 16 ユーザーの認証およびアクセス権限の確認方式



## 拡張可能な認証

拡張可能な認証機能は Java 仮想マシン (JVM) と Java Authentication and Authorization Service (JAAS) を使用して、ユーザーが独自の Java ベースの認証モジュールを EMS サーバーで実行できるようにします。

EMS サーバー内の JVM で実行されている認証モジュール、つまり `LoginModule` は、JAAS インターフェイスを使用して `tibemsd` によってアクセスされます。この方法で、EMS アプリケーションのセキュリティを柔軟に拡張することができます。`LoginModule` を使用して既存の認証プロセスを増やすことができます。または、`LoginModule` を、EMS サーバーによって使用される唯一の認証方式にすることができます。主要設定ファイル内の `user_auth` パラメータによって、`LoginModule` が使用される条件が決まります。

EMS クライアントがサーバーへの接続を確立しようとするたびに、サーバーは接続を受け入れる前にクライアントを認証します。拡張可能な認証が有効になっていると、`tibemsd` はユーザー名とパスワードを `LoginModule` に渡し、それによって許可または拒否の応答が返されます。

複数の認証メカニズムが有効になっている場合、`user_auth` パラメータでの順番によって決定されるとおりに、認証プロセスの順番が適用されることに注意してください。サーバーは各認証ソースを順番に検索し、ユーザーがそこに存在しない場合は、`tibemsd` によってユーザー名とパスワードが次のソースに渡されます。

たとえば、JAAS 認証の前にローカル認証が行われる場合、サーバーは提供されたユーザー名とパスワードを、まず `users.conf` ファイルで検索します。ユーザーがそこに存在しない場合、`tibemsd` はユーザー名とパスワードを `LoginModule` に渡し、これによって接続の試行が許可または拒否されます。

ユーザー名が `avogus` であるクライアントから接続要求が行われた場合を検討してみましょう。`avogus` が `users.conf` に存在する場合、EMS サーバーはそこで検出されたユーザー名とパスワードに基づいて、`avogus` へのアクセスを認証するか、拒否します。`avogus` が `users.conf` に存在しない場合にのみ、サーバーはユーザー名とパスワードを `LoginModule` に渡します。

### 拡張可能な認証の有効化

拡張可能な認証は、`tibemsd.conf` 設定ファイル内のパラメータを使用して、EMS サーバーで有効化されます。必要なパラメータは以下のとおりです。

- `authorization` : セキュアデスティネーションに対して、ユーザーの認証情報とアクセス権限を確認するようにサーバーに指示します。
- `user_auth` : 認証に `LoginModule` を使用するように、EMS サーバーに指示します。

- `jaas_classpath` : LoginModule によって使用される JAR ファイルと依存クラスを指定します。
- `jaas_config_file` : LoginModule を読み込む設定ファイル（通常は `jaas.conf`）を指定します。詳細は、284 ページの「[jaas.conf 設定ファイルの例](#)」を参照してください。

LoginModule は Java 仮想マシンで実行されるため、EMS サーバーで JVM を有効にする必要もあります。詳細は、292 ページの「[JVM の有効化](#)」を参照してください。

## 認証モジュールの作成

LoginModule は、EMS サーバーの JVM 内で実行するカスタムモジュールです。LoginModule は Sun Microsystems によって提供される API セットの一種である JAAS を使用して作成されます。これを使用して、プラグ可能な Java アプリケーションが作成されます。JAAS によって、作成したコードと EMS サーバーの間のインターフェイスを使用できるようになります。JAAS は JRE の標準パーツであり、EMS とともにインストールされます。

### LoginModule の要件

拡張可能な認証を実装するには、JAAS インターフェイスを実装する LoginModule を作成する必要があります。EMS サーバーで実行する LoginModule には、次のようないくつかの要件があります。

- LoginModule は、NameCallback コールバックと PasswordCallback コールバックを介して、EMS サーバーからユーザー名とパスワードを受け取る必要があります。EMS サーバーは、これらのコールバックを使用し、`prompt` 引数は無視して、LoginModule にユーザー名とパスワードを渡します。
- ユーザー名とパスワードの組み合わせが無効な場合、LoginModule は `FailedLoginException` をスローする必要があります。これにより、EMS サーバーは対応する接続の試行を拒否します。
- LoginModule はスレッドセーフである必要があります。つまり、LoginModule はマルチスレッド環境と、シングルスレッド環境の両方で機能できる必要があります。
- LoginModule は、ユーザー名とパスワードの組み合わせが有効かどうかを判別することによってのみ、認証を実行する必要があります。カスタムのアクセス権限についての詳細は、285 ページの「[拡張可能なアクセス権限](#)」を参照してください。
- アクセス権限モジュールと同様に、LoginModule は時間のかかる操作を実行してはならず、値を即時に返す必要があります。これらのモジュールは EMS サーバーのメッセージ処理プロセスの一部になるため、操作に遅延が生じるとパフォーマンスに重大な影響が及ぼされる可能性があります。
- LoginModule には、`EMSUserAuthentication` という名前が付けられる必要があります。

JAAS クラスやインターフェイスのドキュメントなど、JAAS の詳細については <http://java.sun.com/products/jaas/> で入手できます。

### EMS サーバーで の LoginModule の読み込み

EMS サーバーは、`tibemsd.conf` ファイルにある `jaas_config_file` パラメータによって指定された設定ファイルの内容に基づいて、`LoginModule` を検出し、読み込みます。通常、JAAS 設定ファイルには `jaas.conf` という名前が付けられています。このファイルには、`LoginModule` の起動に使用される設定情報が含まれています。

`jaas.conf` ファイルの内容は、以下のサイトで説明されているように、JAAS 設定構文に従う必要があります。

<http://java.sun.com/j2se/1.5.0/docs/api/javax/security/auth/login/Configuration.html>



JAAS 設定ファイル内の `LoginModule` は、`EMSUserAuthentication` という名前である必要があります。

### jaas.conf 設定ファイルの例

```
EMSUserAuthentication {  
  com.tibco.tibems.tibemsd.security.example.FlatFileUserAuthLoginModule  
  required debug=true filename=jaas_users.txt;  
};
```

## 拡張可能なアクセス権限

拡張可能なアクセス権限機能は Java 仮想マシン (JVM) と Java Access Control Interface (JACI) を使用して、ユーザー独自の Java ベースのアクセス権限モジュールを EMS サーバーで実行できるようにします。

アクセス権限モジュールは EMS サーバー内の JVM で実行され、JACI インターフェイスを使用して `tibemsd` に接続します。LoginModule の場合と同様に、アクセス権限モジュールは EMS アプリケーションのセキュリティレベルを高めません。アクセス権限を付与するための標準の EMS プロシージャは置き換えられません。代わりに、このモジュールは既存のプロセスを増やします。

ユーザーがトピックのサブスクライブやメッセージのパブリッシュなど、何らかのアクションを実行しようとする、EMS サーバーは `acl.conf` ファイル、アクセス権限モジュール、および以前のアクセス権限モジュールによる照会でキャッシュされた結果を確認して、認可を行います。このプロセスは、286 ページの「[アクセス権限を付与する方法](#)」で詳しく説明されています。

### キャッシュされたアクセス権限

認可プロセスの速度を上げるために、EMS サーバーはアクセス権限モジュールから受け取った応答を *allow cache* と *deny cache* という 2 種類のプールにキャッシュします。アクセス権限モジュールを起動する前に、まず、サーバーはこれらのキャッシュにユーザーの要求と一致するキャッシュエントリがないかどうかを調べます。

**キャッシュとは** 各キャッシュエントリは、ユーザー名とアクション、およびアクセス権限モジュールからの認可結果の応答によって構成されています。

- ユーザー名は固有です。キャッシュされたアクセス権限はそのユーザーにのみ適用されます。
- アクションも固有です。ひとつのアクションのみが各キャッシュエントリに含まれます。認可が必要なアクションは、`acl.conf` ファイルにリストされているものと同じです。
- デスティネーションにはワイルドカードを含めることができます。つまり、単一のキャッシュエントリを使用して、複数のデスティネーションでアクションを実行するためのユーザーの認可を決定できます。

アクセス権限モジュールからの応答によってアクションが認可された場合、そのアクセス権限は *allow cache* にキャッシュされます。アクションが拒否された場合、アクセス権限は *deny cache* にキャッシュされます。

### アクセス権限 がキャッシュさ れる期間

アクセス権限モジュールの結果にはタイムアウト設定も含まれており、これによってキャッシュエントリが期限切れになるまでキャッシュに保持される期間が決定されます。タイムアウトが期限切れになると、エントリはキャッシュから削除されます。これらのタイムアウトはアクセス権限モジュールによって割り当てられるため、ユーザーはアクセス権限モジュールが呼び出される頻度を制御できます。これにより、EMS サーバーに課せられる負荷も制御できます。



アクセス権限キャッシュエントリのタイムアウトを長く設定すると、パフォーマンスが高まる可能性があります。同時に、アクセス権限の変更に対するシステムの即応性が下がる場合があります。アクセス権限モジュールを作成するときは、タイムアウトの長さについて慎重に検討してください。

### キャッシュの管理

キャッシュ統計を表示してリセットしたり、すべてのキャッシュエントリをクリアしたりすることができます。次のコマンドは、管理ツールで使用できます。

- [132 ページの「jaci showstats」](#)
- [132 ページの「jaci resetstats」](#)
- [132 ページの「jaci clear」](#)

## アクセス権限を付与する方法

EMS クライアントがアクセス権限を必要とするアクションを実行しようとする時、EMS サーバーは次の場所を順番に確認します。

1. まず、サーバーは `acl.conf` に認可があるかどうかを確認します。これは、アクセス権限を付与するための標準的な EMS メカニズムです（255 ページの[第 8 章、「認証とアクセス権限」](#)を参照）。
2. 次に、サーバーはアクセス権限モジュールの `allow cache` に認可があるかどうかを確認します。ユーザー名、アクション、およびデスティネーションと一致するエントリがこのキャッシュ内にある場合、その要求は許可されます。

ワイルドカードが含まれるデスティネーションがこのキャッシュ内に存在する可能性があるため、要求されたデスティネーションを含むワイルドカードデスティネーションがエントリに含まれている場合があります。そのエントリが同じユーザー名とアクションを指定している場合、要求は許可されます。このトピックについての詳細は、以下の[アクセス権限にワイルドカードを使用した場合の影響](#)を参照してください。

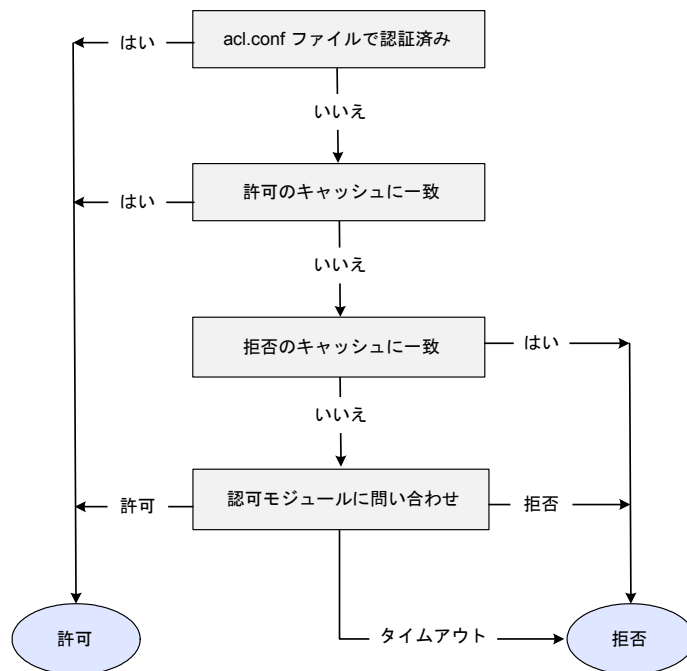
3. 次に、サーバーは `deny cache` に一致するエントリがあるかどうかを調べます。`deny cache` にエントリがある場合、要求は拒否されます。

`allow cache` の場合と同様に、デスティネーションにワイルドカードが使用されている場合、要求されているデスティネーションが含まれているデスティネーションがキャッシュエントリに含まれている可能性があります。そのエントリがユーザー名およびアクションと一致する場合、要求は拒否されます。このトピックについての詳細は、以下の[アクセス権限にワイルドカードを使用した場合の影響](#)を参照してください。

4. 最後に、どちらのキャッシュにも一致するエントリがない場合、サーバーはユーザー名、アクションタイプ、およびデスティネーションをアクセス権限モジュールに渡し、このモジュールが許可または拒否の認可応答を返します。また応答は、その応答に指定されているタイムアウトの期間、キャッシュに保存されます。アクセス権限モジュールが、`tibemsd.conf` ファイル内の `jaci_timeout` パラメータに指定されているタイムアウトに到達するまでに要求に応答しない場合、デフォルトでサーバーは認可を拒否します。

アクセス権限が必要なアクションは、`acl.conf` ファイルにリストされているものと同じです。たとえば、トピックへのサブスクライブやキューへのパブリッシュなどです。アクセス権限は、[229 ページの「acl.conf」](#) で説明されています。[図 17](#) は、アクセス権限を付与または拒否するときにサーバーが使用するデシジョンツリーを示しています。

図 17 アクセス権限のデシジョンツリー



通常、アクセス権限はクライアントが操作を開始するときに確認されます。ブラウジング要求の場合、サーバーはブラウジング操作の間、ある特定のポイントでのみアクセス権限をレビューすることを知っておくと便利です。

サーバーは、クライアントがキューのブラウズを開始したときと、クライアントがブラウズ可能なメッセージのリストを更新する必要があるときは常に、ブラウジングアクセス権限を確認します。クライアントは、ブラウジングを初めて開始したときに、サーバーからメッセージのリストを受け取ります。サーバーは、クライアントが現在のリストの最後までブラウズしたときは必ず、リストを更新し、アクセス権限を再確認します。

**持続サブスクライバ**      持続サブスクライバが EMS サーバーから切断されると、そのサーバーはクライアントのメッセージを蓄積し続けます。ただし、クライアントが切断されている間は、持続サブスクライバに関連付けられるユーザーはいません。このため、サーバーは、クライアントが接続されていないときに受け取られたメッセージのアクセス権限を即時に確認することはできません。

ユーザーが後でサーバーに再接続し、持続サブスクリプションを再サブスクライブすると、サーバーは、そのサブスクライブ操作自体のアクセス権限は確認しますが、バックログ内のすべてのメッセージは追加のアクセス権限チェックを受けずに、コンシューマに配信されます。

**特別な状況**      要求が `acl.conf` ファイルで完全には一致しない場合でも、アクセス権限用のキャッシュまたはアクセス権限モジュールのいずれかを参照することなくその要求が拒否されるという特別な状況があります。`acl.conf` ファイルで次の条件が満たされる場合、すべての要求は拒否されます。

- ユーザー名は存在するが、どのデスティネーションにも関連付けられていない。
- ユーザー名が存在し、いくつかのデスティネーションに関連付けられているが、要求内の特定のデスティネーションには関連付けられていない。
- ユーザー名がグループに含まれているが、そのグループがどのデスティネーションにも関連付けられていない。
- ユーザー名がグループに含まれており、そのグループがいくつかのデスティネーションに関連付けられているが、要求内の特定のデスティネーションには関連付けられていない。

通常、`acl.conf` ファイル内のエントリはアクセス権限モジュール内のエントリより優先されます。これにより、適切に定義された静的ケースで、アクセス権限チェックを最適化できるようになります。ユーザーに関する記述が `acl.conf` ファイルにない場合は、アクセス権限モジュールによってアクセス権限に関するすべての処理が行われます。

## アクセス権限にワイルドカードを使用した場合の影響

デスティネーション名にワイルドカードを入れることで、アクセス権限モジュールからのアクセス権限結果を使用して、さまざまなデスティネーションでアクションを実行するためのユーザー認可を許可または拒否できます。たとえば、アプリケーションがユーザー `mw Walton` にトピック `foo.bar.1` にパブリッシュさせようとする場合でも、アクセス権限モジュールでは、ユーザー `mw Walton` に、トピック `foo.bar.*` にメッセージをパブリッシュするためのアクセス権限を付与できます。この認可がキャッシュされている限り、`mw Walton` はトピック `foo.bar.baz` および `foo.bar.bo` へのパブリッシュも行えます。これは、これらの両方のトピックに `foo.bar.*` が含まれているからです。

デスティネーションで何らかのアクションを実行するためのアクセス権限が `allow cache` にキャッシュされている限り、アクセス権限モジュールによって使用されている外部システムでアクセス権限が取り消されたとしても、ユーザーによるそのアクションの実行は認可されます。このアクセス権限は、ワイルドカードの使用により認可されたデスティネーションに含まれている、すべてのデスティ



ネーションにも適用されます。EMS サーバーは `allow cache` にアクセス権限がないかを調べた後、`deny cache` を調べ、その後 `uncached` アクセス権限要求をアクセス権限モジュールに送信します。つまり、キャッシュエントリのタイムアウトが期限切れになり、エントリがキャッシュから削除されるまで、認可ステータスを変更することはできません。

同様に `deny cache` 内のエントリは、タイムアウトが期限切れになり、エントリが削除されるまでそのキャッシュ内に留まります。このときになって初めて、EMS サーバーは要求をアクセス権限モジュールに送信し、ステータスの変更を有効にできるようになります。

重複するワイルドカードを使用すると、状況はさらに複雑になる可能性があります。たとえば、次の三つのデスティネーションの場合を検討してみましょう。

```
foo.*.baz
foo.bar.*
foo.>
```

`foo.*.baz` がキャッシュ内にある場合、`foo.bar.*` がそれと一致し、そのデスティネーションのアクセス権限がそのキャッシュから送信されるように思えます。しかし実際には、`foo.bar.*` は `foo.*.baz` と交差するだけで、そのサブセットではないため、このキャッシュエントリによってアクセス権限を決定することはできません。つまり、`foo.bar.*` と一致するすべてのデスティネーションが `foo.*.baz` ととも一致するとは限りません。たとえば、デスティネーション `foo.bar.booo` は `foo.bar.*` を設定した場合はアクセス権限を付与されますが、`foo.*.baz` を設定した場合はアクセス権限を付与されません。

`foo.bar.*` が一致するすべてのデスティネーションが `foo.*.baz` ととも一致するわけではないため、`foo.*.baz` は `foo.bar.*` と交差すると見なされます。キャッシュエントリは、要求されたデスティネーションが単なる交差ではなく、キャッシュエントリのサブセットである場合に、アクセス権限を決定できます。交差である場合、アクセス権限をそのキャッシュが決定することはできません。

一方、デスティネーション `foo.>` には、サブセットとして `foo.bar.*` と `foo.*.baz` の両方が含まれます。これは、`foo.bar.*` または `foo.*.baz` のいずれかと一致するすべてのデスティネーションが `foo.>` ととも一致するからです。`foo.>` がキャッシュ内にある場合、アクセス権限はそのキャッシュによって決定されます。

## 拡張可能なアクセス権限の有効化

拡張可能なアクセス権限は、`tibemsd.conf` 設定ファイル内のパラメータを使用して、EMS サーバーで有効化されます。必要なパラメータは以下のとおりです。

- `authorization` : 認可を有効にします。
- `jaci_class` : アクセス権限モジュールを実装するクラスを指定します。
- `jaci_classpath` : アクセス権限モジュールによって使用される JAR ファイルと依存クラスを指定します。

アクセス権限モジュールを使用してアクセス権限が付与されるのは、`topics.conf` および `queues.conf` の設定ファイルでセキュアであると定義されたデスティネーションのみです。`secure` プロパティを含むトピックまたはキューがない場合、サーバーはアクセス権限をまったく確認しないため、アクセス権限モジュールが呼び出されることはありません。

アクセス権限モジュールは Java 仮想マシンで実行するため、EMS サーバーで JVM を有効にする必要もあります。詳細は、292 ページの「[JVM の有効化](#)」を参照してください。

## アクセス権限モジュールの作成

アクセス権限モジュールは、EMS サーバーの JVM 内で実行するカスタムモジュールです。アクセス権限モジュールは、TIBCO Software Inc. によって開発された API セットの一種である JACI を使用して作成されます。これを使用して、EMS クライアント要求を認可する Java モジュールを作成できます。JACI によって、作成したコードと EMS サーバーの間のインターフェイスを使用できるようになります。JACI は EMS の標準コンポーネントです。JACI クラスとインターフェイスについては、`com.tibco.tibems.tibemspd.security` で説明されています。

**要件** 拡張可能なアクセス権限を実装するには、JACI インターフェイスを実装するアクセス権限モジュールを作成する必要があります。EMS サーバーで実行するアクセス権限モジュールには、次のようないくつかの要件があります。

- アクセス権限モジュールは、認可対象の操作に関する情報を受け取る JACI `Authorizer` インターフェイスを実装する必要があります。
- アクセス権限モジュールは、`AuthorizationResult` クラスを介してアクセス権限結果を返す必要があります。アクセス権限結果には、次が含まれます。
  - `allowed` パラメータ。`true` の場合は要求が許可され、`false` の場合は要求が拒否されます。
  - タイムアウト。これによって、アクセス権限結果がキャッシュ内に留まる時間が決定します。結果は、最大 24 時間までキャッシュされるか、まったくキャッシュされないように設定できます。
  - ユーザーがアクションの実行を認可されるデスティネーション。返されるデスティネーションは、要求より包含的になる可能性があります。たとえば、ユーザーがトピック `foo.bar` をサブスクライブするように要求した場合、アクセス権限結果によってユーザーが `foo.*` をサブスクライブできるようにすることができます。デスティネーションがアクセス権限結果に含まれていない場合、許可または拒否の応答は最初に要求されたデスティネーションに限定されます。
  - アクセス権限結果の応答先となるアクションタイプ。たとえば、デスティネーションにパブリッシュする認可や、キューからメッセージを受け取る認可などです。アクセス権限は、`foo.>` に対するパブリッシュやサブスクライブを行うためのアクセス権限など、複数のアクションタイプに付与することができます。EMS サーバーは、結果に指定されたアクションごとにひとつのキャッシュエントリを作成します。

- アクセス権限モジュールは、スレッドセーフである必要があります。つまり、アクセス権限モジュールはマルチスレッド環境と、シングルスレッド環境の両方で機能できる必要があります。
- `LoginModule` と同様に、アクセス権限モジュールは時間のかかる操作を実行してはならず、値を即時に返す必要があります。これらのモジュールは EMS サーバーのメッセージ処理プロセスの一部になるため、操作に遅延が生じるとパフォーマンスに重大な影響が及ぼされる可能性があります。

JACI クラスやインターフェイスのドキュメントについては、[com.tibco.tibems.tibemsd.security](http://com.tibco.tibems.tibemsd.security) で説明されています。

## EMS サーバー内の JVM

---

Java 仮想マシン (JVM) は Java プラットフォーム上の仮想マシンであり、EMS サーバーの内部で実行できます。個々の Java モジュールの選択を JVM で行って、EMS サーバーにプラグインすることができます。JVM では、次の TIBCO Enterprise Message Service 機能を使用する必要があります。

- 拡張可能なセキュリティ : 279 ページの [第 9 章、「拡張可能なセキュリティ」](#) を参照してください。
- データベースストア : 293 ページの [第 10 章、「データベースストアの使用」](#) を参照してください。

### JVM の有効化

Java 仮想マシンは、[tibemsd.conf](#) 設定ファイル内のパラメータを使用して、EMS サーバーで有効化されます。JVM を有効化および設定するパラメータは、次のとおりです。

- [jre\\_library](#) : JVM を有効にします。
- [jre\\_option](#) : 開始時に、Sun Microsystems によって定義された標準 JVM オプションを JVM に渡せるようにします。

これらのパラメータについての詳細は、226 ページの [「JVM パラメータ」](#) を参照してください。

## 第 10 章

## データベースストアの使用

この章では、データベース内にメッセージが保存されるように TIBCO Enterprise Message Service サーバーを設定する方法を説明します。この章では、データベースストアについてのみ説明します。ファイルベースのストアについての詳細は、29 ページの「[複数ストアへのメッセージの保存](#)」を参照してください。



オプションのデータベースストア機能では、Java 用の **Hibernate Core** と関連する jar ファイルをインストールして使用する必要があります。

---

トピック

- 「[データベースストアの概要](#)」、294 ページ
- 「[データベースストアの設定](#)」、295 ページ
- 「[EMS Schema Export Tool](#)」、302 ページ

## データベースストアの概要

---

EMS サーバーはデータベースに接続し、ひとつ以上のデータベースインスタンスにメッセージを保存できます。サーバーは Java 用の **Hibernate Core** を使用してデータベースに接続し、データベースと EMS サーバーの間のインターフェイスを取ります。

**要件** データベースストアを作成するには、次を用意する必要があります。

- Java 用の **Hibernate Core** と関連する jar ファイル。

Java 用の **Hibernate Core** は、TIBCO Enterprise Message Service 製品の配布とともに、[download.tibco.com](http://download.tibco.com) から入手できます。

- **Hibernate** によってサポートされているデータベースサーバー、対応するダイアレクト、および該当する **JDBC** ドライバ。

データベースサーバーが実行している必要があります。また、EMS サーバーが接続するデータベースは、データベース管理者によって事前に作成されている必要があります。

- データベースサーバーに対する読み取りおよび書き込みアクセス権限を持つユーザー名とパスワード。

## データベースストアの設定

このセクションでは、データベースストアの設定と実装の手順について説明します。複数ストア機能の概念に関する一般的な情報については、29 ページの「[複数ストアへのメッセージの保存](#)」を参照してください。

データベースストアを作成および設定するための設定は、EMS サーバーで管理されており、クライアントに対して透過的です。データベースストア機能を設定するには、次の手順を実行してください。

1. `tibemspd.conf` で以下のパラメータを設定し、データベースストア機能を有効にします。

- `dbstore_classpath`
- `dbstore_driver_name`
- `dbstore_driver_dialect`
- `jre_library`

`dbstore` パラメータについての詳細は、[tibemspd.conf の設定](#)を参照してください。EMS サーバーで JVM を有効にする `jre_library` パラメータについての説明は、226 ページの「[JVM パラメータ](#)」を参照してください。

2. `stores.conf` ファイルにストアをセットアップおよび設定します。

複数のデータベースストア、またはデータベースストアとファイルベースストアの組み合わせを作成できます。各ストアには一意の名前が必要です。パラメータを使って、ストアがデータベースストアかどうかを判定し、データベースサーバーの場所を指定し、EMS サーバーがデータベースへのアクセスに使用するユーザー名およびパスワードを指定します。

データベースストアのパラメータのリストについては、以下の [stores.conf の設定](#)を参照してください。

3. 設定されたストアにデスティネーションを関連付けます。

メッセージは、そのデスティネーションに従って他のストアに送信されます。`topics.conf` ファイルと `queues.conf` ファイルで `store` パラメータを使用し、デスティネーションを特定のストアに関連付けます。ストアの関連付けの変更には、EMS 管理ツールの `setprop topic` コマンドまたは `setprop queue` コマンドを使うこともできます。

同一のストアに複数のデスティネーションを明示的にまたはワイルドカードを使用してマッピングできます。ストアが設定されていない場合でも、サーバーはストアに関連付けられていない永続的メッセージをデフォルトのストアに送信します。詳細は、[デフォルトのストアファイル](#)を参照してください。

`store` パラメータの詳細は、69 ページの「[store](#)」を参照してください。

#### 4. データベーステーブルをエクスポートします。

EMS サーバーでメッセージをデータベースに保存するよう設定するときは、サーバーの起動前にデータベーススキーマをエクスポートする必要があります。データベーステーブルの作成、ドロップ、更新には EMS Schema Export Tool を使います。

詳細は、302 ページの「[EMS Schema Export Tool](#)」を参照してください。

## tibemsd.conf の設定

これらのパラメータは、[tibemsd.conf](#) 設定ファイルで設定します。

### dbstore\_classpath

`dbstore_classpath = pathname`

データベースストア機能を利用するときに EMS サーバーに必要なすべての JAR ファイルを含めることができます。タイプが `dbstore` のストアを [stores.conf](#) ファイルで作成したときは、このパラメータを必ず設定する必要があります。

必要な JAR ファイルは、どの Hibernate リリースがインストールされているかによって決まります。詳細は、Hibernate ディストリビューションの `lib/` ディレクトリにある `README.txt` ファイルに記載されています。JAR ファイルの多くはバージョン固有のため、新しい Hibernate がリリースされると必要なバージョンが変わる可能性があります。必要なバージョンを確認し、それに応じて `dbstore_classpath` 変数を修正してください。

例として、Hibernate のリリース 3.2.5 を使用している場合は、以下の JAR ファイルへのパスを `dbstore_classpath` に含める必要があります。

- `hibernate3.jar`
- `dom4j-1.6.1.jar`
- `commons-collections-2.1.1.jar`
- `commons-logging-1.0.4.jar`
- `ehcache-1.2.3.jar`
- `jta.jar`
- `cglib-2.1.3.jar`
- `antlr-2.7.6.jar`
- `c3p0-0.9.1.jar`
- `asm.jar`
- `asm-attrs.jar`
- データベース固有の JAR ドライバファイル。サポートされている jar ファイルのリストについては、TIBCO Enterprise Message Service のインストールの Database Servers and Drivers を参照してください。



例については、`EMS_HOME/samples/config/tibemsd-db.conf` を参照してください。

### dbstore\_driver\_name

```
dbstore_driver_name = name
```

Hibernate で使用する JDBC ドライバの名前を指定します。

例：

- MySQL InnoDB データベースサーバーを使用する場合  
`dbstore_driver_name=com.mysql.jdbc.Driver`
- Microsoft SQL Server を使用する場合  
`dbstore_driver_name=com.microsoft.sqlserver.jdbc.SQLServerDriver`
- Oracle 10g を使用する場合  
`dbstore_driver_name=oracle.jdbc.driver.OracleDriver`
- IBM DB2 Server を使用する場合  
`dbstore_driver_name=com.ibm.db2.jcc.DB2Driver`

### dbstore\_driver\_dialect

```
dbstore_driver_dialect = dialect
```

SQL コマンドの構文を決定する Hibernate の SQL ダイアレクトを指定します。

たとえば、MySQL を InnoDB データベースサーバーで使用する場合は次のとおりです。

```
dbstore_driver_dialect = org.hibernate.dialect.MySQL5InnoDBDialect
```

SQL ダイアレクトは Hibernate で定義されています。データベースとダイアレクトの対応については、Hibernate のインストールディレクトリのアーカイブにある `readme.txt` ファイルを参照してください。

## stores.conf の設定

このセクションでは、`stores.conf` ファイルで各データベースストア用に設定するパラメータについて説明します。`stores.conf` には、データベースとファイルベースストアの両方の定義が含まれます。ファイルベースストアの設定についての詳細は、[243 ページの「stores.conf」](#) を参照してください。

ファイルは、以下の形式で記述します。

```
[store_name] # mandatory -- square brackets included.
type = dbstore
dbstore_driver_url = JDBCURL
dbstore_driver_username = username
dbstore_driver_password = password
[processor_id = processor-id]
```

表 49 データベースストアファイルのパラメータ

パラメータ名	説明
[store_name]	<p>[store_name] は、このストア設定を識別する名前です。</p> <p>角かっこ [ ] は、store_name がオプションであることを示しているわけではありません。名前は角かっこで囲まれている必要があります。</p>
type=dbstore	<p>ストアタイプを識別します。このパラメータはすべてのストアタイプで必須です。type には次のいずれかを指定できます。</p> <ul style="list-style-type: none"><li>file : ファイルベースのストア</li><li>mstore : mstore</li><li>dbstore : データベースストア</li></ul> <p>ファイルベースストアの設定に使用するパラメータについての詳細は、<a href="#">243 ページの「stores.conf」</a>を参照してください。</p>
dbstore_driver_url	<p>データベースサーバーの場所を指定します。URL は、使用するデータベースの JDBC ドライバで指定されている構文を使って入力します。</p> <p>詳細は、使用する JDBC ドライバのマニュアルを参照してください。Oracle RAC データベースを使用する場合は、<a href="#">301 ページの「TAF を設定した URL の使用」</a>も参照してください。</p>
dbstore_driver_username	<p>データベースへのアクセスに使用する EMS サーバーのユーザー名。</p> <p>このユーザーには、データベースへの読み書き権限が必要です。</p>

表 49 データベースストアファイルのパラメータ

パラメータ名	説明
dbstore_driver_password	<p>サーバーがデータベースにアクセスするために、dbstore_driver_username で指定されたユーザー名と一緒に使用するパスワードです。</p> <p>パスワードは、他のパスワードと同様、tibemsadmin ツールを使用して解読不能にできます。tibemsadmin を使ってパスワードを解読不能にする方法についての詳細は、118 ページの表 15、「tibemsadmin のオプション」を参照してください。</p>
processor_id	<p>このパラメータを指定した場合、EMS サーバーは指定したプロセッサにこのストアのストレージスレッドをバインドします。</p> <p>デフォルトの動作で十分なスループットが得られる場合は、このパラメータは使用しません。ストアにプロセッサ ID が指定されていない場合、そのストアは特定のプロセッサにはバインドされません。</p> <p>processor-id は整数で指定します。プロセッサ ID の番号は 0 で始まり、利用可能なプロセッサ数から 1 を差し引いた数まで続きます。たとえば、四つのプロセッサがある場合、利用可能なプロセッサ ID は 0、1、2、3 です。</p> <p>このパラメータには、tibemsd.conf の <a href="#">processor_ids</a> パラメータと同様の要件、制限、および利点があります。使用上のガイドラインについては、115 ページの「パフォーマンス調整」を参照してください。</p>

## MySQL サーバーを使用する例

```
[ $sys.failSAFE]
  type=dbstore
  dbstore_driver_url=jdbc:mysql://mysqlsrv_1:3306/sysfs
  dbstore_driver_username=admin
  dbstore_driver_password=admin123

[ $sys.meta]
  type=dbstore
  dbstore_driver_url=jdbc:mysql://mysqlsrv_1:3306/sysmeta
  dbstore_driver_username=admin
  dbstore_driver_password=admin123
```

### Microsoft SQL Server を使用する例

```
[ $sys.meta]
  type=dbstore
  dbstore_driver_url=jdbc:sqlserver://sqlsrv_1:3415;databaseName=sysmeta
  dbstore_driver_username=admin
  dbstore_driver_password=admin123

[ $sys.fail-safe]
  type=dbstore
  dbstore_driver_url=jdbc:sqlserver://sqlsrv_1:3415;databaseName=sysfs
  dbstore_driver_username=admin
  dbstore_driver_password=admin123
```

### Oracle 10g を使用する例

```
[ $sys.meta]
  type=dbstore
  dbstore_driver_url=jdbc:oracle:thin:adminmeta/admin123@osrv_1:1521:orclperf
  dbstore_driver_username=adminmeta
  dbstore_driver_password=admin123

[ $sys.fail-safe]
  type=dbstore
  dbstore_driver_url=jdbc:oracle:thin:adminfs/admin123@osrv_1:1521:orclperf
  dbstore_driver_username=adminfs
  dbstore_driver_password=admin123
```

### Oracle RAC 10g を使用する例

```
[ $sys.fail-safe]
  type=dbstore
  dbstore_driver_url=jdbc:oracle:oci:<user>/<passwd>@(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP)(HOST=<host1>)(PORT=1521))(ADDRESS=(PROTOCO
  L=TCP)(HOST=<host2>)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=orcl)(
  FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASIC)(RETRIES=180)(DELAY=5))))
  dbstore_driver_username=admin
  dbstore_driver_password=admin123
```

詳細は、以下の [Oracle RAC データベースの設定](#) を参照してください。

### IBM DB2 Server を使用する例

```
[ $sys.meta]
  type=dbstore
  dbstore_driver_url=jdbc:db2://db2srv_1:50000/SYSMETA
  dbstore_driver_username=admin
  dbstore_driver_password=admin123

[ $sys.fail-safe]
  type=dbstore
  dbstore_driver_url=jdbc:db2://db2srv_1:50000/SYSFS
  dbstore_driver_username=admin
  dbstore_driver_password=admin123
```

## Oracle RAC データベースの設定

TIBCO Enterprise Message Service サーバーを Oracle RAC 10g または 11g データベースに接続するには、Oracle JDBC OCI ドライバと TAF 設定を使用する必要があります。

### OCI ドライバのインストール

ドライバには、Oracle Instant Client の使用をお勧めします。Oracle Instant Client は、Oracle が提供する最適化済みの軽量 OCI ドライバパッケージです。

<http://www.oracle.com/technology/tech/oci/instantclient/index.html>

Oracle Instant Client のインストールは指定の手順に従ってください。

### TAF を設定した URL の使用

データベースのフェイルオーバー中に EMS サーバーからデータベースへの接続が失われるようにするには、サーバーからデータベースへの接続に透過的アプリケーションフェイルオーバー（TAF: Transparent Application Failover）が設定された URL を使用する必要があります。例：

```
jdbc:oracle:oci:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host1)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=orcl)(FAILOVER_MODE=(TYPE=SELECT)(METHOD=BASIC)(RETRIES=180)(DELAY=5))))
```



完全な TAF はサポートされていません。EMS サーバーがトランザクション処理を実行中にデータベースのフェイルオーバーが発生すると、EMS サーバーは失敗したトランザクションを再実行または再起動しません。ただし、フェイルオーバーが発生した時点で処理中のトランザクションがない限りは、TAF 接続によって EMS サーバーを完全に回復できます。

## EMS Schema Export Tool

EMS サーバーで使用する各データベースストアには、データベースを指す設定パラメータが含まれます。EMS Schema Export Tool は、データベースストア用にデータベーステーブルの作成とエクスポートを行うためのツールです。データベース管理者は Schema Export Tool を使ってスキーマを選択的にエクスポートし、環境内のデータベースシステムとメッセージングシステムに合わせてスキーマを調整できます。



ひとつ以上のデータベースストアを設定するときは、データベーステーブルのエクスポートに EMS Schema Export Tool を使用する必要があります。言い換えれば、タイプが `dbstore` のストアを設定するときは、EMS サーバーの起動前に、データベーススキーマをエクスポートする必要があります。

Schema Export Tool は、`tibemsd` と同じディレクトリにある `tibemsd_util.jar` という名前の JAR ファイルです。コマンドラインオプションの説明は 303 ページの「表 50」に記載されています。データベーステーブルの作成またはドロップ、コンソールへの出力、ファイルへの保存、データベースへのエクスポートなどの実行はコマンドラインオプションで指定します。

Schema Export Tool を起動する前に、以下を実行する必要があります。

- EMS サーバーのグローバルデータベースストアのパラメータを設定します。グローバルデータベースストアを設定するパラメータは、`dbstore_` で始まります。これらのパラメータについての詳細は、296 ページの「[tibemsd.conf の設定](#)」を参照してください。
- タイプが `dbstore` のストアを少なくともひとつ設定してください。データベースストアの設定についての詳細は、297 ページの「[stores.conf の設定](#)」を参照してください。

### Schema Export Tool の動作原理

Schema Export Tool を起動すると、`tibemsd.conf` ファイルまたは `tibemsd.json` ファイルを使ってデータベースストアのパラメータが確認されます。続いて、`stores.conf` ファイルまたは JSON 設定ファイルで設定されているストアの解析が行われ、起動時に指定されたオプションによって、設定されているストアでタイプが `dbstore` のストア用に、データベーステーブルの作成、ドロップまたは更新が実行されます。

本ツールでは、選択したアクションをすべてのデータベースストアに対して実行したり、特定のストアに対してのみ実行したりできます。また、作成するデータベーステーブルのコンソールへの出力や、データベースまたは指定ファイルへのエクスポートも実行できます。

Schema Export Tool の実行

Schema Export Tool はコマンドラインから起動します。ツールが置かれているディレクトリから起動するか、tibems\_util.jar ファイルへの絶対パスを指定して起動できます。例：

```
Windows の場合
> java -jar EMS_HOME¥bin¥tibemsd_util.jar options

または
> java -jar c:¥tibco¥ems¥7.0¥bin¥tibemsd_util.jar options
```

```
Unix の場合
> java -jar EMS_HOME/bin/tibemsd_util.jar options

または
$ java -jar /opt/tibco/ems/7.0/bin/tibemsd_util.jar options
```

Schema Export Tool で使用するオプションを次の表に示します。

表 50 EMS Schema Export Tool のオプション

オプション	説明
-tibemsdconf <i>pathname</i>	<p>tibemsd.conf ファイルまたは tibemsd.json ファイルへの絶対パス。UNIX システムでの例は次のとおりです。</p> <p>/opt/tibco/ems/7.0/samples/config/tibemsd.conf</p> <p>本ツールは、Central Administraion のサポート対象プラットフォームで実行する場合のみ、JSON 設定ファイルをサポートします。サポート対象のプラットフォームのリストについては、『TIBCO Enterprise Message Service のインストール ガイド』の Central Administration を参照してください。</p> <p>テキストベースの tibemsd.conf ファイルはすべてのプラットフォームでサポートされています。</p>
-exporttofile	<p>store-name.ddl.log という名前のファイルにスキーマをエクスポートします。store-name はデータベースストアの名前です。複数のデータベースストアを設定している場合は、各データベースストアにひとつのファイルが作成されます。</p> <p>exporttofile オプションまたは export オプションのいずれも含まれていない場合は、コンソールにスキーマが出力されます。</p> <p>-exporttofile オプションと -export オプションの両方が含まれている場合は、両方の場所にデータベーススキーマがエクスポートされます。</p>

表 50 EMS Schema Export Tool のオプション

オプション	説明
-export	<p>ストアに設定されているデータベースにスキーマをエクスポートします。</p> <p><b>export</b> オプションまたは <b>exporttofile</b> オプションのいずれも含まれていない場合は、コンソールにスキーマが出力されます。</p> <p><b>-export</b> オプションと <b>-exporttofile</b> オプションの両方が含まれている場合は、両方の場所にデータベーススキーマがエクスポートされます。</p>
-store storename=create update drop	<p>ストア設定ファイルに指定したひとつ以上のストアについて、スキーマを作成、更新またはドロップします。</p> <p>すでに存在するスキーマで <b>create</b> オプションを選択すると、スキーマが再作成されます。</p> <p><b>create</b> はスキーマを画面に出力するのみで、実装しません。スキーマを実装するには、<b>export</b> または <b>exporttofile</b> を使う必要があります。</p>
-createall	<p>ストア設定ファイルで見つかったすべてのストアを作成します。このオプションは、新規ストアの作成時にすべての既存設定をドロップします。</p>
-dropall	<p>ストア設定ファイルで見つかったすべてのストアをドロップします。</p>
-updateall	<p>ストア設定ファイルで見つかったすべての設定済みストアのスキーマを更新します。</p>
-help	<p>Schema Export Tool とオプションに関する情報を出力してツールを終了します。</p>

例：

各種の設定で Schema Export Tool を使ってデータベーススキーマを作成する方法を以下の例に示します。

例 1： この例は、tibemsd\_util.jar への絶対パスを指定して任意のディレクトリから Schema Export Tool を起動する方法を示しています。

```
$ java -jar /opt/tibco/ems/7.0/bin/tibemsd_util.jar -help
```



例 2 : この例は、指定した `tibemsd-mssqlserver.conf` ファイルの `stores.conf` で見つかったすべてのストアのデータベーススキーマを作成してエクスポートします。

```
java -jar /opt/tibco/ems/7.0/bin/tibemsd_util.jar -tibemsdconf
/opt/tibco/ems/7.0/samples/config/tibemsd.conf -createall -export
```

例 3 : この例は、`sys.failSAFE` ストアのデータベーススキーマをデータベースにエクスポートします。

```
jar -jar /opt/tibco/ems/7.0/bin/tibemsd_util.jar -tibemsdconf
/opt/tibco/ems/7.0/samples/config/tibemsd.conf -export -store
¥$sys.failSAFE=createe
```

例 3 : この例は、`sys.failSAFE` ストアのデータベーススキーマを `sys.failSAFE.ddl.log` ファイルにエクスポートします。

```
$ jaav -jar /opt/tibco/ems/7.0/bin/tibemsd_util.jar -tibemsdconf
/opt/tibco/ems/7.0/samples/config/tibemsd.conf -exporttofile
-store ¥$sys.failSAFE=create
```

例 4 : この例は、`mystore1` ストアのデータベーススキーマを作成してエクスポートしますが、`mystore2` に対応するスキーマをドロップして変更点をエクスポートします。

```
java -jar /opt/tibco/ems/7.0/bin/tibemsd_util.jar -tibemsdconf
/opt/tibco/ems/7.0/samples/config/tibemsd.conf -store
mystore1=create -store mystore2=drop -export
```



## 第 11 章

## EMS クライアントアプリケーションの開発

この章では、Java、C、および C# の各プログラミング言語を使用した EMS クライアントアプリケーションの開発方法の概要を説明します。

トピック

---

- [「JMS 仕様」、308 ページ](#)
- [「サンプルクライアント」、311 ページ](#)
- [「プログラマー用チェックリスト」、312 ページ](#)
- [「接続ファクトリ」、323 ページ](#)
- [「EMS サーバーへの接続」、326 ページ](#)
- [「セッションの作成」、328 ページ](#)
- [「例外リスナーの設定」、329 ページ](#)
- [「トピックとキューの動的作成」、331 ページ](#)
- [「メッセージプロデューサの作成」、333 ページ](#)
- [「メッセージコンシューマの作成」、336 ページ](#)
- [「メッセージの操作」、341 ページ](#)

# JMS 仕様

EMS は JMS 1.1 仕様を実装しており、古いクライアント用に以前のバージョンである JMS 1.0.2b 仕様も実装しています。JMS 1.0.2b のインタフェースは引き続きサポートされますが、JMS 仕様の今後のリリースでは廃止される可能性があります。新しく開発するアプリケーションには JMS 1.1 インタフェースを採用し、古いインタフェースの使用はできるだけ早くうち切ることをお勧めします。

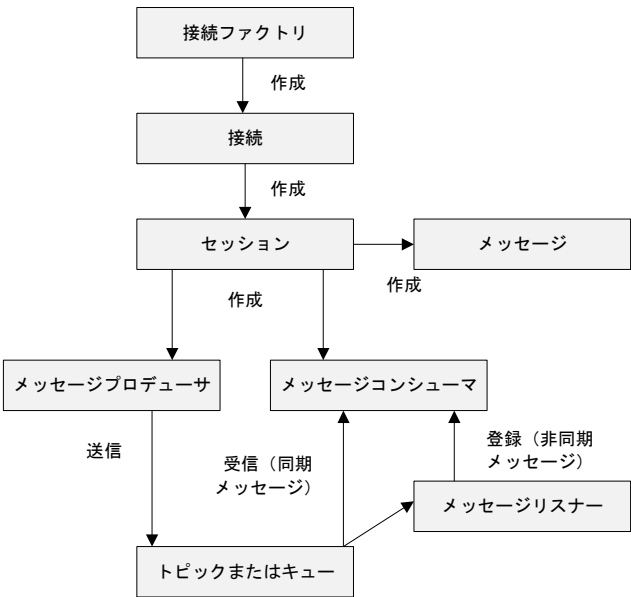
この章のコードサンプルでは、JMS 1.1 インタフェースを使用しています。

## JMS 1.1 仕様

JMS 1.1 仕様では、ポイントツーポイント（キュー）モデルとパブリッシュ / サブスクライブ（トピック）モデルを採用しているアプリケーションは、オブジェクトの作成に同じインタフェースを使用します。JMS 1.1 仕様では、これらのインタフェースを使ってトピックまたはキューのいずれかに使用可能なオブジェクトを作成するため、これらのインタフェースは、「共通基盤（Common Facilities）」と呼ばれています。

図 18 は、JMS API で使われているインタフェースを示しています。

図 18 JMS 1.1 プログラミングモデル

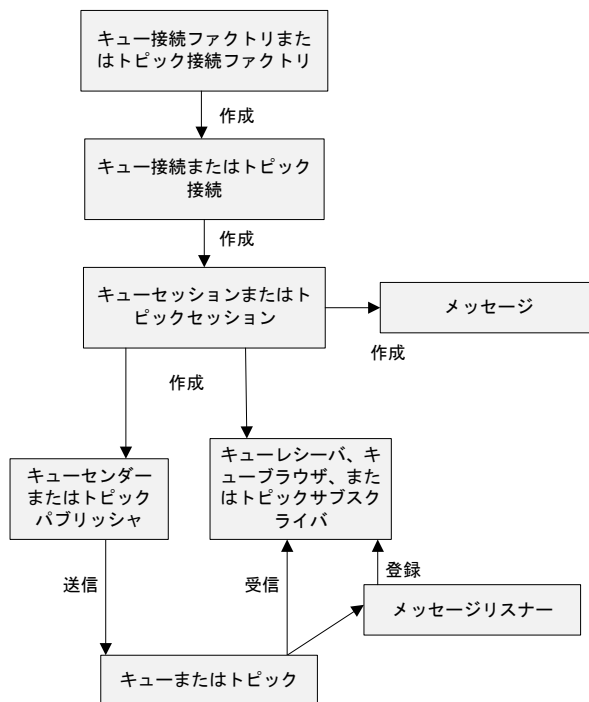


## JMS 1.0.2b 仕様

JMS 1.0.2b 仕様は、トピックとキューに対する特定のインタフェースを定義します。

JMS 1.0.2b インタフェースは JMS 1.1 共通基盤と同じ構造を持っていますが、インタフェースは、トピックとキューで、それぞれ固有です。図 19 は、JMS API で使われていた以前のインタフェースを示しています。

図 19 JMS 1.0.2b プログラミングモデル



JMS 1.1 および JMS 1.0.2b インタフェースの概要

表 51 に、JMS 1.1 共通基盤と JMS 1.0.2b 固有の両方について、それぞれをサポートするために JMS API で使われているインタフェースの概要を示します。

表 51 JMS API オブジェクトの概要（1 / 2 シート）

共通基盤インタフェース (JMS 1.1)	固有のインタフェース (JMS 1.0.2b)	説明
ConnectionFactory	QueueConnectionFactory TopicConnectionFactory	EMS サーバーへの接続の作成に使われるオブジェクト。
Connection	QueueConnection TopicConnection	プロバイダ（サーバー）との物理的接続をカプセル化する接続。セッションの作成に使用されます。
Session	QueueSession TopicSession	<p>セッションは、メッセージプロデューサ、メッセージコンシューマ、メッセージ、およびトランザクションされたメッセージグループのインスタンスを作成するシングルスレッドのオブジェクトです。</p> <p>セッションをトランザクション化することもできます。トランザクションセッションでは、メッセージグループは1回のトランザクションで送受信されます。</p>
MessageProducer	QueueSender TopicPublisher	メッセージプロデューサはセッションによって作成されるオブジェクトです。デスティネーションへのメッセージ送信に使用されます。
MessageConsumer	QueueBrowser QueueReceiver TopicSubscriber	メッセージコンシューマはセッションによって作成されるオブジェクトで、デスティネーションに送信されたメッセージを受信します。
MessageListener	MessageListener	メッセージリスナーは、メッセージの非同期イベントハンドラとして機能するオブジェクトです。メッセージリスナーは各 MessageConsumer に登録される必要があります。

表 51 JMS API オブジェクトの概要 (2 / 2 シート)

共通基盤インタフェース (JMS 1.1)	固有のインタフェース (JMS 1.0.2b)	説明
MessageSelector	MessageSelector	<p>メッセージセレクトは、アプリケーションで使用可能なオプションのフィルタです。フィルタリング作業をメッセージコンシューマではなく、メッセージプロバイダに転送します。</p> <p>1 つのメッセージセレクトは、単一の式を含む <b>String</b> です。式の構文は、<b>SQL92</b> 条件式構文のサブセットに基づきます。</p>
Message	Message	<p>タイプの異なるメッセージ本文をトピックとキューに利用できます。</p>
Queue	Queue	<p>メッセージの送受信が可能なデスティネーション。</p> <p>通常、デスティネーションはサーバー側で作成、管理しますが、クライアント側でも <b>Session</b> オブジェクトにメソッドを使用することでデスティネーションを動的に作成できます。</p>
Topic	Topic	

## サンプルクライアント

TIBCO Enterprise Message Service には、EMS のさまざまな機能が記述されているサンプルクライアントアプリケーションが含まれています。各機能についてこのマニュアルの説明を読む際に、これらのサンプルクライアントを参照することができます。

サンプルは、EMS のインストールディレクトリ内の *EMS\_HOME/samples/java*、*EMS\_HOME/samples/c*、および *EMS\_HOME/samples/cs* の各サブディレクトリに格納されています。それぞれのサブディレクトリには、サンプルクライアントのコンパイルおよび実行の方法を説明した **Readme** ファイルがあります。

EMS 環境のセットアップおよびサンプルクライアントの実行に関する概略的な手順については、87 ページの第 4 章、「基本操作」を参照してください。

## プログラマー用チェックリスト

このセクションでは、プログラミング言語ごとのチェックリストを使って、EMS アプリケーションの作成手順の概要を説明します。

- 312 ページの「[Java プログラマーのチェックリスト](#)」
- 313 ページの「[C プログラマーのチェックリスト](#)」
- 319 ページの「[C# プログラマーのチェックリスト](#)」

## Java プログラマーのチェックリスト

### インストール

- EMS ソフトウェアをインストールすると、自動的に EMS jar ファイルが `EMS_HOME/lib` サブディレクトリに格納されます。
- 以下の jar ファイルのフルパス名を `CLASSPATH` に追加します。  
`jms.jar`  
`tibjms.jar`
- 通信に SSL を使用する場合は、以下のファイルを `CLASSPATH` に追加します。  
`tibcrypt.jar`
- 非共有ステートフェイルオーバー API をプログラムで使用する場合は、以下のファイルを `CLASSPATH` に追加します。  
`tibjmsufo.jar`



このセクションで挙げた jar ファイルは、すべて TIBCO Enterprise Message Service のインストールディレクトリの `lib` サブディレクトリにあります。

EMS クライアントで **Entrust** を使用するには、**Entrust** ライブラリを別途購入し、インストールする必要があります。**Entrust** ライブラリを使用するには、`CLASSPATH` で **JSSE JAR** ファイルの「前に」それらのライブラリを指定しなければなりません。**JDK** で **Entrust** を使用するには、**Sun** の **Web** サイトから無制限の強度ポリシー（**unlimited strength policy**）の **JAR** ファイルをダウンロードし、ローカルの **JDK** インストールにインストールする必要があります。インストールと設定についての詳細は、**Entrust** のマニュアルを参照してください。

セキュアな実装に必要な条件についての詳細は、110 ページの、「セキュリティに関する注意事項」を参照してください。



## コーディング

EMS アプリケーションに以下のパッケージをインポートします。

```
import javax.jms.*;
import javax.naming.*;
```

## コンパイル

javac コンパイラを使って EMS アプリケーションをコンパイルし、.class ファイルを生成します。

例：

```
javac MyApp.java
```

MyApp.class ファイルが作成されます。

## 実行

java コマンドを使って、EMS .class ファイルを実行します。

例：

```
java MyApp
```

# C プログラマーのチェックリスト

EMS C プログラムの開発者は、開発サイクルの 5 つの各段階でこのチェックリストを使用できます。

## インストール

- EMS ソフトウェアをインストールすると、自動的に EMS クライアントライブラリ、バイナリ、ヘッダファイルが EMS\_HOME/lib サブディレクトリに格納されます。

## コーディング

アプリケーションプログラムの設定は、以下の手順に従います。

- EMS\_HOME/include をインクルードパスに追加します。(OpenVMS 環境の場合、インクルードパスは必要ないため、この手順はスキップしてください。)
- tibems.h ヘッダファイルをインクルードします。  
#include <tibems/tibems.h>
- C の Administration API をプログラムで使用する場合は、emsadmin.h ヘッダファイルをインクルードします。  
#include <tibems/emsadmin.h>

- 非共有ステートフェイルオーバー API をプログラムで使用する場合は、`tibufo.h` ヘッダファイルをインクルードします。  
`#include <tibems/tibufo.h>`
- `tibems_Open()` を呼び出して EMS C API の初期化を行い、完了時には、`tibems_Close()` を呼び出して EMS が使用していたメモリの割り当てを開放します。

## コンパイルとリンク

- ANSI 準拠の C コンパイラを使用してプログラムをコンパイルします。
- 必要な EMS C ライブラリファイルをリンクします (314 ページの「[ライブラリファイルのリンク](#)」を参照)。

詳細は、`samples/c/readme` ファイルを参照してください。

## 実行

- **UNIX**: EMS のダイナミックライブラリを UNIX プラットフォームで使用する場合、環境変数 `$LD_LIBRARY_PATH` に `EMS_HOME/lib` ディレクトリ (共有ライブラリファイルを含む) を含める必要があります (UNIX プラットフォームのバリエーションによっては、この変数は `$SHLIB_PATH` または `$SYLIB_LIBRARY_PATH` と呼ばれます)。
- **Windows**: `PATH` に `ems¥7.0¥bin` ディレクトリを含める必要があります。
- **OpenVMS**: インストール処理によって自動的に、EMS ダイナミックライブラリを使用するのに必要な共有可能イメージがインストールされます。
- **すべてのプラットフォーム**: アプリケーションは、EMS サーバーのプロセス (`tibemsd`) に接続できる必要があります。

## ライブラリファイルのリンク

EMS C プログラムでは、適切なライブラリファイルをリンクする必要があります。以下のセクションでは、各オペレーティングシステムのプラットフォームでリンクするファイルについて説明します。

- 315 ページの「[32 ビット UNIX](#)」
- 315 ページの「[64 ビット UNIX](#)」
- 316 ページの「[Microsoft Windows](#)」
- 318 ページの「[OpenVMS](#)」

## 32 ビット UNIX

32 ビット UNIX 環境では、共有ライブラリと静的ライブラリを利用できます。フォワードマイグレーションをスムーズに行うため、共有ライブラリを使用することをお勧めします。

表 52 32 ビット UNIX のリンカフラグ

リンカフラグ	説明
-ltibems	すべてのプログラムは、このライブラリフラグを使ってリンクする必要があります。
-lssl -lcrypto	SSL を使用するプログラムは、これらのライブラリフラグを使ってリンクする必要があります。
-lz	圧縮を使用するプログラムは、このライブラリフラグを使ってリンクする必要があります。
-ltibemslookup -lldap -lxml2 -llber	EMS LDAP 照合を使用するプログラムは、これらのライブラリフラグを使ってリンクする必要があります。
-ltibemsadmin	C の <b>Administration</b> ライブラリを使用するプログラムは、このライブラリフラグを使ってリンクする必要があります。
-ltibemsufo	非共有ステートフェイルオーバーライブラリを使用するプログラムは、このライブラリフラグを使ってリンクする必要があります。

## 64 ビット UNIX

64 ビット UNIX 環境では、共有ライブラリと静的ライブラリを利用できます。フォワードマイグレーションをスムーズに行うため、共有ライブラリを使用することをお勧めします。このリリースでは、64 ビットライブラリは、HP-UX、Solaris、AIX、および Linux (2.4 glibc 2.3) の各プラットフォームで利用できます。

64 ビットライブラリを使用するには、`TIBCO_HOME/ems/7.0/lib/64` をライブラリパスにインクルードし、ライブラリパスで他の EMS ディレクトリよりも前に置く必要があります。

表 53 64 ビット UNIX のリンカフラグ (1 / 2 シート)

リンカフラグ	説明
-ltibems64	すべてのプログラムは、このライブラリフラグを使ってリンクする必要があります。
-lssl -lcrypto	SSL を使用するプログラムは、これらのライブラリフラグを使ってリンクする必要があります。

表 53 64 ビット UNIX のリンカフラグ（2 / 2 シート）

リンカフラグ	説明
-lz	圧縮を使用するプログラムは、このライブラリフラグを使ってリンクする必要があります。
-ltibemslookup64 -lldap -lxml2 -llber	EMS LDAP 照合を使用するプログラムは、これらのライブラリフラグを使ってリンクする必要があります。
-ltibemsadmin64	C の Administration ライブラリを使用するプログラムは、このライブラリフラグを使ってリンクする必要があります。
-ltibemsufo64	非共有ステートフェイルオーバーライブラリを使用するプログラムは、このライブラリフラグを使ってリンクする必要があります。

Microsoft Windows

リリース 7.0 でサポートされる Windows プラットフォームについては、インストールディレクトリの `readme.txt` ファイルに記載されているリストを参照してください。DLL と静的ライブラリの両方が利用可能です。フォワードマイグレーションをスムーズに行うため、DLL を使用することをお勧めします。

表 54 Microsoft Windows のダイナミックライブラリファイル（1 / 2 シート）

ライブラリファイル	説明
動的ライブラリ（DLL）では、/MT コンパイラオプションを使用します。	
tibems.lib ws2_32.lib	すべてのプログラムは、これらのライブラリをリンクする必要があります。
tibemslookup.lib libxml2.lib	EMS LDAP 照合を使用するプログラムは、これらのライブラリをリンクする必要があります。
liboldap32.lib olber32.lib libldap32_d.lib liblber32_d.lib	さらに、EMS LDAP 照合を使用するプログラムは、これらのライブラリペアのひと組をリンクする必要があります。
tibemsadmin.lib	C の Administration ライブラリを使用するプログラムは、このライブラリを使ってリンクする必要があります。

表 54 Microsoft Windows のダイナミックライブラリファイル (2 / 2 シート)

ライブラリファイル	説明
tibemsufo.lib	C の非共有ステートフェイルオーバーライブラリを使用するプログラムは、このライブラリを使ってリンクする必要があります。

表 55 Microsoft Windows の静的ライブラリファイル

ライブラリファイル	説明
静的ライブラリ (DLL) では、/MD コンパイラオプションを使用します。	
libtibems.lib ws2_32.lib ssleay32mt.lib libeay32mt.lib zlib.lib	すべてのプログラムは、これらのライブラリをリンクする必要があります。
libtibemslookup.lib libxml2.lib	EMS LDAP 照合を使用するプログラムは、このライブラリをリンクする必要があります。
liboldap32.lib olber32.lib libldap32_d.lib liblber32_d.lib	さらに、EMS LDAP 照合を使用するプログラムは、これらのライブラリペアのひと組をリンクする必要があります。
libtibemsadmin.lib	C の Administration ライブラリを使用するプログラムは、このライブラリを使ってリンクする必要があります。
libtibemsufo.lib	C の非共有ステートフェイルオーバーライブラリを使用するプログラムは、このライブラリを使ってリンクする必要があります。

OpenVMS

OpenVMS 環境では、共有ライブラリと静的ライブラリを利用できます。フォワードマイグレーションをスムーズに行うため、共有ライブラリを使用することをお勧めします。



EMS 4.3 から 4.4 以降のバージョンへアップグレードする場合、EMS 4.3 ダイナミックライブラリ（共有可能イメージ）にリンクされていた EMS クライアントの実行可能ファイルは、関連するサードパーティ製ライブラリと共に EMS 4.4 をインストールした後の新しいライブラリに再リンクされる必要があります。サードパーティ製ライブラリは、EMS フルインストールの一部としてインストールされます。

表 56 OpenVMS の共有可能イメージライブラリファイル

ライブラリファイル	説明
LIBTIBEMSSHR.EXE	すべてのプログラムは、このライブラリをリンクする必要があります。
LIBCRYPTOSHR.EXE LIBSSLKHR.EXE	SSL を使用するプログラムは、これらのライブラリをリンクする必要があります。
LIBZKHR.EXE	データ圧縮を使用するプログラムは、このライブラリをリンクする必要があります。
LIBTIBEMSADMINSHR.EXE	C の Administration ライブラリを使用するプログラムは、このライブラリをリンクする必要があります。

表 57 OpenVMS の静的ライブラリファイル

ライブラリファイル	説明
LIBTIBEMS.OLB	すべてのプログラムは、このライブラリをリンクする必要があります。
LIBCRYPTO.OLB LIBSSL.OLB	SSL を使用するプログラムは、これらのライブラリをリンクする必要があります。
LIBZ.OLB	データ圧縮を使用するプログラムは、このライブラリをリンクする必要があります。
LIBTIBEMSADMIN.OLB	C の Administration ライブラリを使用するプログラムは、このライブラリをリンクする必要があります。

## C# プログラマーのチェックリスト

EMS C# プログラムの開発者は、開発サイクルの四つの各段階でこのチェックリストを使用できます。

### インストール

- EMS ソフトウェアをインストールすると、自動的に EMS アセンブリ DLL ファイルが `EMS_HOME\bin` サブディレクトリに格納されます。

### コーディング

- 適切な EMS アセンブリをインポートします (表 58 を参照)。

表 58 EMS アセンブリ DLL

バージョン	DLL
.NET API	TIBCO.EMS.dll
.NET Administration API	TIBCO.EMS.ADMIN.dll
.NET 非共有ステート API	TIBCO.EMS.UFO.dll
.NET Compact Framework API	TIBCO.EMS-CF.dll

### コンパイル

- 任意の .NET コンパイラを使ってコンパイルします。

### 実行

- EMS アセンブリは、グローバルアセンブリキャッシュ (この場所を推奨) か、システムパス、またはプログラムの実行ファイルと同じディレクトリのいずれかに配置する必要があります。
- 最新の .NET アセンブリに自動アップグレードするには、適切なポリシーファイルをグローバルキャッシュに含めます。詳細は、[バージョン間の自動アップグレード](#)を参照してください。
- アプリケーションは、EMS のデーモンプロセス (`tibemsd`) に接続する必要があります。

アセンブリのバージョン管理

TIBCO Enterprise Message Service のアセンブリ DLL は、`1.0.release.version` の形式でバージョン管理が行われています。`release` は EMS のリリース番号で、`version` は任意の値です。たとえば、ソフトウェアリリース 6.0.0 のアセンブリバージョン番号は `1.0.600.8` のようになります。



EMS 6.0 より前のリリースを使用するアプリケーションは、標準の .NET バージョン管理を使用しません。TIBCO Enterprise Message Service リリース 6.0.0 までは、すべての EMS .NET アセンブリでアセンブリバージョン番号を `1.0.0.0` と表示し、再ビルドなしでクライアントアプリケーションが EMS の最新バージョンにアップグレードできるようにしていました。

この機能は、現在は `policy` DLL ファイルを通じて利用できるようになっています。

バージョン間の自動アップグレード

リリース間のシームレスなアップグレードを可能とするため、既存アプリケーションを古いアセンブリから新しいアセンブリへとリダイレクトするポリシーファイルおよび設定ファイルが TIBCO Enterprise Message Service のインストールに含まれています。ポリシーファイルと設定ファイルは EMS ライブラリごとにあります。

- `policy.1.0.assembly` ファイル。例：`policy.1.0.TIBCO.EMS.dll`。自動アップグレードを有効にするには、ポリシーファイルをグローバルキャッシュに含める必要があります。
- `assembly.config` ファイル。例：`TIBCO.EMS.dll.config`。関連するポリシーファイルをグローバルキャッシュに追加するときは、設定ファイルが存在している必要があります。

表 59 は、各 EMS アセンブリのポリシーファイルと設定ファイルを示しています。

表 59 EMS ポリシーファイル

バージョン	ファイル
.NET API	<code>policy.1.0.TIBCO.EMS.dll</code>
	<code>TIBCO.EMS.dll.config</code>
.NET Administration API	<code>policy.1.0.TIBCO.EMS.ADMIN.dll</code>
	<code>TIBCO.EMS.ADMIN.dll.config</code>
.NET 非共有ステート API	<code>policy.1.0.TIBCO.EMS.UFO.dll</code>
	<code>TIBCO.EMS.UFO.dll.config</code>

.NET Compact Framework API は自動アップグレードに対応していません。

アップデートの有効化

ライブラリの自動アップデートを有効にするには、適切なポリシーファイルをグローバルキャッシュに追加します。ポリシーファイルをグローバルキャッシュに追加するときは、ディレクトリに関連する設定ファイルが存在している必要があります。



## 自動アップグレードの無効化

古いアプリケーションを新しいバージョンに自動的に移行させないようにするには、`policy DLL` をグローバルキャッシュに含めません。`policy.1.0.assembly` ファイルが存在しないと、クライアントアプリケーションはアップグレードされません。

## 異なる EMS リリースからの複数クライアントの実行

異なる TIBCO Enterprise Message Service リリースでビルドされたふたつ以上のアプリケーションを実装するには、以下の手順に従います。

1. 異なる .NET クライアントアセンブリを使ってクライアントをビルドします。
2. 必要な .NET クライアントアセンブリをグローバルキャッシュに含めます。
3. `policy DLL` をグローバルキャッシュに含めないようにします。

## 非対応の機能と制限

このセクションは、.NET ライブラリまたは .NET Compact Framework ライブラリのいずれかで利用できない機能をまとめています。

圧縮、SSL の使用、LDAP による管理対象オブジェクトの照合は、Microsoft .NET Framework 2.0 でのみ利用できます。

表 60 .NET 機能のサポート

機能	.NET	.NET Compact Framework
外部トランザクションマネージャ用 XA プロトコル	—	—
ConnectionConsumer、ServerSession、ServerSessionPool	—	—
圧縮	はい	—
SSL	はい	—
ソケットバッファサイズの変更 (HTML リファレンスの <code>Tibems.SetSocketReceiveBufferSize</code> と <code>Tibems.SetSocketSendBufferSize</code> を参照)	はい	—
デーモン スレッド (HTML リファレンスの <code>Tibems.SetSessionDispatcherdaemon</code> を参照)	はい	—

## 文字エンコーディング

.NET プログラムはメッセージ中の文字列をバイト配列として表します。EMS プログラムは、アウトバウンドメッセージを送信する前に、プログラムが指定するエンコーディングを使って文字列をそれぞれのバイト配列に変換します。逆に、EMS プログラムがインバウンドメッセージを受信するときは、同じエンコーディングを使ってバイト配列から文字列を再構築します。

プログラムがエンコーディングを指定すると、メッセージの本文（名前および値）とプロパティ（名前および値）のすべての文字列に適用されます。ヘッダの名前と値には適用されません。`BytesMessage.WriteUTF` 方式では、常に UTF-8 をエンコーディングに使用します。

### アウトバウンド メッセージ

プログラムがアウトバウンドメッセージの文字列エンコーディングを決定する方法には 3 通りあります。

- デフォルトのグローバルエンコーディングである UTF-8 を使用する。
- `Tibems.SetEncoding` でデフォルト以外のグローバルエンコーディングを（すべてのアウトバウンドメッセージに対して）設定する。
- `Tibems.SetMessageEncoding` で個別のメッセージに対してエンコーディングを設定する。

### インバウンドメ ッセージ

他の EMS クライアントからのインバウンドメッセージは明示的にエンコーディングを伝えてきます。受信側のクライアントは、それに応じたエンコーディングを使ってメッセージをデコードします。

文字エンコーディングの詳細は、35 ページの「[メッセージの文字エンコーディング](#)」を参照してください。

## .NET Compact Framework (CF)

このセクションでは、EMS .NET Compact Framework API を使ってハンドヘルドデバイス用にアプリケーションを開発する場合の推奨事項を説明します。

### スレッド

.NET Compact Framework はバックグラウンドスレッドをサポートしません。スレッド関連の問題を避けるため、プログラムの終了時は、終了前にすべての EMS リソースを解放するようお勧めします。たとえば、不要になった EMS 接続は閉じてください（HTML リファレンスの `Connection.Close` を参照）。

### クロック精度

クロックの精度はすべての時間関連呼び出しやパラメータの精度に影響します。たとえば、`MessageConsumer.Receive(timeout)` や接続遅延などです。ハンドヘルドデバイスの一部では、予想以上にクロック精度が低い場合があります。時間の値を選択する場合は、あらかじめ対象デバイスの精度を確認してください。

## 接続ファクトリ

クライアントが、JMS 処理を実行するには、実行中の EMS サーバーのインスタンスに接続する必要があります。接続ファクトリとは、EMS サーバーへのクライアント接続の定義に使用されるデータをカプセル化しているオブジェクトです。接続ファクトリで最低限設定が必要なパラメータは、EMS サーバーへのクライアント接続の接続タイプ（JMS 1.1 の接続タイプは「ジェネリック」、JMS 1.0.2b の接続タイプは「トピック」または「キュー」）および URL（ホスト名、トランスポートプロトコル、ポート ID）です。

接続ファクトリは、アプリケーションによって動的に作成されるか、名前付けサービスを使って JNDI (Java Naming and Directory Interface) サーバーまたは LDAP (Lightweight Directory Access Protocol) サーバーなどのデータストアから取得します。

### 接続ファクトリの照合

EMS は接続ファクトリの保存に使用できる JNDI 実装を提供します。Java、C、および C# クライアントは、EMS の JNDI 実装を使用して接続ファクトリを照合します。

接続ファクトリは、JNDI 準拠の名前付けサービスまたは LDAP サーバー内に保存することもできます。Java クライアントは、JNDI 準拠の名前付けサービス内の接続ファクトリを照合できます。C と C# のクライアントは LDAP サーバーを使用します。

EMS サーバー内の接続ファクトリを照合する方法については、350 ページの「[EMS に保存されている管理対象オブジェクトの照合](#)」を参照してください。EMS サーバーに接続ファクトリを作成する方法については、348 ページの「[EMS での管理対象オブジェクトの作成と変更](#)」を参照してください。

### 接続ファクトリの動的作成

通常、クライアントアプリケーションは JNDI を使用して接続ファクトリオブジェクトを参照します。しかし、状況によっては、サーバーに直接、クライアントを接続することが必要になる場合があります。EMS サーバーに直接接続するには、アプリケーションは動的に接続ファクトリを作成する必要があります。

以下の例は、JMS 1.1 接続でサポートされる各言語で接続ファクトリを作成する方法を示しています。各 API は、JMS 1.1 の XA 接続の接続ファクトリもサポートします。

各コード例のこれらの式に使用されている `serverUrl` パラメータには、EMS サーバーの実行インスタンスのプロトコルとアドレスを定義する文字列を指定します。`serverUrl` パラメータは次の形式で指定します。

```
serverUrl = protocol://host:port
```

`protocols` には、**tcp** または **ssl** のいずれかを指定できます。例：

```
serverUrl = tcp://server0:7222
```

フォールトトレランス接続の場合、複数の URL を指定できます。例：

```
serverUrl = tcp://server0:7222,tcp://server1:7344
```

詳細は、492 ページの「[フォールトトレランスサーバー用のクライアントの設定](#)」を参照してください。SSL を使用したサーバーへのセキュアな接続の作成方法についての詳細は、460 ページの「[EMS クライアントでの SSL の設定](#)」と 348 ページの「[セキュア接続のための接続ファクトリの作成](#)」を参照してください。

## Java

Java クライアントで `TibjmsConnectionFactory` オブジェクトを動的に作成するには、次のように記述します。

```
ConnectionFactory factory = new
    com.tibco.tibjms.TibjmsConnectionFactory(serverUrl);
```

実行可能な例は、`tibjmsMsgProducer.java` サンプルクライアントを参照してください。

## C

C クライアントで `tibemsConnectionFactory` 型を動的に作成するには、次のように記述します。

```
factory = tibemsConnectionFactory_Create();
status = tibemsConnectionFactory_SetServerURL(
    factory, serverUrl);
```

実行可能な例は、`tibemsMsgProducer.c` サンプルクライアントを参照してください。

## C#

C# クライアントに `ConnectionFactory` オブジェクトを動的に作成するには、次のように記述します。

```
ConnectionFactory factory = new
    TIBCO.EMS.ConnectionFactory(serverUrl);
```

実行可能な例は、`csMsgProducer.cs` サンプルクライアントを参照してください。

## 接続の試行回数、タイムアウト、および遅延時間のパラメータ設定

デフォルトでは、クライアントは 500 ミリ秒の間隔（遅延時間）でサーバーへの接続を 2 回試みるように設定されています。クライアントのこの動作を変更するには、接続回数と遅延時間の値を新しく設定します。また、クライアントが EMS サーバーへの接続を試みる際に、ハングアップを引き起こす要因が数多くあるた

め、接続のタイムアウト値を設定し、指定した時間が経過した以降は、接続を試行しないように設定することができます。最適な結果を得るには、最低 500 ミリ秒とすることをお勧めします。EMS では、フォールトトレランスによるスイッチオーバーが発生した後の再接続の試行回数、遅延時間、タイムアウトをそれぞれ個別に設定することができます (493 ページの「[再接続障害パラメータの設定](#)」を参照)。

以下の各コード例は、試行回数 10、遅延時間 1000 ミリ秒、タイムアウト 1000 ミリ秒の接続の設定例です。

## Java

`TibjmsConnectionFactory` オブジェクトの `setConnAttemptCount()`、`setConnAttemptDelay()`、および `setConnAttemptTimeout()` メソッドを使って、新しい接続障害パラメータを設定します。

```
factory.setConnAttemptCount(10);
factory.setConnAttemptDelay(1000);
factory.setConnAttemptTimeout(1000);
```

## C

`tibemsConnectionFactory_SetConnectAttemptCount` および `tibemsConnectionFactory_SetConnectAttemptDelay` 関数を使って、新しい接続障害パラメータを設定します。

```
status = tibemsConnectionFactory_SetConnectAttemptCount(
    factory, 10);

status = tibemsConnectionFactory_SetConnectAttemptDelay(
    factory, 1000);

status = tibemsConnectionFactory_SetConnectAttemptTimeout(
    factory, 1000);
```

## C#

`ConnectionFactory.SetConnAttemptCount`、`ConnectionFactory.SetConnAttemptDelay`、および `ConnectionFactory.SetConnAttemptTimeout` メソッドを使って、新しい接続障害パラメータを設定します。

```
factory.setConnAttemptCount(10);
factory.setConnAttemptDelay(1000);
factory.setConnAttemptTimeout(1000);
```

## EMS サーバーへの接続

---

EMS サーバーとの接続は、接続ファクトリから取得される `Connection` オブジェクトによって定義されます (323 ページの「[接続ファクトリ](#)」を参照)。

`Connection` オブジェクトはかなり重いオブジェクトであるため、クライアントのほとんどは、いったん接続を作成すると、クライアントが終了するまでこのオブジェクトを開いたままにしています。必要に応じて、アプリケーションに複数の接続を作成できます。

以下の例は `Connection` オブジェクトを作成する方法を示しています。

### Java

`Connection` オブジェクトを作成するには、`TibjmsConnectionFactory` オブジェクトの `createConnection()` メソッドを使用します。

```
Connection connection =  
    factory.createConnection(userName,password);
```

実行可能な例は、`tibjmsMsgProducer.java` サンプルクライアントを参照してください。

### C

`tibemsConnectionFactory_CreateConnection` 関数を使って、`tibemsConnection` 型の接続を作成します。

```
tibemsConnection connection = NULL;  
  
status = tibemsConnectionFactory_CreateConnection(factory,  
    &connection, userName, password);
```

接続ファクトリがない場合、C クライアントは `tibemsConnection_Create` 関数を使って、動的に `tibemsConnection` 型の接続を作成します。

```
status = tibemsConnection_Create(&connection,  
    serverUrl,NULL,userName,password);
```

`tibemsConnection_Create` は下位互換性のために用意されている関数ですが、ファクトリから `tibemsConnection` オブジェクトを作成する方法をお勧めします。

実行可能な例は、`tibemsMsgProducer.c` サンプルクライアントを参照してください。

## C#

Connection オブジェクトを作成するには、ConnectionFactory.CreateConnection メソッドを使用します。

```
Connection connection =  
    factory.CreateConnection(userName, password);
```

実行可能な例は、csMsgProducer.cs サンプルクライアントを参照してください。

## 接続の開始、停止、および切断

メッセージをコンシュームする前に、メッセージコンシューマクライアントは接続を「開始」する必要があります。メッセージコンシューマについての詳細は、336 ページの「[メッセージコンシューマの作成](#)」を参照してください。

メッセージ配信を一時的に停止する場合、接続を「停止」できます。

クライアントアプリケーションを終了する場合、開いているすべての接続を「切断」する必要があります。使用されていない、開いていたままの接続は、最終的に切断されますが、他のアプリケーションのために使用可能なリソースが消費されたままになります。接続を切断すると、その接続で作成されたセッションも閉じられます。

Java Connection オブジェクト、C の tibemsConnection タイプ、および C# の Connection オブジェクトの「start」（開始）、「stop」（停止）、「close」（切断）の各メソッドを参照してください。

## セッションの作成

セッションは、メッセージをプロデュースおよびコンシュームするためのシングルスレッドコンテキストです。Session オブジェクトを使ってメッセージプロデュースまたはメッセージコンシューマを作成します。セッションをトランザクション化すると、1 回のトランザクションでメッセージのグループを送受信することができます。トランザクション処理を行わない（非トランザクション）セッションでは、セッションで受信されたメッセージオブジェクトの確認応答モードを定義できます。詳細は、38 ページの「[メッセージの確認応答](#)」を参照してください。

### Java

Session オブジェクトを作成するには、Connection オブジェクトの `createSession()` メソッドを使用します。

たとえば、`AUTO_ACKNOWLEDGE` 配信モードを使用する非トランザクション Session を作成するには、次のように記述します。

```
Session session = connection.createSession(
    false, javax.jms.Session.AUTO_ACKNOWLEDGE);
```

`NO_ACKNOWLEDGE` などの拡張された EMS 確認応答モードでは、EMS 配信モードを指定する際に、`com.tibco.tibjms.Tibjms` 定数をインクルードする必要があります。たとえば、`NO_ACKNOWLEDGE` 配信モードを使用する非トランザクション Session を作成するには、次のように記述します。

```
Session session = Connection.createSession(
    false, com.tibco.tibjms.Tibjms.NO_ACKNOWLEDGE);
```

実行可能な例は、`tibjmsMsgProducer.java` サンプルクライアントを参照してください。

### C

`tibemsConnection_CreateSession` 関数を使って、`tibemsSession` タイプのセッションを作成します。

```
tibemsSession session = NULL;

status = tibemsConnection_CreateSession(connection,
    &session, TIBEMS_FALSE, TIBEMS_AUTO_ACKNOWLEDGE);
```

実行可能な例は、`tibemsMsgProducer.c` サンプルクライアントを参照してください。

### C#

Session オブジェクトを作成するには、`Connection.CreateSession` メソッドを使用します。

```
Session session = connection.CreateSession(false,
    Session.AUTO_ACKNOWLEDGE);
```

実行可能な例は、`csMsgProducer.cs` サンプルクライアントを参照してください。



## 例外リスナーの設定

どの API についても、接続の切断時や、フォールトトレランス スイッチオーバーが発生した場合に呼び出される例外リスナーを、接続に設定することができます。

イベントが切断の場合、例外ハンドラはさまざまな EMS メソッドを、何の問題もなく呼び出すことができます。しかし、イベントがフォールトトレランス スイッチオーバーの場合、例外ハンドラは EMS メソッドを呼び出すことができません。デッドロックが発生する恐れがあるからです。setExceptionOnFTSwitch メソッドを呼び出して、フォールトトレランス スイッチオーバーが発生した後に、新しいサーバーの URL を含む例外を受信することができます。

以下の例は、接続の例外リスナーを作成する方法を示しています。

### Java

ExceptionListener.onException メソッドを実装して、Connection オブジェクトの setExceptionListener メソッドを使って例外リスナーを登録し、Tibjms.setExceptionOnFTSwitch を呼び出すと、フォールトトレランスのスイッチオーバーが発生した後に例外ハンドラを呼び出します。

```
public class tibjmsMsgConsumer
    implements ExceptionListener
{
.....
    public void onException(JMSEException e)
    {
        /* Handle exception */
    }
.....
    connection.setExceptionListener(this);

    com.tibco.tibjms.Tibjms.setExceptionOnFTSwitch(true);
.....
}
```

実行可能な例は、tibjmsMsgConsumer.java サンプルクライアント (setExceptionOnFTSwitch 呼び出しは記述されていません) を参照してください。

**C**

例外を処理する `onException` 関数を定義してから、エラーの発生時に `onException` を呼び出すために、`tibemsConnection_SetExceptionListener` 関数を使用します。また、フォールトトレランス スイッチオーバーの発生後に例外ハンドラを呼び出すには、`tibems_setExceptionOnFTSwitch` を使用します。

```
void onException(
    tibemsConnection conn,
    tibems_status     reason,
    void*             closure)
{
    /* Handle exception */
}
.....

status = tibemsConnection_SetExceptionListener(
                                           connection,
                                           onException,
                                           NULL);

tibems_setExceptionOnFTSwitch(TIBEMS_TRUE);
```

実行可能な例は、`tibemsMsgConsumer.c` サンプルクライアント (`setExceptionOnFTSwitch` 呼び出しは記述されていません) を参照してください。

**C#**

`IEnumerator.OnException` メソッドを実装し、`Connection` オブジェクトの `ExceptionListener` プロパティを使って例外リスナーを登録します。また、フォールトトレランスのスイッチオーバーの発生後に例外ハンドラを呼び出すには、`Tibems.SetExceptionOnFTSwitch` を使用します。

```
public class csMsgConsumer : IEnumerator
{
    .....
    public void OnException(EMSEException e)
    {
        /* Handle exception */
    }
    .....
    connection.ExceptionListener = this;

    TIBCO.EMS.Tibems.SetExceptionOnFTSwitch(true);
    .....
}
```

実行可能な例は、`csMsgConsumer.cs` サンプルクライアント (`setExceptionOnFTSwitch` 呼び出しは記述されていません) を参照してください。

## トピックとキューの動的作成

EMSはトピックとキューの保存に使用できる JNDI 実装を提供します。Java、C、および C# クライアントは、EMS の JNDI 実装を使用してトピックとキューを参照します。

トピックとキューを JNDI 準拠の名前付けサービスまたは LDAP サーバー内に保存することもできます。Java クライアントは、JNDI 準拠の名前付けサービス内のトピックとキューを照合できます。C と C# のクライアントは LDAP サーバーを使用します。

EMS サーバー内のトピックとキューを照合する方法については、350 ページの「[EMS に保存されている管理対象オブジェクトの照合](#)」を参照してください。

また、必要に応じて、クライアントはデスティネーションの作成もできます。クライアントが作成を要求したデスティネーションがすでに存在する場合は、既存のデスティネーションが使用されます。デスティネーションが存在せず、[topics.conf](#)、[queues.conf](#)、または [acl.conf](#) のファイルの指定でそのデスティネーションを許可している場合、サーバーは動的に新しいデスティネーションを作成します。新しく作成されたデスティネーションは、親のプロパティおよびアクセス権限を継承します (75 ページの「[ワイルドカードと動的に作成されたデスティネーション](#)」を参照)。デスティネーションは、そのデスティネーションを使用するクライアントが実行されている限り、サーバーによって管理されます。



動的デスティネーションは設定ファイルの要素にはないため、クライアントは動的に作成されたキューとトピックの照合に JNDI を使用できません。

以下の例は、デスティネーションを動的に作成する方法を示しています。

### Java

Session オブジェクトの `createTopic()` メソッドを使って、トピックを Destination オブジェクトとして作成します。

```
Destination topic = session.createTopic(topicName);
```

Session オブジェクトの `createQueue()` メソッドを使って、キューを Destination オブジェクトとして作成します。

```
Destination queue = session.createQueue(queueName);
```

実行可能な例は、`tibjmsMsgProducer.java` サンプルクライアントを参照してください。

### C

`tibemsTopic_Create` 関数を使って、`tibemsDestination` タイプのトピックを作成します。

```
tibemsDestination topic = NULL;
status = tibemsTopic_Create(&topic, topicName);
```

`tibemsQueue_Create` 関数を使って、`tibemsDestination` タイプのキューを作成します。

```
tibemsDestination queue = NULL;  
status = tibemsQueue_Create(&queue, queueName);
```

実行可能な例は、`tibemsMsgProducer.c` サンプルクライアントを参照してください。

## C#

Topic オブジェクトを作成するには、`Session.CreateTopic` メソッドを使用します。

```
Destination topic = session.CreateTopic(topicName);
```

Queue オブジェクトを作成するには、`Session.CreateQueue` メソッドを使用します。

```
Destination queue = session.CreateQueue(queueName);
```

実行可能な例は、`csMsgProducer.cs` サンプルクライアントを参照してください。

## メッセージプロデューサの作成

「メッセージプロデューサ」とは、メッセージをトピックにパブリッシュするか、メッセージをキューに送信するか、そのいずれかの処理を行う EMS クライアントです (3 ページの「[JMS メッセージモデル](#)」を参照)。トピックをパブリッシュする場合、メッセージプロデューサは、通常、「パブリッシャ」と呼ばれます。オプションで、メッセージプロデューサの作成時に、デスティネーションに `NULL` を指定しておき、実際にメッセージを送信またはパブリッシュするときに、デスティネーションを指定することができます (343 ページの「[メッセージの送信](#)」を参照)。

キューにメッセージを送信するメッセージプロデューサを作成するには、そのキューへの `send` アクセス権限を持っている必要があります。トピックの新しい持続サブスクライバを作成するには、そのトピックの `durable` アクセス権限を持っている必要があります。また、トピックの既存の持続サブスクライバに接続するには、少なくとも `use_durable` アクセス権限を持っている必要があります。詳細は、273 ページの「[ユーザーのアクセス権限](#)」を参照してください。

以下の各コード例では、331 ページの「[トピックとキューの動的作成](#)」で動的に作成したキューに、メッセージを送信するメッセージプロデューサを作成します。

### Java

`MessageProducer` オブジェクトを作成するには、`Session` オブジェクトの `createProducer()` メソッドを使用します。

```
MessageProducer QueueSender = session.createProducer(queue);
```

実行可能な例は、`tibjmsMsgProducer.java` サンプルクライアントを参照してください。

### C

`tibemsMsgProducer` 型のメッセージプロデューサを作成するには、`tibemsSession_CreateProducer` 関数を使用します。

```
tibemsMsgProducer QueueSender = NULL;
status = tibemsSession_CreateProducer(session,
                                     &QueueSender, queue);
```

実行可能な例は、`tibemsMsgProducer.c` サンプルクライアントを参照してください。

### C#

`MessageProducer` オブジェクトを作成するには、`Session.CreateProducer` メソッドを使用します。

```
MessageProducer QueueSender = session.CreateProducer(queue);
```

実行可能な例は、`csMsgProducer.cs` サンプルクライアントを参照してください。

## メッセージプロデューサの設定

メッセージプロデューサは、メッセージのルーティングと配信の方法を定義したデフォルトのヘッダおよびプロパティを含むメッセージを生成するように設定できます。具体的には、以下のことができます。

- プロデューサのデフォルトの配信モードを設定する。
- メッセージ ID を無効にするかどうかを設定する。
- メッセージのタイムスタンプを無効にするかどうかを設定する。
- プロデューサのデフォルトの優先度を設定する。
- プロデュースされたメッセージがメッセージシステムに保持されるデフォルトの期間を設定する。

たとえば、メッセージ配信モードに、PERSISTENT、NON\_PERSISTENT、または RELIABLE\_DELIVERY のいずれかのモードを設定できます (24 ページの「[メッセージ配信モード](#)」を参照)。

### Java

メッセージプロデューサで RELIABLE\_DELIVERY 配信モードをデフォルトに設定するには、MessageProducer オブジェクトの setDeliveryMode() メソッドを使用します。

```
QueueSender.setDeliveryMode(
    com.tibco.tibjms.Tibjms.RELIABLE_DELIVERY);
```

To configure the Message Producer with a default delivery mode of NON\_PERSISTENT:

```
QueueSender.setDeliveryMode(
    javax.jms.DeliveryMode.NON_PERSISTENT);
```

実行可能な例は、tibjmsMsgProducerPerf.java サンプルクライアントを参照してください。



配信モードは、Message.setJMSDeliveryMode() メソッドを使って設定できません。JMS 仕様に従い、パブリッシャは、メッセージをパブリッシュする際に JMSDeliveryMode ヘッダフィールドの値を無視します。

### C

プロデュースする各メッセージのデフォルトの配信モードを RELIABLE\_DELIVERY に設定するには、tibemsMsgProducer\_SetDeliveryMode 関数を使用します。

```
tibems_int deliveryMode = TIBEMS_RELIABLE;
status tibemsMsgProducer_SetDeliveryMode(QueueSender,
                                          deliveryMode);
```

## C#

`MessageProducer` オブジェクトの `DeliveryMode` を `RELIABLE_DELIVERY` に設定します。

```
QueueSender.DeliveryMode = DeliveryMode.RELIABLE_DELIVERY;
```

実行可能な例は、`csMsgProducerPerf.cs` サンプルクライアントを参照してください。

## メッセージコンシューマの作成

メッセージコンシューマとは、トピックにパブリッシュされたメッセージまたはキューに送信されたメッセージを受け取るクライアントです。トピックからメッセージを受け取る場合、メッセージコンシューマは、通常、「サブスクライバ」と呼ばれます。

メッセージコンシューマの作成時に、メッセージのコンシュームを特定のプロパティの値に制限する「メッセージセクタ」を指定できます。トピックのメッセージコンシューマを作成する際、`noLocal` 属性を設定できます。この属性により、メッセージがコンシュームされるのと同じ接続上でパブリッシュされるメッセージをコンシュームすることはできません。

キューコンシューマで使用されているメッセージセクタに細心の注意を払ってください。キューコンシューマのメッセージセクタに一致しないメッセージは他のコンシューマによって取得されるまでキューに残ります。そのため、一致しないメッセージではセクタの失敗が多数発生します。これは特に、キューコンシューマが接続し、メッセージをコンシュームした直後に切断する場合に起こります。

トピックにパブリッシュされたメッセージは、そのトピックに対してアクティブなサブスクライバによってのみコンシュームされます。そのトピックに対してアクティブなサブスクライバがない場合、メッセージはコンシュームされず、後で取得することもできません (5 ページの「[トピックの持続サブスクライバ](#)」を参照)。持続サブスクライバを作成すると、サブスクライバが現在起動していない場合でも、トピックにパブリッシュされたメッセージを、サブスクライバが確実に受け取れるようにできます。キューの場合、メッセージコンシューマによってコンシュームされるか、メッセージの有効期限が切れるか、キューの最大サイズに達するかのいずれかの条件が満たされるまで、メッセージはキューに保存されます。

以下の例では、キューのメッセージをコンシュームするメッセージコンシューマとトピックのメッセージをコンシュームする持続サブスクライバを作成します。キューとトピックは、331 ページの「[トピックとキューの動的作成](#)」で動的に作成されたものです。



`createDurableSubscriber` メソッドは、トピックに新しい持続サブスクライバを作成するか、既存の持続サブスクライバにクライアントを接続します。トピックに新しい持続サブスクライバを作成するには、そのトピックに対する `durable` アクセス権限が必要です。トピックの既存の持続サブスクライバに接続するには、少なくともそのトピックに対する `use_durable` アクセス権限が必要です。詳細は、273 ページの「[ユーザーのアクセス権限](#)」を参照してください。

### Java

`MessageConsumer` オブジェクトを作成するには、`Session` オブジェクトの `createConsumer()` メソッドを使用します。

```
MessageConsumer QueueReceiver = session.createConsumer(queue);
```

実行可能な例は、`tibjmsMsgConsumer.java` サンプルクライアントを参照してください。



以下の `Session.createDurableSubscriber()` メソッドでは、「MyDurable」という名前の持続サブスクライバを作成します。

```
TopicSubscriber subscriber =
    session.createDurableSubscriber(topic, "myDurable");
```

実行可能な例は、`tibjmsDurable.java` サンプルクライアントを参照してください。

## C

`tibemsMsgConsumer` 型のメッセージコンシューマを作成するには、`tibemsSession_CreateConsumer` 関数を使用します。

```
tibemsMsgConsumer QueueReceiver = NULL;
status = tibemsSession_CreateConsumer(session,
    &QueueReceiver, queue, NULL, TIBEMS_FALSE);
```

実行可能な例は、`tibemsMsgConsumer.c` サンプルクライアントを参照してください。

以下の `tibemsSession_CreateDurableSubscriber` 関数は、「myDurable」という名前の `tibemsMsgConsumer` 型の持続サブスクライバを作成します。

```
tibemsMsgConsumer msgConsumer = NULL;
status = tibemsSession_CreateDurableSubscriber(session,
    &msgConsumer, topic, "myDurable",
    NULL, TIBEMS_FALSE);
```

実行可能な例は、`tibemsDurable.c` サンプルクライアントを参照してください。

## C#

`MessageConsumer` オブジェクトを作成するには、`Session.CreateConsumer` メソッドを使用します。

```
MessageConsumer QueueReceiver = session.createConsumer(queue);
```

実行可能な例は、`csMsgConsumer.cs` サンプルクライアントを参照してください。

以下の `Session.CreateDurableSubscriber` メソッドでは、「MyDurable」という名前の持続サブスクライバを作成します。

```
TopicSubscriber subscriber =
    session.CreateDurableSubscriber(topic, "myDurable");
```

実行可能な例は、`csDurable.cs` サンプルクライアントを参照してください。

## 非同期でメッセージをコンシュームするメッセージリスナーの作成

EMS では、メッセージコンシューマは、同期または非同期のいずれかでメッセージをコンシュームできます。同期コンシュームの場合、メッセージコンシューマはトピックまたはキューに対して明示的に `receive` メソッドを呼び出します。非同期コンシュームの場合、メッセージの非同期イベントハンドラとして機能する「メッセージリスナー」を実装できます。

メッセージリスナーの実装には、メッセージがデスティネーションに届くと EMS サーバーによって呼び出される `onMessage` というメソッドがひとつだけあります。`onMessage` メソッドを実装し、メッセージが届いたときに希望のアクションが実行されるようにします。例外ハンドラを実装する場合、すべての例外を処理できなければならない、スローされる例外があってはけません。

メッセージリスナーを作成したら、接続の `start` メソッドを呼び出してメッセージの受信を開始する前に、特定のメッセージコンシューマにリスナーを登録する必要があります。

メッセージリスナーは、デスティネーションのタイプに限定されません。リスナーが登録されているメッセージコンシューマに設定されたデスティネーションに従って、同じリスナーがキューまたはトピックからメッセージを取得できます。



J2EE 1.3 プラットフォームには、メッセージリスナーの特殊な形態である MDB (message-driven bean) が導入されています。MDB の詳細は、J2EE のマニュアルを参照してください。

## Java

`MessageListener` インタフェースの実装を作成し、`MessageConsumer` を作成して `MessageConsumer` オブジェクトの `setMessageListener()` メソッドを使って、メッセージリスナーをメッセージコンシューマに登録します。

```
public class tibjmsAsyncMsgConsumer implements MessageListener
{
    /* Create a connection, session and consumer */
    ...

    MessageConsumer QueueReceiver =
        session.createConsumer(queue);

    QueueReceiver.setMessageListener(this);

    connection.start();
}
```



`Session.setMessageListener()` メソッドを使用しないでください。このメソッドは、アプリケーションではなく、アプリケーションのサーバーによって使用されています。

`onMessage()` メソッドを実装して、メッセージが届いたときに希望のアクションが実行されるようにします。

```
public void onMessage(Message message)
{
    /* Process message and handle exceptions */
}
```

実行可能な例は、`tibjmsAsyncMsgConsumer.java` サンプルクライアントを参照してください。

**C**

`onMessage()` 関数を実装して、メッセージが届いたときに希望のアクションが実行されるようにします。

```
void onMessage(tibemsMsgConsumer QueueReceiver,
               tibemsMsg message, void* closure)
{
    /* Process message and handle exceptions */
}
```

別の関数で、`tibemsMsgConsumer` を作成し、`tibemsMsgConsumer_SetMsgListener` 関数を使ってメッセージコンシューマのメッセージリスナーを作成し、コールバック関数として `onMessage()` を指定します。

```
void run()
{
    tibemsMsgConsumer QueueReceiver = NULL;
    /* Create a connection, session and consumer */
    ...

    status = tibemsSession_CreateConsumer(session,
                                           &QueueReceiver, queue, NULL, TIBEMS_FALSE);

    status = tibemsMsgConsumer_SetMsgListener(QueueReceiver,
                                              onMessage, NULL);

    status = tibemsConnection_Start(connection);
}
```

実行可能な例は、`tibemsAsyncMsgConsumer.c` サンプルクライアントを参照してください。

**C#**

`IMessageListener` インタフェースの実装を作成し、`Session.CreateConsumer` を使って `MessageConsumer` を作成し、`MessageConsumer` オブジェクトの `MessageListener` プロパティを設定し、メッセージリスナーをメッセージコンシューマに登録します。

```
public class csAsyncMsgConsumer : IMessageListener
{
    /* Create a connection, session and consumer */
    ...

    MessageConsumer QueueReceiver =
        session.CreateConsumer(queue);

    QueueReceiver.MessageListener = this;

    connection.Start();
}
```

`IMessageListener.OnMessage` メソッドを実装して、メッセージが届いたときに希望のアクションが実行されるようにします。

```
public void OnMessage(Message message) {  
    try  
    {  
        /* Process message and handle exceptions */  
    }  
}
```

実行可能な例は、`csAsyncMsgConsumer.cs` および  
`csAsyncMsgConsumerUsingDelegate.cs` サンプルクライアントを参照してください。

## メッセージの操作

---

メッセージは、自己完結的な情報のまとまりで、データの交換または処理の要求のために JMS アプリケーションによって使用されます。

### メッセージの作成

21 ページの「[JMS メッセージ本文](#)」で説明したとおり、EMS では次のメッセージタイプを扱います。

- 本文のないメッセージ
- テキストメッセージ
- マップメッセージ
- バイトメッセージ
- ストリームメッセージ
- オブジェクトメッセージ

各メッセージタイプには、専用の `create` メソッドが用意されています。

以下の例は、文字列「Hello」を含む単純なテキストメッセージを作成する方法を示しています。

#### Java

`Session` オブジェクトの `createTextMessage()` メソッドを使って、`TextMessage` を作成します。

```
TextMessage message = session.createTextMessage("Hello");
```

実行可能な例は、`tibjmsMsgProducer.java` サンプルクライアントを参照してください。

#### C

`tibemsTextMsg_Create` 関数を使って、`tibemsTextMsg` 型のテキストメッセージを作成します。

```
tibemsTextMsg message = "Hello";  
status = tibemsTextMsg_Create(&message);
```

実行可能な例は、`tibemsMsgProducer.c` サンプルクライアントを参照してください。

**C#**

`Session.CreateTextMessage` メソッドを使って、`TextMessage` タイプのテキストメッセージを作成します。

```
TextMessage message = session.CreateTextMessage("Hello");
```

実行可能な例は、`csMsgProducer.cs` サンプルクライアントを参照してください。

**メッセージプロパティの設定と取得**

メッセージを送信する前に、クライアントは「`set property`」メソッドを使ってメッセージのプロパティを設定できます (19 ページの「[EMS メッセージのプロパティ](#)」を参照)。クライアントは、「`get property`」メソッドを使ってメッセージのプロパティを確認できます。

**Java**

[JMS\\_TIBCO\\_PRESERVE\\_UNDELIVERED](#) プロパティに `true` を設定するには、`Message` オブジェクトの `setBooleanProperty()` メソッドを使用します。

```
message.setBooleanProperty("JMS_TIBCO_PRESERVE_UNDELIVERED",
    true);
```

[JMS\\_TIBCO\\_SENDER](#) のユーザー ID を取得するには、`getStringProperty()` メソッドを使用します。

```
userID = message.getStringProperty("JMS_TIBCO_SENDER");
```

**C**

[JMS\\_TIBCO\\_PRESERVE\\_UNDELIVERED](#) プロパティに `true` を設定するには、`tibemsMsg_SetBooleanProperty` 関数を使用します。

```
status = tibemsMsg_SetBooleanProperty(message,
    "JMS_TIBCO_PRESERVE_UNDELIVERED", true);
```

[JMS\\_TIBCO\\_SENDER](#) のユーザー ID を取得するには、`tibemsMsg_GetStringProperty` 関数を使用します。

```
char* userID = NULL;
status = tibemsMsg_GetStringProperty(message,
    "JMS_TIBCO_SENDER", &userID);
```

**C#**

[JMS\\_TIBCO\\_PRESERVE\\_UNDELIVERED](#) プロパティに `true` を設定するには、`Message.SetBooleanProperty` メソッドを使用します。

```
message.SetBooleanProperty("JMS_TIBCO_PRESERVE_UNDELIVERED",
    true);
```

`JMS_TIBCO_SENDER` のユーザー ID を取得するには、`Message.GetStringProperty` メソッドを使用します。

```
string userID = message.GetStringProperty("JMS_TIBCO_SENDER");
```

## メッセージの送信

メッセージプロデューサクライアントを作成して (333 ページの「[メッセージプロデューサの作成](#)」を参照)、デスティネーションにメッセージを送信できます。メッセージは、メッセージプロデューサで指定されているデスティネーションに送信することも、または、メッセージプロデューサのデスティネーションに `NULL` を設定して、明示的にデスティネーションを指定して送信することもできます。いずれの場合も、各メッセージを送信する際に、オプションで `JMSDeliveryMode`、`JMSExpiration` を設定し、`JMSPriority` メッセージヘッダフィールドを設定できます (17 ページの「[JMS メッセージのヘッダフィールド](#)」を参照)。

以下の各コード例は、それぞれのプログラミング言語でテキストメッセージを送信する 2 通りの方法を示しています。最初の例は、メッセージプロデューサ (333 ページの「[メッセージプロデューサの作成](#)」で作成した `QueueSender`) にメッセージを送信します。2 番目の例は、デスティネーションに `NULL` を指定したメッセージプロデューサ (`NULLsender`) を使用して、331 ページの「[トピックとキューの動的作成](#)」で作成したトピックにメッセージを送信します。

メッセージの作成についての詳細は、[第 2 章](#)、「[メッセージ](#)」を参照してください。

### Java

`MessageProducer` オブジェクトで指定されているデスティネーションにメッセージを送信するには、`MessageProducer` オブジェクトの `send()` メソッドを使用します。

```
QueueSender.send(message);
```

以下の `send()` メソッドの形式を使って、指定されたデスティネーションにメッセージを送信します。

```
MessageProducer NULLsender = session.createProducer(null);
....
NULLsender.send(topic, message);
```

実行可能な例は、`tibjmsMsgProducer.java` サンプルクライアントを参照してください。

### C

`tibemsMsgProducer` で指定されているデスティネーションにメッセージを送信するには、`tibemsMsgProducer_Send` 関数を使用します。

```
status = tibemsMsgProducer_Send(QueueSender, message);
```

指定されたデスティネーションにメッセージを送信するには、`tibemsMsgProducer_SendToDestination` 関数を使用します。

```
status = tibemsMsgProducer_SendToDestination(NULLsender,
                                              topic, message);
```

実行可能な例は、`tibemsMsgProducer.c` サンプルクライアントを参照してください。



Java および C# の API とは異なり、C の API では、デスティネーションが `tibemsMsgProducer` で指定されているかどうかに関わらず、`tibemsMsgProducer_SendToDestination` 関数を使ってデスティネーションを指定できます。

## C#

`MessageProducer` で指定されているデスティネーションにメッセージを送信するには、`MessageProducer.Send` メソッドを使用します。

```
QueueSender.Send(message);
```

以下の `MessageProducer.Send` メソッドの形式を使って、指定されたデスティネーションにメッセージを送信します。

```
MessageProducer NULLsender = session.CreateProducer(NULL);
NULLsender.Send(topic, message);
```

実行可能な例は、`csMsgProducer.cs` サンプルクライアントを参照してください。

## メッセージの受信

336 ページの「[メッセージコンシューマの作成](#)」で作成したメッセージコンシューマは、デスティネーションからメッセージを受け取り、セッションで設定された確認応答モードを使って、メッセージを受信したことを確認応答します (328 ページの「[セッションの作成](#)」を参照)。

メッセージを受信する前に、メッセージコンシューマは EMS サーバーへの接続を開始する必要があります。メッセージコンシューマを終了する前に、EMS サーバーとの接続を切断する必要があります。

以下の例では、326 ページの「[EMS サーバーへの接続](#)」で作成した接続を開始し、331 ページの「[トピックとキューの動的作成](#)」で作成したキューからメッセージを同期的に受信して、接続を切断します。



メッセージコンシューマにメッセージリスナーを実装することで、メッセージを非同期で受信することもできます (337 ページの「[非同期でメッセージをコンシュームするメッセージリスナーの作成](#)」)。



## Java

`Connection` オブジェクトの `start()` メソッドを使って、接続を開始します。

```
connection.start();
```

`MessageConsumer` オブジェクトの `receive()` メソッドを使って、メッセージを受信します。クライアントがメッセージを受信している間は、通常、この方法がループ中で使用されます。

```
Message message = QueueReceiver.receive();
```

メッセージの受信が完了すると、クライアントは `Close()` メソッドを使って接続を切断します。

```
connection.close();
```

実行可能な例は、`tibjmsMsgConsumer.java` サンプルクライアントを参照してください。

## C

`tibemsConnection_Start` 関数を使って、接続を開始します。

```
status = tibemsConnection_Start(connection);
```

`tibemsMsgConsumer_Receive` 関数を使って、メッセージを受信します。クライアントがメッセージを受信している間は、通常、この方法がループ中で使用されます。

```
tibemsMsg message = NULL;
```

```
status = tibemsMsgConsumer_Receive(QueueReceiver,&message);
```

メッセージの受信が完了すると、クライアントは `tibemsConnection_Close` 関数を使って接続を切断します。

```
status = tibemsConnection_Close(connection);
```

実行可能な例は、`tibemsMsgConsumer.c` サンプルクライアントを参照してください。

## C#

`Connection.Start` 関数を使って、接続を開始します。

```
connection.Start();
```

`MessageConsumer.Receive` 関数を使って、メッセージを受信します。クライアントがメッセージを受信している間は、通常、この方法がループ中で使用されます。

```
Message message = QueueReceiver.receive();
```

メッセージの受信が完了すると、クライアントは `Connection.Close` 関数を使って接続を切断します。

```
connection.Close();
```

実行可能な例は、`csMsgConsumer.cs` サンプルクライアントを参照してください。



## 第 12 章

# JNDI を実装した EMS の使用

EMS サーバーには、接続ファクトリやトピックおよびキュー（総称して「管理対象オブジェクト」と呼ぶ）の照合を可能にする JNDI が実装されています。Java クライアントは、標準の JNDI 呼び出しを使用して、EMS に保存されている管理対象オブジェクトを照合できます。C および C# でも、同様の EMS サーバー内のオブジェクトデータを照合するための API 呼び出しが使用できます。

トピックとキューを作成する方法については、71 ページの「[デスティネーションの作成と変更](#)」を参照してください。

### トピック

---

- 「[EMS での管理対象オブジェクトの作成と変更](#)」、348 ページ
- 「[EMS に保存されている管理対象オブジェクトの照合](#)」、350 ページ

## EMS での管理対象オブジェクトの作成と変更

EMS 内に保存する管理対象オブジェクトを作成するには、管理ツールまたは管理 API のいずれかを使用するか、または直接、設定ファイルを編集します。このセクションでは、管理ツールを使用して管理対象オブジェクトを作成する方法を説明します。

接続ファクトリを作成するには、EMS 管理ツールで `create factory` コマンドを実行します。たとえば、`server1` のポート 7344 に TCP 接続を確立する `myFactory` という名前の汎用的な接続ファクトリを作成するには、管理ツールで以下のコマンドを入力します。

```
create factory myFactory generic URL=tcp://server1:7344
```

接続ファクトリのデータは、EMS サーバーの `factories.conf` ファイルに保存されています。EMS サーバーのすべての接続ファクトリを表示する場合は `show factories` コマンドを使用し、ある接続ファクトリの設定の詳細を表示する場合は `show factory` コマンドを使用します。

接続ファクトリには、オプションで、サーバーの負荷分散と接続試行のしきい値を設定するためのプロパティを指定できます（235 ページの「[接続ファクトリのパラメータ](#)」を参照）。`addprop factory`、`setprop factory`、および `removeprop factory` コマンドを使って、ファクトリの作成時にこれらのプロパティを指定したり、既存のファクトリのプロパティを変更することができます。

たとえば、`myFactory` という名前の接続ファクトリへの接続の最大試行回数をデフォルトの 2 回から 5 回に変更するには、管理ツールを起動して、次のように入力します。

```
addprop factory myFactory connect_attempt_count=5
```

値を「2」に戻すには、次のように入力します。

```
setprop factory myFactory connect_attempt_count=2
```

## セキュア接続のための接続ファクトリの作成

このセクションでは、静的な接続ファクトリを作成して、SSL 接続を設定する方法を説明します。接続ファクトリの照合を行う際にも、同じ SSL パラメータの設定を使用する必要があります（[セキュア照合の実行](#)を参照）。

SSL を使用してセキュアな接続を行うには、転送プロトコルに「`ssl`」を指定します。また、214 ページの「[SSL サーバーのパラメータ](#)」に挙げた SSL 設定パラメータをいくつでも指定できます。

たとえば、`server1` のポート 7243 に SSL 接続を確立する `mySecureFactory` という名前の汎用的な接続ファクトリを作成するには、EMS 管理ツールで、次のように入力します。

```
create factory mySecureFactory generic URL=ssl://server1:7243
```

ファクトリを作成し、汎用的な接続をセットアップし、サーバーの証明書をチェックしてサーバーの名前が `myServer` であることを確認するには、次のように（すべて1行に）入力します。

```
create factory MySSLFactory generic url=ssl://7243
ssl_verify_host=enabled ssl_expected_hostname=myServer
ssl_trusted=certs/server_root.cert.pem
```

トピック接続のセットアップ用のファクトリを作成し、サーバーの証明書をチェックし（ただし、証明書内の名前はチェックしない）、接続の作成時にのみクライアントが SSL を使用するように `ssl_auth_only` パラメータを設定するには、次のように（すべて1行に）入力します。

```
create factory AnotherSSLFactory topic url=ssl://7243
ssl_verify_host=enabled ssl_verify_hostname=disabled
ssl_trusted=certs/server_root.cert.pem ssl_auth_only=enabled
```



これらのサンプルコードは、`server_root.cert.pem` 証明書ファイルが、サーバーの実行可能ファイルが格納されているディレクトリの「`certs`」サブディレクトリに配置されていることを前提としています。

詳細は、453 ページの第 18 章、「[SSL プロトコルの使用](#)」を参照してください。

## フォールトトレランス接続のための接続ファクトリの作成

フォールトトレランスクライアントを EMS へ接続する場合は、接続ファクトリに EMS サーバーをふたつ以上指定する必要があります。フォールトトレランスクライアントの接続ファクトリを作成する場合は、`create factory` コマンドの `url` 引数に複数のサーバーの URL を指定します。

たとえば、プライマリサーバー `server0` のポート 7545 とバックアップサーバー `server1` のポート 7344 への TCP 接続を確立する `myFtFactory` という名前の汎用的な接続ファクトリを作成するには、EMS 管理ツールで、次のように（すべて1行に）入力します。

```
create factory myFtFactory generic url=tcp://server0:7545,
tcp://server1:7344
```

`server0` が利用できない場合、クライアントは `server1` に接続します。詳細は、473 ページの第 19 章、「[フォールトトレランス](#)」を参照してください。

## EMS に保存されている管理対象オブジェクトの照合

このセクションでは、名前で EMS サーバー内のオブジェクトを照合する方法について説明します。

すべてのクライアントは、EMS の名前付けサービス内のオブジェクトを照合できます。また、Java アプリケーションは、サードパーティ製の JNDI サーバー内のオブジェクトを照合でき、C と C# クライアントはサードパーティ製の LDAP サーバー内のオブジェクトを照合できます。

EMS に保存されている管理対象オブジェクトを照合するには、名前付けサービスプロバイダの URL およびその他のプロパティ（クライアントをサーバーに認証させるユーザー名およびパスワードなど）を指定する初期コンテキストを作成する必要があります。名前付けサービスプロバイダの URL は、次の形式で指定します。

```
tibjmsnaming://host:port
```

以下の各コード例は TIBCO Enterprise Message Service を使用する際の、JMS 管理対象オブジェクトへアクセスする方法を示しています。各例とも、ConFac という名前の接続ファクトリが `factories.conf` ファイルに、`topic.sample` トピックが `topics.conf` ファイルに、`queue.sample` キューが `queues.conf` ファイルに、それぞれ存在することを前提としています。

### Java

プロバイダのコンテキストファクトリと JNDI プロバイダの URL、およびクライアントを EMS サーバーに認証させるユーザー名とパスワードから構成される初期コンテキストとして `InitialContext` オブジェクトを作成します。

```
Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY,
        "com.tibco.tibjms.naming.TibjmsInitialContextFactory");
env.put(Context.PROVIDER_URL, "tibjmsnaming://localhost:7222");
env.put(Context.SECURITY_PRINCIPAL, "userName");
env.put(Context.SECURITY_CREDENTIALS, "password");
InitialContext jndiContext = new InitialContext(env);
```

初期コンテキスト内にある ConFac という名前の接続ファクトリと `topic.sample` および `queue.sample` という名前のデスティネーションを照合します。

```
ConnectionFactory factory =
    (javax.jms.ConnectionFactory)
    jndiContext.lookup("ConFac");

javax.jms.Topic sampleTopic =
    (javax.jms.Topic)jndiContext.lookup("topic.sample");
javax.jms.Queue sampleQueue =
    (javax.jms.Queue)jndiContext.lookup("queue.sample");
```

EMS\_HOME/samples/java/JNDI ディレクトリにある `tibjmsJNDI.java` サンプルクライアントを参照してください。

**C**

JNDI プロバイダの URL、およびクライアントを EMS サーバーに認証させるユーザー名とパスワードから構成される初期コンテキストとして `tibemsLookupContext` オブジェクトを作成します。

```
tibemsLookupContext* contextstatus = NULL;
status = tibemsLookupContext_Create(
    &context,
    "tibjmsnaming://localhost:7222",
    "userName",
    "password");
```

初期コンテキスト内で、`tibemsLookupContext_LookupConnectionFactory` 関数を使って `ConFac` という名前の接続ファクトリを照合し、`tibemsLookupContext_LookupDestination` 関数を使って `topic.sample` および `queue.sample` という名前のデスティネーションを照合します。

```
tibemsConnectionFactory factory = NULL;
tibemsDestination sampleTopic = NULL;
tibemsDestination sampleQueue = NULL;

status = tibemsLookupContext_Lookup(context,
    "ConFac",
    (void*)&factory);

status = tibemsLookupContext_Lookup(context,
    "sample.queue",
    (void*)&sampleQueue);

status = tibemsLookupContext_Lookup(context,
    "topic.sample",
    (void*)&sampleTopic);
```

**C#**

JNDI プロバイダの URL、およびクライアントを EMS サーバーに認証させるユーザー名とパスワードから構成される初期コンテキストとして `ILookupContext` オブジェクトを作成します。

```
Hashtable env = new Hashtable();
env.Add(LookupContext.PROVIDER_URL,
    "tibjmsnaming://localhost:7222");
env.Add(LookupContext.SECURITY_PRINCIPAL, "myUserName");
env.Add(LookupContext.SECURITY_CREDENTIALS, "myPassword");

LookupContextFactory factory = new LookupContextFactory();
```

```

ILookupContext searcher = factory.CreateContext(
    LookupContextFactory.TIBJMS_NAMING_CONTEXT,
    env);

ILookupContext.Lookup メソッドを使って、初期コンテキスト内で ConFac という
名前の接続ファクトリと topic.sample および queue.sample という名前のデ
スティネーションを照合します。

ConnectionFactory factory =
    (ConnectionFactory) searcher.Lookup("ConFac");

Topic sampleTopic =
    (Topic)searcher.Lookup("topic.sample");

TIBCO.EMS.Queue sampleQueue =
    (TIBCO.EMS.Queue)searcher.Lookup("queue.sample");

```

## 完全 URL 名によるオブジェクトの検索

Java クライアントは、完全 URL 名を使用して、EMS に保存されている管理対象オブジェクトを照合できます。この場合、Context.PROVIDER\_URL プロパティの代わりに、Context.URL\_PKG\_PREFIXES を使用します。例：

```

Hashtable env = new Hashtable();
env.put(Context.URL_PKG_PREFIXES, "com.tibco.tibjms.naming");
env.put(Context.PROVIDER_URL, "tibjmsnaming://localhost:7222");
env.put(Context.SECURITY_PRINCIPAL, "userName");
env.put(Context.SECURITY_CREDENTIALS, "password");
jndiContext = new InitialContext(env);

```

完全 URL 名を使用する場合、次の例のようにオブジェクトを照合できます。

```

Topic sampleTopic = (javax.jms.Topic)jndiContext.lookup(
    "tibjmsnaming://jmshost:7222/topic.sample");
Queue sampleQueue = (javax.jms.Queue)jndiContext.lookup(
    "tibjmsnaming://jmshost:7222/queue.sample");

```

完全 URL 名の使用方法についての詳細は、EMS\_HOME/samples/java/JNDI ディレクトリにある tibjmsJNDIRead.java サンプルを参照してください。



## セキュア照合の実行

TIBCO Enterprise Message Service のクライアントプログラムは、SSL (Secure Sockets Layer) プロトコルを使用して、セキュアな JNDI 照合を実行できます。セキュアな JNDI 照合を実行するには、`InitialContext` の作成時に、接続環境に SSL プロパティを設定する必要があります。SSL プロパティは、TIBCO Enterprise Message Service サーバーの SSL プロパティとよく似ています。TIBCO Enterprise Message Service サーバーでの SSL の使用についての詳細は、[第 18 章、「SSL プロトコルの使用」](#)を参照してください。

以下の例は、SSL プロトコルを使用した JNDI 照合の実行に使用できる `InitialContext` の作成方法を説明しています。

### Java

この例では、`Context.PROVIDER_URL` に指定されているポート番号を、サーバーの設定ファイル `tibjmsd.conf` で指定されている SSL リスポートに設定します。`TibjmsContext.SECURITY_PROTOCOL` の値に `ssl` を設定します。最後に、`TibjmsContext.SSL_ENABLE_VERIFY_HOST` の値を「`false`」に設定して、サーバーの認可をオフにします。これにより、信頼する証明書を提供する必要がなくなり、クライアントは、サーバーの証明書に対して JNDI 照合を行うときに使用するサーバーを検証しなくなります。

```
Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY,
        "com.tibco.tibjms.naming.TibjmsInitialContextFactory");
env.put(Context.PROVIDER_URL, tibjmsnaming://jmshost:7223);
env.put(Context.URL_PKG_PREFIXES, "com.tibco.tibjms.naming")
env.put(TibjmsContext.SECURITY_PROTOCOL, "ssl");
env.put(TibjmsContext.SSL_ENABLE_VERIFY_HOST,
        new Boolean("false"));
Context context = new InitialContext(env);
```

### C

`tibemsSSLParams` オブジェクトを作成し、`tibemsSSLParams_SetIdentityFile` 関数を使って、`pkcs12` ファイルによるクライアントの識別を設定します。`tibemsLookupContext_CreateSSL` 関数を使用して、SSL 接続を使用する `tibemsLookupContext` オブジェクトを初期コンテキストとして作成します。

```
tibemsLookupContext*      context      = NULL;
tibemsConnection_Factory factory      = NULL;
tibemsSSLParams           sslParams    = NULL;
tibems_status             status       = TIBEMS_OK;

sslParams = tibemsSSLParams_Create();
```

```

status = tibemsSSLParams_SetIdentityFile(
    ssl_params,
    "client_identity.p12",
    TIBEMS_SSL_ENCODING_AUTO);

status = tibemsLookupContext_CreateSSL(
    &context,
    "tibjmsnaming://localhost:7222",
    "userName",
    "password",
    sslParams,
    "pk_password");

```

## C#

SSL 接続の初期コンテキストとして `ILookupContext` オブジェクトを作成します。SSL の Store Info は、クライアントを識別する `pkcs12` ファイルとクライアントのパスワードから構成され、`EMSSSLFileStoreInfo` オブジェクトに格納されます。

```

string ssl_identity = client_identity.p12;
string ssl_target_hostname = "server";
string ssl_password = "password";

EMSSSLFileStoreInfo StoreInfo = new EMSSSLFileStoreInfo();
info.SetSSLClientIdentity(ssl_identity);
info.SetSSLPassword(ssl_password.ToCharArray());

Hashtable env = new Hashtable();
env.Add(LookupContext.PROVIDER_URL, "adc1.na.tibco.com:10636");
env.Add(LookupContext.SECURITY_PRINCIPAL, "myUserName");
env.Add(LookupContext.SECURITY_CREDENTIALS, "myPassword");
env.Add(LookupContext.SECURITY_PROTOCOL, "ssl");
env.Add(LookupContext.SSL_TARGET_HOST_NAME,
    ssl_target_hostname);
env.Add(LookupContext.SSL_STORE_TYPE,
    EMSSSLStoreType.EMSSSL_STORE_TYPE_FILE);
env.Add(LookupContext.SSL_STORE_INFO, StoreInfo);

```

## フォールトトレランス照合の実行

TIBCO Enterprise Message Service では、フォールトトレランスの JNDI 照合が実行可能です。プライマリサーバーに障害が起きて、バックアップサーバーがプライマリになると、JNDI プロバイダは自動的に新しいプライマリサーバーを JNDI 照合に使用します。フォールトトレランス JNDI 照合を行うには、`InitialContext` の作成時に、`Context.PROVIDER_URL` プロパティに複数の URL を設定します。プロパティに複数の URL を列挙して指定する場合は、コンマ (,) で区切ります。

## 例

以下のコード例は、`Context.PROVIDER_URL` プロパティに、`emshost` という名前のマシン上にあるプライマリサーバーの URL と、`backuphost` という名前のマシン上にあるバックアップサーバーの URL を設定する場合の例です。

```
env.put(Context.PROVIDER_URL, "tibjmsnaming://jmshost:7222,  
tibjmsnaming://backuphost:7222");
```

最初の EMS サーバーに障害が発生すると、JNDI プロバイダが自動的に `backuphost` ホスト上の EMS サーバーに切り替わり、JNDI 照合を行います。`emshost` が修復されて、再起動すると、バックアップ EMS サーバーに変わります。

## フォールトトレランス JNDI 照合の制限

フォールトトレランス JNDI 照合は、以下のシナリオでは行われません。

- 照合メソッドの引数に完全 URL 名を使用する場合。
- LDAP などの外部の名前付け / ディレクトリサービスに関係付けられているオブジェクトを照合する場合。



## 第 13 章

# マルチキャストの使用法

マルチキャストとは、EMS サーバーで、既存のネットワークを介して複数コンシューマにメッセージを一斉ブロードキャストするメッセージングモデルです。この章では、TIBCO Enterprise Message Service サーバーでのマルチキャストの使用法と設定方法について説明します。

### トピック

---

- [「マルチキャストの概要」、358 ページ](#)
- [「マルチキャストの設定」、362 ページ](#)
- [「マルチキャストの実行」、367 ページ](#)
- [「監視と統計情報」、368 ページ](#)

## マルチキャストの概要

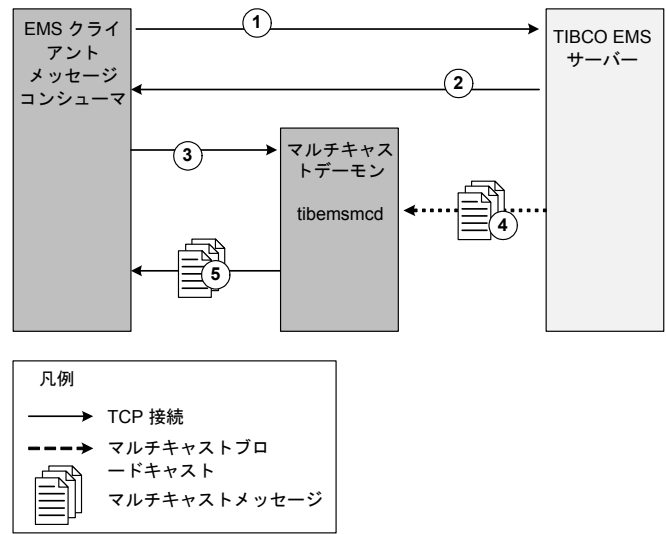
マルチキャストとは、サブスクライバであるコンシューマ宛てにメッセージを個別送信するのではなく、ブロードキャストで多数のコンシューマ宛てに一斉送信するモデルです。TIBCO Enterprise Message Service では、マルチキャスト対応のトピックにパブリッシュされたメッセージを、既存ネットワークを介してブロードキャストするのに PGM を使用します。マルチキャスト対応でないトピックに送信されたメッセージは、TCP を使用してメッセージコンシューマに配信されます。

サーバーは、「マルチキャストチャンネル」経由でマルチキャストメッセージを送信します。マルチキャスト対応のトピックは、それぞれチャンネルに関連付けます。チャンネルによって、サーバーからメッセージを送信する宛先のマルチキャストポートとマルチキャストグループアドレスが決まります。

マルチキャストメッセージは、メッセージコンシューマと同じコンピュータ上で実行される「マルチキャストデーモン」が受信します。EMS クライアントがマルチキャスト対応のトピックをサブスクライブすると、自動的にマルチキャストデーモンに接続します。マルチキャストデーモンはトピックに関連付けられたチャンネルのリッスンを開始し、ブロードキャストメッセージを受信すると、サブスクライバとなっているクライアントに配信します。

図 20 にマルチキャストメッセージのコンシューマ、EMS サーバー、およびマルチキャストデーモンの間の通信フローを示します。

図 20 マルチキャストメッセージのコンシューマの作成



マルチキャストメッセージのコンシューマを作成する手順は、以下のとおりです。

1. EMS クライアントは EMS サーバーに接続すると、ひとつ以上のマルチキャスト対応トピックをサブスクライブします。
2. EMS サーバーは、マルチキャスト デーモン に対する指示と設定情報を含む応答をクライアントに送信します。
3. クライアントはマルチキャスト デーモン に接続し、サーバーから受信した設定情報を渡します。これで、マルチキャスト デーモン はサーバーからのマルチキャストメッセージのリسنを開始します。
4. サーバーがメッセージのブロードキャストを開始すると、マルチキャスト デーモン がそれらを受信します。
5. マルチキャスト デーモン はクライアントにメッセージを配信します。

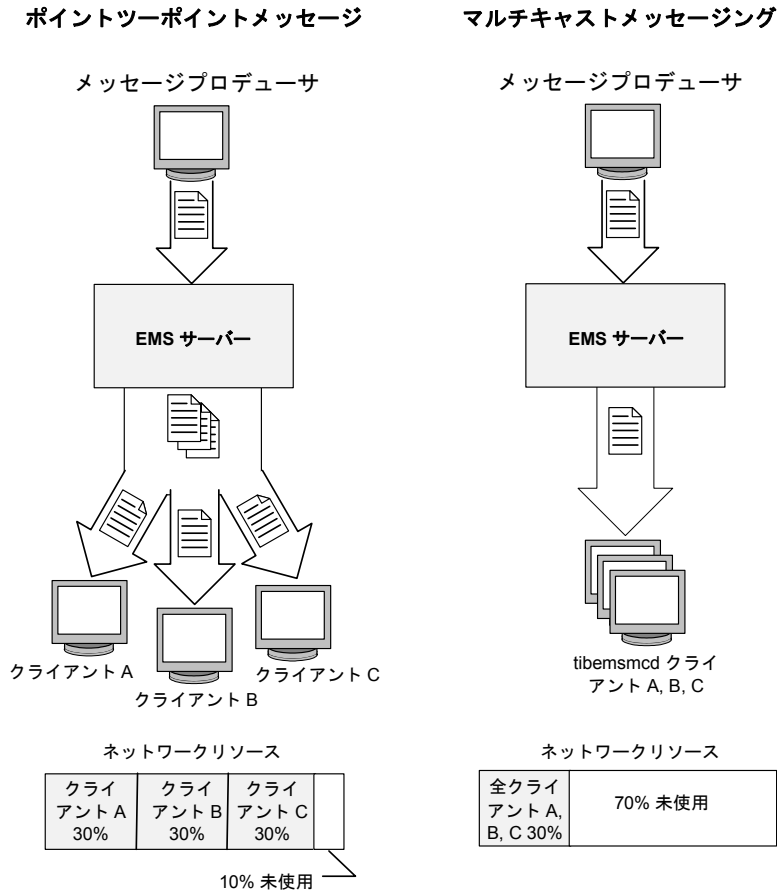
クライアントは、マルチキャストでないメッセージについては、引き続きサーバーから直接受信します。

## マルチキャストの使用目的

マルチキャストは、サーバーで実行する処理数を減少させ、パブリッシュ - サブスクライブ型のモデルに比べて帯域幅の量を低減するため、スケーラビリティの高いモデルです。

360 ページの「[図 21](#)」に、メッセージの送信に使用される帯域幅の量がマルチキャストによって低減される仕組みを示します。パブリッシュ - サブスクライブ型のメッセージングでは、メッセージコンシューマごとにパブリッシュされたメッセージを 1 通作成しますが、マルチキャストではメッセージを 1 回ブロードキャストするだけです。チャンネルをリسنしている複数のマルチキャスト デーモン が同じブロードキャストを受信します。

図 21 マルチキャストの利点



マルチキャストはサーバーで利用するネットワークリソースを削減しますが、あらゆるシステムに最適なメッセージングモデルとは限りません。マルチキャストは、最終ホップの配信にのみ使用し、サーバー間のメッセージ送信には使用できません。ただし、マルチキャスト対応のトピックに送信されたメッセージは、サブスクライバとなっている他のサーバーに、標準の TCP 接続を介して配信されます。



マルチキャストはメッセージの配信を保証しません。高い信頼性が要求されるメッセージには、マルチキャストの使用を避けてください。





マルチキャスト デーモン とメッセージコンシューマは常に同じマシン上に存在し、PGM を使ってブロードキャストメッセージを EMS サーバーからデーモンへ配信します。PGM はセキュリティ機能を持たないため、セキュリティが優先されるアプリケーションにはマルチキャストを使用しないようにします。

## 要件

EMS メッセージングアプリケーションでマルチキャストを使用する場合の要件は、以下のとおりです。

- EMS サーバーでマルチキャストを使用できるように設定する必要があります。
  - サーバーをマルチキャスト対応にする必要があります。
  - マルチキャストチャネルを設定する必要があります。
  - マルチキャスト対応にする必要があるトピックをマルチキャストチャネルに関連付けます。マルチキャストはキューで利用できません。

マルチキャストの設定についての詳細は、362 ページの「[マルチキャストの設定](#)」を参照してください。

- **NO\_ACKNOWLEDGE** モードを使ってマルチキャスト対応メッセージのコンシューマを作成する必要があります。詳細は、38 ページの「[メッセージの確認応答](#)」を参照してください。
- サブスクリバと同じコンピュータ上でマルチキャスト デーモン が実行されている必要があります。詳細は、367 ページの「[マルチキャスト デーモンの起動](#)」を参照してください。

## 下位互換性

マルチキャストは下位互換性があります。EMS クライアントで同一バージョンの TIBCO Enterprise Message Service を使用しないアプリケーションがあっても利用可能です。マルチキャスト経由でメッセージを受信する EMS サーバーおよびクライアントには、ソフトウェアリリース 5.0 以降を使用する必要があります。

マルチキャストの設定は主に EMS サーバーで行われ、大部分は EMS クライアントには表示されません。サーバー側でトピックをマルチキャスト対応に設定するため、マルチキャストメッセージを送信するためにメッセージプロデューサをマルチキャスト対応にする必要はありません。なお、クライアントはデフォルトでマルチキャスト対応です。

マルチキャストが無効化されているか、クライアントが EMS リリース 5.0 より前のリリースを使用しているかのいずれかの理由で、クライアントがマルチキャスト対応でない場合は、メッセージトピックがマルチキャスト対応であっても、クライアントは TCP 接続経由でサーバーからのメッセージを受信します。

## マルチキャストの設定

マルチキャストは、EMS サーバー設定ファイルで定義します。設定は、以下の 3 段階から成る簡単な手順に従います。

1. EMS サーバー内でマルチキャストを有効にします。

**tibemsd.conf** ファイルで **multicast** パラメータを有効にします。マルチキャスト デモンのデフォルトのポートやマルチキャストトラフィックの最大許容量など、マルチキャストのオプションパラメータでその他の設定を制御します。詳細は、207 ページの「**マルチキャストパラメータ**」を参照してください。

2. マルチキャストチャンネルを作成します。

チャンネルに名前を付けて **channels.conf** ファイルで指定します。チャンネル設定ファイルについての詳細は、231 ページの「**channels.conf**」を参照してください。

3. トピックとチャンネルを関連付けます。

**topics.conf** 設定ファイルで、マルチキャスト対応とするトピックの定義に **channel** プロパティを追加します。チャンネルプロパティについての詳細は、56 ページの「**channel**」を参照してください。トピックを関連付けることができるマルチキャストチャンネルはひとつだけです。

## マルチキャストの動的な設定

マルチキャストの設定の大部分は静的な設定です。ランタイム時に変更できるマルチキャスト設定は限られています。EMS サーバーをいったん起動すると、マルチキャスト設定で変更できるのはトピックの **channel** プロパティのみです。マルチキャストチャンネルをトピックに割り当てる、または割り当てを解除するには管理ツールを使用します。**channels.conf** ファイルのチャンネル設定は変更できません。これを変更するには、サーバーを停止する必要があります。

トピックの **channel** プロパティを変更するには、以下のコマンドを使用します。

- **addprop topic** : **channel** プロパティをトピックに追加します。たとえば、次の例はトピック **foo.bar** の **channel** プロパティを **mychannel** に設定します。  
**addprop topic foo.bar channel=mychannel**

ただし、これでトピック **foo.bar** はマルチキャスト対応となりますが、既存のサブスクライバは TCP 経由で引き続きメッセージを受信します。既存のメッセージコンシューマは、コンシューマの停止と再起動を行わない限り、マルチキャスト経由で **foo.bar** に送信されるメッセージを受信することはありません。

- **setprop topic** : **addprop topic** と同様の機能を提供しますが、大きな違いがひとつあります。**setprop topics** を使用すると、その他のプロパティがデフォルトの値にリセットされます。

このコマンドもトピックをマルチキャスト対応にしますが、トピックの既存サブスクライバはマルチキャスト経由でメッセージを受信するようにはなりません。`channel` プロパティを設定した後に作成したメッセージコンシューマのみが、マルチキャストメッセージを受信します。

- **removeprop topic** : トピックから `channel` プロパティを削除します。現在、マルチキャストでメッセージを受信しているサブスクライバは、以降メッセージを TCP 経由で受信します。

サーバーまたはマルチキャスト デーモン にマルチキャストメッセージのバックログが存在する場合、EMS クライアントが受信するメッセージの一部は順序が入れ替わる可能性があります。また、メッセージが失われる場合もあります。マルチキャスト デーモンはバックログが解消するまでキューにあるメッセージの配信を続けますが、EMS サーバーからは新しいメッセージがただちに配信されます。



マルチキャストチャンネルをあるチャンネルから他のチャンネルに変更すると、現在のトピックサブスクライバは、以降メッセージを受信しなくなります。これは次のような場合に起こる可能性があります。

- **addprop topic** または **setprop topic** を使ってチャンネルを明示的に変更した場合。
- **removeprop topic** を使ってチャンネルを削除した場合で、トピックが親トピックから他のチャンネルを継承している場合。

トピックに割り当てられているチャンネルが変更されると、そのトピックの現在のサブスクライバは、該当するトピックを再サブスクライブしない限り、メッセージを受信しません。トピックに他のチャンネルが割り当てられている場合、サーバーから該当クライアントへは TCP 経由でメッセージが送信されません。

通常は、サブスクライバを停止した状態でのみ、チャンネルを変更することをお勧めします。

## マルチキャスト デーモンの設定

マルチキャスト デーモン (`tibemsmcd`) は、マルチキャストメッセージを EMS サーバーから受信し、個別のクライアントに配信するプロセスです。マルチキャスト デーモンはクライアントと同じローカルホストコンピュータで実行されます。ひとつのデーモンで複数のサーバーからメッセージを受信し、複数のクライアントに配信できます。

マルチキャスト デーモンの設定は EMS サーバーで行います。EMS クライアントで作成したマルチキャストメッセージのコンシューマがマルチキャストデーモンに接続すると、サーバーからデーモンに設定が渡されます。デーモンの設定を直接変更する必要がある場合には、コマンドラインオプションを使用します。

たとえば、1 台のコンピュータに複数のネットワークインタフェースが必要な設定の場合は、複数のマルチキャスト デーモン をローカルホスト上で実行できます。デーモンのインタフェースを変更するには、`-ifc` コマンドラインオプションを使用します。詳細は、[コマンドラインオプション](#)を参照してください。

コマンドラインオプション

マルチキャスト デーモン が受け入れるコマンドラインオプションはいくつかあります。`tibemsmcd` サーバーを起動する際に、以下のオプションを指定できます。

表 61 tibemsmcd オプション

オプション	説明
<code>-ifc interface</code>	<p>マルチキャスト デーモン がマルチキャストデータの受信に使用するネットワークインタフェースを識別する IP アドレスを選択します。</p> <p>このオプションを指定しないと、マルチキャスト デーモン は IP アドレス <code>INADDR_ANY</code> で決定されるデフォルトのインタフェースを使用します。</p> <p>複数のインタフェースが必要な設定の場合、各インタフェースにひとつのマルチキャスト デーモンのインスタンスが必要です。利用可能なインタフェースやマルチキャストをサポートしているかどうかを調べるには、Windows の <code>ipconfig</code> コマンドや UNIX システムの <code>ifconfig</code> コマンドが役に立ちます。</p>
<code>-help</code> または <code>-h</code>	ヘルプ画面を表示します。
<code>-logfile file</code>	トレースメッセージの出力先ログファイルを指定します。
<code>-logfile-max-size size[KB MB GB]</code>	<p>トレースメッセージのログファイル最大サイズを指定します。指定したサイズに達すると、ログファイルのローテーションが行われます。デフォルトでは、ログファイルのサイズに制限はありません(ローテーションは行われません)。</p> <p>ゼロは特別な値で、最大サイズを指定しないことを示します。ゼロを指定しない場合は、64KB 以上の値を指定する必要があります。</p>

表 61 tibemsmcd オプション

オプション	説明
<b>-listen</b> [ <i>ip-address</i> :] <i>tcp-port</i>	<p>デーモンがEMSクライアントからの接続をリスンするIPアドレスとTCPポートを変更します。</p> <ul style="list-style-type: none"> <li>• <b>ip-address</b>: (オプション) 指定した場合、特定IPアドレスへのクライアント接続をマルチキャストデーモンが受け入れるインタフェースを限定するパラメータです。<b>ip-address</b>を指定しない場合、マルチキャストデーモンはすべてのインタフェースでEMSクライアントをリスンします。</li> <li>• <b>tcp-port</b>: デーモンがEMSクライアントからの接続をリスンするTCPポートです。デフォルトのポートは7444です。</li> </ul> <p>例:</p> <pre>-listen 127.0.0.1:7444.</pre> <p>デーモンがリスンするデフォルトのTCPポートを変更する場合、同じTCPポートのデーモンに接続を試みるよう、クライアントに指示する必要があります。クライアントで使用するポートを変更するには、<a href="#">tibemsd.conf</a> ファイルの <a href="#">multicast_daemon_default</a> パラメータを設定します。</p>
<b>-trace</b>	<p>マルチキャストデーモンのトレースを有効にします。このオプションを指定すると、クライアントからデーモンへの接続やチャネルの作成などのイベントについてのトレース情報がファイルに書き込まれます。</p>
<b>-max-msg-memory</b> <i>size</i> [MB KB]	<p>コンシューマへの送信待ちメッセージに使用するメモリの最大容量を指定します。最大メモリ制限に達すると、新しいメッセージは破棄されます。MB または GB 単位で指定します。最小許容サイズは、8MB です。</p>
<b>-max-loss-rate</b> <i>percentage</i>	<p>最大許容消失率を1～100のパーセンテージで指定します。消失率が指定した値を上回ると、マルチキャストデーモンはNAKの送信を停止します。</p> <p>デフォルトでは、最大消失率は10%です。</p>
<b>-no-console-trace</b>	<p>マルチキャストデーモンからコンソールへのトレースメッセージの送信を抑制します。</p>

## マルチキャスト対応トピックへのアクセス制御

マルチキャスト対応トピックのパブリッシュおよびサブスクライブの権限は、マルチキャスト対応でないトピックと同様の方法で制御します。デスティネーションへのアクセス制御についての詳細は、266 ページの「[デスティネーション制御](#)」を参照してください。

## マルチキャストの実行

---

マルチキャストメッセージングの例については、98 ページの「[マルチキャストメッセージングの例](#)」を参照してください。

### マルチキャスト デーモンの起動

マルチキャスト デーモン は `installation_path/bin` ディレクトリにインストールされているスタンドアロン型の実行可能ファイルです。UNIX の場合は `tibemsmcd`、Windows の場合は `tibemsmcd.exe` というファイル名です。



Windows が起動しているコンピュータで、スタートメニューから [プログラム] > [TIBCO] > [TIBCO EMS 7.0] > [Start EMS Multicast Daemon] を選択して、EMS マルチキャスト デーモン を起動することもできます。

### マルチキャストコンシューマの作成

EMS クライアントはデフォルトでマルチキャストに対応しており、特別な設定は必要ありません。マルチキャストデータを受信するためにクライアントで実行する必要があるのは、[NO\\_ACKNOWLEDGE](#) モードでマルチキャスト対応のトピックをサブスクライブしてマルチキャストコンシューマを作成することだけです (38 ページの「[メッセージの確認応答](#)」を参照してください)。

クライアントでマルチキャストを無効にするには、API 呼び出しを使用します。詳細は、ご使用の言語の API のマニュアルを参照してください。

## 監視と統計情報

実装済みのマルチキャストをいくつかの局面から分析し、実装の動作状態およびステータスを判断して、トラブルシューティングに役立てることができます。

### 監視

サーバーは、マルチキャスト専用のふたつの監視トピックにメッセージをパブリッシュします。ふたつのトピックとは、`$sys.monitor.multicast.status` と `$sys.monitor.multicast.stats` です。

トピック `$sys.monitor.multicast.status` には、サーバーから監視メッセージがパブリッシュされます。監視メッセージには、マルチキャストコンシューマのステータスと、コンシューマの接続先マルチキャスト デモン についての情報が含まれています。これには、コンシューマがマルチキャストグループに正常に参加したこと、コンシューマでエラーが発生していること（例：マルチキャスト デモン での回復不能な消失）を示す情報などがあります。マルチキャストのエラーを監視することによって、問題が発生しているコンシューマを検出し、是正措置をとることができます。

トピック `$sys.monitor.multicast.stats` には、下位レベルのマルチキャスト統計情報を含む監視メッセージがパブリッシュされます。統計情報には、マルチキャストグループに送信されたバイト数や、マルチキャスト デモン から送信された NAK 数などの情報が含まれます。これらのマルチキャスト統計情報を TIBCO のテクニカルサポートにご提供いただくと、マルチキャスト実装のトラブルシューティングに役立ちます。通常、一般ユーザーにとって、この統計情報はあまり多くの意味を持ちません。マルチキャスト統計情報がパブリッシュされるのは、サーバーの `multicast_statistics_interval` がゼロ以外の値に設定されているときのみです。デフォルトでは、`multicast_statistics_interval` はゼロに設定されています。

監視についての詳細は、[第 17 章](#)、「[サーバーアクティビティの監視](#)」を参照してください。

### 統計情報

サーバーのマルチキャストチャネル統計情報は、管理 API または管理コマンドラインツールを使って表示できます。マルチキャストチャネル統計情報には、以下の情報が含まれます。

- 1 秒あたりの平均送信メッセージ数。
- 1 秒あたりの平均送信バイト数。
- 合計送信メッセージ数。
- 合計送信バイト数。
- チャネルを使用するトピックごとの詳細統計情報。

統計情報についての詳細は、448 ページの「[サーバーの統計情報の操作](#)」を参照してください。



## 第 14 章

## マルチキャストの実装とトラブルシューティング

この章では、マルチキャストを実装する場合の重要な注意事項について解説し、実装時の一般的な問題に対処するためのヒントや推奨事項を説明します。

トピック

---

- 「[実装に関する注意事項](#)」、370 ページ
- 「[マルチキャスト実装の概略](#)」、376 ページ
- 「[EMS のマルチキャストのトラブルシューティング](#)」、385 ページ

## 実装に関する注意事項

マルチキャストを適切に実装するには、通常のユニキャストを実装するときよりも、周到な検討を事前に行う必要があります。このセクションでは、TIBCO Enterprise Message Service にマルチキャストを実装する前に検討する必要がある事項を説明します。

マルチキャストの実装で検討を要する事項は、マルチキャスト接続の確保、マルチキャストトラフィックの制限、および帯域幅の管理の三つの領域に分類できます。これらは、以下の三つの基本的な質問に言い換えることができます。

1. マルチキャストトラフィックが目的のホストのすべてに到達できるか。370 ページの「[接続](#)」を参照してください。
2. マルチキャストトラフィックが目的外のホストに送信されないか。372 ページの「[マルチキャストトラフィックの制限](#)」を参照してください。
3. ユニキャストトラフィックとマルチキャストトラフィックでどのようにネットワーク帯域幅を共用するか。372 ページの「[帯域幅の管理](#)」を参照してください。

### 接続

ユニキャストアプリケーションと同様、マルチキャストアプリケーションでも、マルチキャストデータをセNDERからレシーバへと転送するためのパスを提供するネットワークレイヤが必要です。ただし、マルチキャストの使用と調整のために、ルーターやスイッチで追加の設定が必要な場合があります。接続の確保や制限をする場合の最初の手順は、チャンネルの定義とチャンネルへのマルチキャストグループアドレスの割り当てです。

### マルチキャストアドレス

`channels.conf` 設定ファイルで定義した各マルチキャストチャンネルに、マルチキャストアドレスを割り当てます。TIBCO Enterprise Message Service では、クラス D のアドレス範囲 (224.0.0.0 ~ 239.255.255.255) に属する有効なマルチキャストアドレスを任意に割り当てられます。ただし、IANA (Internet Assigned Numbers Authority) の予約済みアドレスのリストを参照し、競合が起こらないようにしてください。

<http://www.iana.org/assignments/multicast-addresses>

使用するチャンネルにアドレスを割り当てるときは、以下の注意事項にも配慮してください。

- マルチキャストアドレス 224.0.1.78 および 224.0.1.79 は TIBCO EMS の内部使用に予約されています。TIBCO のマルチキャストトラフィックと衝突する可能性がありますので、これらのアドレスは使用しないでください。
- 原則的には、239.0.0.0 ~ 239.255.255.255 の範囲からマルチキャストアドレスを選択してください。これらのアドレスは管理上留保された範囲のブロック

であり、IANA によって予約されることはありません。そのため、外部の競合を気にすることなく、組織内で自由に使用できます。

- MAC アドレスと IP アドレスの間には、1 対 1 のマッピングはありません。したがって、x.0.0.x のアドレスは選択しないでください。選択すると、予約済みアドレスに割り当てられて動作しない場合があります。マルチキャストに割り当てられているクラス D の IP アドレス範囲は 28 ビット幅ですが、マルチキャストに割り当てられている MAC アドレスは 23 ビット幅しかありません。IP アドレスの下位の 23 ビットのみが MAC アドレスを構成するために割り当てられるため、重複が生じます。たとえば、マルチキャストアドレスに 239.0.0.1 を選択すると、予約されている 224.0.0.1 と誤って重複する可能性があります。

## チャンネルの定義

TIBCO Enterprise Message Service では、EMS サーバーやマルチキャスト デーモンで設定して使用できるチャンネル数に制限はありません。ただし、特定のホストが同時に参加できる IP マルチキャストグループアドレスの数は、外部要因によって制約される場合があります。NIC によって数が制限される場合が少なくないものの、一般的に NIC のマニュアルにはこの制限に関する仕様の記載がありません。

実験を行うしか、特定の NIC と OS での制限を確かめる方法がないことがよくあります。一部の NIC では、参加するグループが多すぎると NIC を「無作為検出モード」に設定するため、パフォーマンスに悪影響が出ることがあります。

さらに留意が必要な点があります。チャンネルは IP マルチキャストグループアドレスとデスティネーションポートの両方を表します。そのため、チャンネルとマルチキャストグループは、必ずしも 1 対 1 で関連します。

マルチキャストデーモンが IP マルチキャストグループアドレスをリスンすると、グループに参加したことになります。チャンネルは IP マルチキャストグループアドレスとデスティネーションポートの両方を表します。そのため、チャンネルとマルチキャストグループは、必ずしも 1 対 1 で関連します。たとえば、10 のマルチキャストチャンネルがあり、すべてが同じマルチキャストグループアドレスを異なるポートで使用している場合、マルチキャストデーモンが参加するグループは最大でひとつです。ところが、10 のマルチキャストチャンネルがすべて異なるマルチキャストグループアドレスを使用する場合、マルチキャストデーモンは最大で 10 のグループに参加する可能性があります。

選択したマルチキャスト IP アドレスとポートの組み合わせは、TIBCO EMS でのみ使用するようにしてください。TIBCO のマルチキャストデーモンで破損したネットワークデータをフィルタリングすることは可能ですが、EMS 向けでないデータパケットを受信すると予期しない結果を招き、ネットワークが不安定になる場合があります。

## マルチキャスト接続の確保

前述のとおり、ユニキャストアプリケーションと同様、マルチキャストアプリケーションでも、マルチキャストデータをセNDERからレシーバへと転送するためのパスを提供するネットワークレイヤが必要です。デフォルトでは、ルーターとスイッチのほとんどでマルチキャストルーティングは無効になっているため、有効にするための追加設定が必要です。接続で問題がある場合は、最初にこの設定を確認してください。

たとえば、マルチキャストルーティングを有効にするのに、CISCO のルーターでは `ip multicast-routing` コマンドを実行する必要があります。マルチキャストに使用するハードウェアの設定は、このマニュアルでは説明しません。ご自身のネットワークと組織に固有の設定については、ネットワーク管理者または TIBCO プロフェッショナルサービス部にお問い合わせください。

## マルチキャストトラフィックの制限

また、多くの場合、マルチキャスト実装では、特に、高帯域幅ストリームが複数の低帯域幅リンクも含むネットワーク上に存在する場合や、セキュリティ上の理由からネットワークレイヤーでアクセスが制御されている場所では、マルチキャストストリームが望まない場所に転送されないようにする必要があります。

LAN 内では、Ethernet スイッチはユニキャストトラフィックに必要なポートにのみ転送できます。通常、ルータおよびスイッチは、デフォルトでマルチキャストパケット転送が無効になっているため、マルチキャストトラフィックの制限は問題になりません。ただし、マルチキャスト実装を計画するときは、この問題があることを認識している必要があります。

## 帯域幅の管理

このセクションでは、マルチキャスト実装に固有の帯域幅の注意事項について説明します。帯域幅には次の三つの主要な側面があります。

- 「[使用可能な帯域幅の決定](#)」、373 ページ：使用可能な帯域幅と、パフォーマンスを最大化するための帯域幅制限の設定を決定する。
- 「[チャンネル間での帯域幅の分割](#)」、374 ページ：使用可能な帯域幅を最大限に利用するチャンネルを作成する。
- 「[低速なアプリケーションの処理](#)」、375 ページ：マルチキャストネットワーク全体の速度が低下しないように少数の低速なアプリケーションを管理する。

## 使用可能な帯域幅の決定

TCP などの信頼できるユニキャストトランスポートでは、ユニキャストトランスポートで競合するすべてのセッション全体で自動的にその帯域幅を共有します。管理者がこのプロセスで担う役割はありません。使用可能な帯域幅は、管理者がラウンドトリップ時間およびパケット消失率を測定するときに、プロトコルスタックによって動的に決定されます。このプロセスは、「輻輳制御」と呼ばれます。すべてのストリームの優先度が等しく、それに応じて帯域幅が自動的に分割されることを前提としています。

一方、マルチキャストの場合、ネットワーク配信のための帯域幅の量が確実に予約済みまたは使用可能である状態にするのは管理者の役目です。TIBCO Enterprise Message Service では、管理者が `maxrate` 設定パラメータを使用して各マルチキャストチャネルのネットワーク帯域幅を割り当てます (231 ページの「`channels.conf`」を参照)。帯域幅を正しく割り当ててことは、アプリケーションでの輻輳の発生を防止します。

輻輳はパケット消失の原因になることがあります。その結果、異常な動作やアプリケーション障害も発生する可能性があります。これは、マルチキャストとユニキャストのもうひとつの大きな違いです。ユニキャストでは、輻輳によりアプリケーションの実行速度は低下しますが、障害が発生することはありません。

TIBCO Enterprise Message Service では帯域幅制限が課されないため、送信速度を慎重に検討し、制限する必要があります。ネットワークが実際にデータを配信できる速度より早く送信しようとする、ネットワークが実際に配信できる速度より若干低い速度を要求した場合より、スループットが「大幅に」低下します。

これは多少矛盾していますが、EMS サーバーに 1 Gbps を配信できるネットワークレイヤーを介して 900Mbps を配信するように要求すると、サーバーはこれを実行します。一方、サーバーに 1 Gbps ネットワークレイヤーを介して 1 Gbps を超える速度で配信するように要求すると、わずか 400Mbps の速度しか得られません。この場合、消失率や他の要素に基づいた混乱した動作が発生する可能性が高くなります。

このことから、通常とは異なるルールが適用されます。つまり、「スループットが低すぎる」場合は、より低いスループットを要求すると、「スループットを向上できるチャンスがある」ということです。依然としてスループットが「ワイヤースピード」をかなり下回っている場合でも、このテストを実行することが重要です。つまり、輻輳による消失は、ワイヤースピード制限以外の多くの原因（同一ネットワーク上の TCP データなど）から発生する可能性があります。これは簡単なテストですが、要求する帯域幅の量を減らすと実際のスループットが上がる結果が出る場合は、ネットワークのセンダーとレシーバの間のどこかで、輻輳が原因で消失が発生している確実な兆候といえます。



ネットワークの最大容量を若干「下回る」速度にマルチキャストトラフィックを制限します。スループット速度が予測より遅い場合は、さらにこの速度を制限します。こうすることで、スループットが実際に向上することがわかります。

チャネルのマルチキャストトラフィックの速度を設定するには、231 ページの「`channels.conf`」の `maxrate` パラメータを参照してください。

チャンネルに指定されている帯域幅速度は、ネットワークレイヤーが EMS に対して行う配信の約束と考えることができます。ネットワークレイヤーがこの約束を守れない場合、EMS マルチキャストのスループットは、ネットワークが実際に配信できる速度を大幅に下回る速度まで低下します。

## チャンネル間での帯域幅の分割

理想的には、ルーティング先サブネットのセット、または VLAN 内の実装は同じ速度の異機種混在のインタフェースを持つホストを使用します。これを順守しない実装は、受信インタフェースがリンクおよび送信インタフェースより低速である場合は消失が発生する可能性があるため、お勧めしません。これは、より高速なネットワーク上のデータの突発的増加を、より低速なインタフェースが処理できないために発生します。また、WAN リンクを介して EMS マルチキャストを使用することはお勧めしません。



次の推奨事項では、帯域幅の不一致によるデータ消失を最小限に抑えるうえで役立ちます。

- マルチキャストパブリッシャおよびサブスクライバでは同じ速度のネットワークインタフェースを使用する。
- 最適なマルチキャスト実装は LAN または VLAN を介して行う。

たとえば、100Mb NIC カードを使用しているクライアントがいくつかあり、その他のクライアントは 1 Gb NIC カードを使用している場合、100Mb NIC からより低速なレシーバに送信し、1 Gb NIC からより高速なレシーバに送信するアーキテクチャをお勧めします。これを実現するには、より高速なセNDERおよびレシーバ向けとより低速なセNDERおよびレシーバ向けのふたつのマルチキャストチャンネルを設定します。

または、ひとつのチャンネルを設定し、帯域幅を最も低速なレシーバ、つまり 100Mb に制限できます。ただし、最良の解決策は、マルチホームマシンを使用してふたつのインタフェースに異なるチャンネルを定義することでアプリケーションを分離し、各チャンネルが最適な速度で動作できるようにすることです。

たとえば、次のふたつのチャンネル設定は、100Mb NIC カードおよび 1 Gb NIC カード向けに最適化されています。

```
--- channels.conf ---
[channel_100mb]
  address = 239.1.1.1:10
  maxrate = 7MB
  interface=10.99.99.99

[channel_1Gb]
  Address = 239.1.1.2:10
  maxrate = 95MB
  interface=10.99.99.100
```

100Mb マシンで実行されているアプリケーションでは、`channel_100Mb` が割り当てられているトピックを使用し、1 Gb NIC カードを備えるマシンで実行されているアプリケーションは `channel_1Gb` が割り当てられているトピックを使用します。また、373 ページの「[使用可能な帯域幅の決定](#)」での提案に従って、一部の帯域幅はその他の TCP データ用に残されています。

## 低速なアプリケーションの処理

「低速」であることが判明しているか、または WAN 上のアプリケーションまたはホストが少ないものの、マルチキャストが有効になっているトピックのデータをサブスクライブする必要がある場合は、アプリケーションで EMS マルチキャストを無効にすることをお勧めします。マルチキャストは API 呼び出しを使用してクライアントでマルチキャストを無効にすることができます。ご使用の言語の API のマニュアルを参照してください。

低速なアプリケーションは TCP を介してサーバーからメッセージを受信します。これにより、マルチキャストストリームからメッセージを効果的に削除し、輻輳やその他のマルチキャストレシーバの速度の低下を防止します。マルチキャスト帯域幅の計算で、これを実行するアプリケーションが使用する TCP の帯域幅を考慮することは非常に重要です。



マルチキャスト非対応の EMS クライアントがマルチキャスト対応のトピックをサブスクライブしている場合、メッセージは TCP を介してクライアントに配信されます。帯域幅の制限を設定するときは、373 ページの「[使用可能な帯域幅の決定](#)」の説明に従って、この TCP トラフィックを考慮してください。



## マルチキャスト実装の概略

---

このセクションでは、TIBCO Enterprise Message Service マルチキャスト実装の簡単な例のセットアップに必要な手順について説明します。

- 「[ステップ 1 : マルチキャストネットワークアーキテクチャの設計](#)」、376 ページ
- 「[ステップ 2 : EMS のインストールおよびセットアップ](#)」、378 ページ
- 「[ステップ 3 : ネットワーク機能およびアプリケーション機能の決定](#)」、381 ページ

この例では、マルチキャスト接続が存在し、ネットワーク上の使用可能な帯域幅を把握していることを前提とします。この例では、マルチキャスト実装のすべての側面が網羅されているわけではありません。マルチキャスト実装に適用される一般的な考え方を示しています。

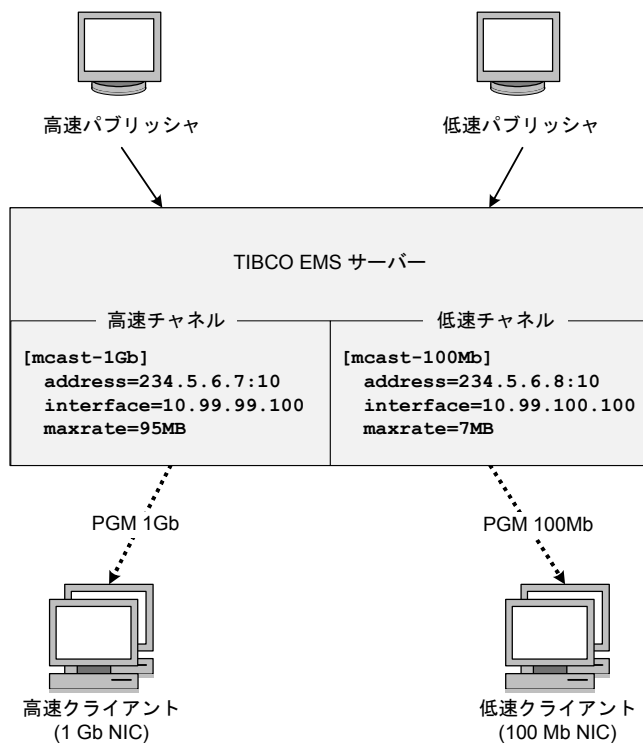
### ステップ 1 : マルチキャストネットワークアーキテクチャの設計

マルチキャスト実装にとって EMS サーバーおよびクライアントの場所は非常に重要です。マルチキャストデータを受信するために、マルチキャストパケットがすべてのネットワークノードに到達できるようにする必要があります。これは、マルチキャストデータが移動するネットワークおよびネットワークセグメント全体のすべての帯域幅を考慮する必要があります。TIBCO Enterprise Message Service は、一般的な接続の問題を検出および報告しますが、通常、接続が確立されているかどうかについては EMS 実装をテストする前に非常に簡単に判定できます。この件に関してはネットワーク管理者に問い合わせてください。

この例では、ふたつのデータのストリームをマルチキャストすることを前提とします。ひとつは、1 Gb ネットワーク上の一部の高パフォーマンスプロセス向けの高速なデータフィードで、もうひとつは、別個の 100Mb ネットワーク上の多数のデスクトップアプリケーション向けのより低速なストリームです。これについては、[図 22](#) に示すアーキテクチャで確認できます。



図 22 サンプルのマルチキャスト実装アーキテクチャ



異なるインタフェースを使用するふたつの別個のチャンネルはサーバーで設定されます。サーバーは高速ギガビットネットワークおよびより低速な 100Mb ネットワークで同時にマルチキャストできます。

## ステップ 2 : EMS のインストールおよびセットアップ

TIBCO Enterprise Message Service のインストールに説明するように、インストールは簡単です。通常の EMS インストール以外に必要な唯一の要件は、マルチキャストデータを受信するすべてのマシンでマルチキャスト デーモン が実行されている必要があるということです。

Windows システムでは、`emsntsrgr` ユーティリティを使用してサービスとしてマルチキャスト デーモン を登録できます。詳細は、106 ページの「`emsntsrgr`」を参照してください。

### EMS サーバーのセットアップ

マルチキャストデータを送信する前に、まず、EMS サーバーを設定する必要があります。EMS サーバーの設定では、`tibemspd.conf` ファイルの複数のグローバル設定を変更し、`channels.conf` ファイルのマルチキャストチャンネルを設定する必要があります。チャンネルの設定後は、`topics.conf` ファイルの `channel` プロパティを設定することによりマルチキャストのトピックを有効にします。

#### サーバーのマルチキャストの有効化

まず、いくつかの一般的な設定を EMS サーバーの主要な設定ファイル `tibemspd.conf` で設定する必要があります。

- `multicast=enabled` を設定して、サーバーでマルチキャストを有効にします。
- `console_trace=+MULTICAST` を設定して、コンソールトレースのマルチキャストを有効にします。

このトレースの有効化は必須ではありませんが、マルチキャスト関連の警告およびエラーが表示されるため、初回の実装時には非常に役に立ちます。

- `flow_control=enabled` を設定しフロー制御を有効にします。

負荷の大きい状況では、サーバーがデータをマルチキャスト送信するより、パブリッシャーの方がより迅速にサーバーにデータを供給できます。フロー制御を有効にすると、サーバーがパブリッシャーでプッシュバックするため、サーバーが遅延した場合、パブリッシャーの速度が低下します。この設定は必須ではありませんが、この問題が発生した場合に消失を最小限に抑えるための余地をサーバーに提供するために、設定することを強くお勧めします。

`tibemspd.conf` に次の行を追加します。

```
multicast=enabled
console_trace= DEFAULT,+MULTICAST
flow_control=enabled
```



また、アーキテクチャでマルチキャストが必要な場合は、サーバーの `startup_abort_list` に `MULTICAST` を追加します。

## マルチキャストチャネルの設定

次のステップでは、マルチキャストチャネルを設定します。この例では、`mcast-1Gb` と `mcast-100Mb` のふたつのマルチキャストチャネルがあります。以下のセクション [channels.conf のサンプル設定](#) では、これらのステップの具体的な設定を示します。

1. `channels.conf` ファイルを作成します。  
このファイルは、[231 ページの「channels.conf」](#) で詳しく説明されています。
2. `channels.conf` ファイルに `[mcast-1Gb]` と `[mcast-100Mb]` のふたつのチャネルを作成します。
3. `address` パラメータを使用して、各チャネルのアドレスおよびデスティネーションポートを設定します。
4. `interface` パラメータを使用して各チャネルのインタフェースを設定します。  
この例では、サーバーはマルチホームマシン上にあります。そのため、各チャネルのインタフェースを明示的に指定する必要があります。インタフェースが指定されていない場合、EMS サーバーではデフォルトのインタフェースが使用されます。これは、マルチキャスト デーモン にも該当します。マルチホームマシンでマルチキャスト デーモン を実行する場合は、[364 ページの「コマンドラインオプション」](#) の説明に従って、`-ifc` コマンドラインパラメータを使用します。
5. 各チャネルの `maxrate` を設定します。

`maxrate` パラメータでは、サーバーがチャネルを介してメッセージを送信する速度を制限します。最大速度を決定する方法の説明については、以下の [最大速度の推定](#) を参照してください。

### channels.conf の サンプル設定

チャネル設定が完了すると、`channels.conf` ファイルには次の行が含まれています。

```
[mcast-1Gb]
  address=239.1.1.1:10
  interface=10.99.99.99
  maxrate=112MB

[mcast-100Mb]
  address=239.1.1.2:10
  interface=10.99.99.100
  maxrate=8MB
```

**最大速度の推定**

この例では、ネットワーク容量を推定できるように、任意のネットワーク使用率の数値を使用して、`maxrate` プロパティを設定しています。`maxrate` の推定に使用するプロセスは、次のとおりです。

まず、予測されるマルチキャストデータを含まない、平均のネットワーク使用率を確認します。これには、基準データ速度の測定が前提となります。

- 1 Gb ネットワークでは、約 10% の使用率を想定すると、900Mb が使用可能です。
- 100Mb のネットワークでは、30% の使用率を想定すると、70Mb が使用可能です。

このことから、ネットワークの秒あたりの使用可能なバイト数を計算します。

- $900\text{Mb} \times 1\text{byte}/8\text{bits} \approx 112\text{ MB}$  (切り捨て)
- $70\text{Mb} \times 1\text{byte}/8\text{bits} \approx 8\text{ MB}$  (切り捨て)

これらの最初の速度はテスト用であるため、パフォーマンスを最大化するために後で変更されます。マルチキャストパフォーマンスの基本ルールは、「速度を向上させるためには速度を低下させる」必要がある場合があるということです。速度が速すぎると消失が発生しやすくなります。つまり、メッセージが再送信され、実際の速度はネットワークの処理能力をはるかに下回るまで低下します。



この例では、1 ネットワークあたり 1 チャネルのみを使用します。アーキテクチャで複数のマルチキャストグループ（異なるアドレスプロパティが設定されている複数のチャネル）を使用している場合、最大帯域幅の計算にネットワークのすべてのチャネルを含めます。そのため、チャネル全体でデータ速度をある程度分散する必要があります。

**マルチキャストトピックの設定**

チャネルの定義後は、サーバーがマルチキャストを使用してトピックをサブスクライブしているマルチキャスト対応のコンシューマにメッセージを送信できるように、トピックの `channel` プロパティを設定する必要があります。チャネルプロパティは `topics.conf` 設定ファイルで設定されます。

この例では、`feed-1Gb` と `feed-100Mb` のふたつのトピックを使用します。これらのトピック名は任意です。重要なのは、正しいチャネルをトピックに割り当てることです。

`topics.conf` のサ  
ンプル

```
feed-1Gb channel=mcast-1Gb
feed-100Mb channel=mcast-100Mb
```

## EMS クライアントのセットアップ

EMS クライアントがマルチキャストデータを受信するには、次のふたつの主要な要件があります。

- クライアントがマルチキャストトピックをサブスクライブする場合、NO\_ACKNOWLEDGE の確認応答モードを使用する必要があります。詳細は、367 ページの「[マルチキャストコンシューマの作成](#)」を参照してください。
- マルチキャスト デーモン はクライアントと同じコンピュータで実行されている必要があります。367 ページの「[マルチキャスト デーモンの起動](#)」を参照してください。

また、TIBCO Software は、アプリケーションがマルチキャスト関連のイベント、エラー、および警告が通知されるように、マルチキャスト例外リスナーを利用することを強くお勧めします。これは、以下の java コードに示すふたつの単純なステップで実行されます。

1. まず、[TibjmsMulticastExceptionListener](#) を実装するクラスを作成します。

```
class MulticastExceptionHandler implements
com.tibco.tibjms.TibjmsMulticastExceptionListener
{
    public void onMulticastException(Connection connection,
                                     Session session,
                                     MessageConsumer consumer,
                                     JMSException e)
    {
        System.out.println(e.getMessage());
    }
}
```

2. 次に、マルチキャスト例外リスナーを設定します。理想的には、これは、マルチキャスト対応のトピックのコンシューマを作成する前に実行します。

```
com.tibco.tibjms.Tibjms.setMulticastExceptionListener(new MulticastExceptionHandler());
```

C API を使用してマルチキャスト例外リスナーをセットアップするには、TIBCO Enterprise Message Service C & COBOL API リファレンスを参照してください。

.NET API を使用してマルチキャスト例外リスナーをセットアップするには、HTML 文書インタフェースを使用して入手できる TIBCO Enterprise Message Service .NET API リファレンスを参照してください。

## ステップ 3 : ネットワーク機能およびアプリケーション機能の決定

ネットワークが処理できる EMS のデータ速度を把握することは重要です。アプリケーションがネットワークで対応可能な速度と少なくとも同じ速度でデータを処理できる場合、ネットワークがスループットのボトルネックとなる状況は異例です。つまり、このようにデータの速度が要件を満たしている限り、これは、理想的です。

## ネットワーク機能の決定

サーバーが有効になっているため、チャンネルで指定されている `maxrate` をテストして、微調整することができます。このセクションでは、設定をテストするひとつの方法について説明します。

この例では、ネットワーク上のメッセージのマルチキャストが少なく、メッセージあたり平均 100 バイトであると想定します。

次のステップでは、ネットワーク帯域幅設定のテスト方法を説明します。

1. 103 ページの「[オプションを使用した EMS サーバーの起動](#)」の説明に従って、`-trace FLOW` オプションを使用して、EMS サーバーを起動します。
2. コマンドラインから、`tibemsmcd -trace` コマンドを使用してマルチキャストデーモンを起動します。

`-trace` オプションの使用は必須ではありませんが、問題の検出に役立つ場合があります。詳細は、367 ページの「[マルチキャストデーモンの起動](#)」を参照してください。

3. マルチキャストデータを受信する各ノードで、コマンドラインウィンドウを開き、`TIBCO_HOME/ems/7.0/samples/java` フォルダに移動します。
4. EMS に含まれる `tibjmsPerfSlave` サンプルプログラムを起動します。

```
> java tibjmsPerfSlave -server serverURL
```

マルチキャストデータを受信するすべてのノードで `jmsPerfSlave` アプリケーションを実行することは非常に重要です。EMS マルチキャストは、最も低速のレシーバのレベルで実行するように調整する必要があります。これを行わない場合、輻輳または消失が発生する可能性があります。

5. マルチキャストデータをパブリッシュするノードごとに、次を起動します。

```
> java tibjmsPerfMaster -topic feed-1Gb -channel mcast-1Gb
-ackmode NO -time 30 -size 100
```

または

```
> java tibjmsPerfMaster -topic feed-100Mb -channel mcast-100Mb
-ackmode NO -time 30 -size 100
```

これらのパフォーマンスアプリケーションはパブリッシャが実行されるノードごとに実行します。

6. サーバーおよびマルチキャストデーモンの出力で警告またはエラーを確認します。消失を示すトレースメッセージが表示されるか、大幅な速度の変更が発生する場合は、通常、選択されている最大速度を超えている可能性があります。

たとえば、マルチキャストエラーは、次のようになります。

```
channel='mcast-1Gb', Loss Detected, status=IO failed
```

サーバーでは、一般的に次のように表示されます。

```
2008-11-13 17:11:57.300 Multicast channel 'mcast-100Mb' has
exceeded its allotted bandwidth
```

フロー制御およびフロートレースが有効になっていると、次のように表示されます。

```
2008-11-13 17:11:57.781 Flow control engaged on topic
'feed-100Mb'
....
```

フロー制御が有効になっている場合、これは、単にサーバーがパブリッシャでプッシュバックし、マルチキャストチャネルによって定義されている速度まで低下することを意味します。

トレースメッセージがマルチキャストチャネルがその帯域幅を超えたことを示している場合、これは、チャネルの最大速度が低すぎることで、つまり、パブリッシャがチャネルの `maxrate` が許可する速度を超えてパブリッシュしていることを示しています。一方、`maxrate` が高すぎる場合、消失が検出されたことを示すエラーが表示されます。

トレースメッセージが示す内容によっては、チャネルの最大速度を引き上げたり、引き下げたり (`maxrate` プロパティ) して調整し、このテストを繰り返します。

## マルチキャストレシーバアプリケーションの評価

マルチキャスト実装の成功のひとつの鍵は、EMS サーバーでアプリケーションのデータのオーバーランが発生しないようにすることです。これは、多くの場合、ネットワークおよび EMS のみが処理できる速度を下回る速度に、配信速度 (チャネルの `maxrate` プロパティ) を設定することを意味します。



チャネルの最大配信速度、つまり `maxrate` が、最も低速のメッセージのコンシューマが受信メッセージをコンシュームできる速度を超えないようにする必要があります。

最も低速なアプリケーションが処理できる速度をメッセージの最大速度にすることで、試行錯誤のテストに費やす時間を削減できます。アプリケーションでネットワークの配信速度を超えてデータを処理できる場合、ネットワーク機能を決定することで最大速度が決まります。

概して、最も低速のアプリケーションが受信データを処理できる最大速度の判定は、試行錯誤のプロセスです。コンシュームのアプリケーションの最大速度をプログラムを使用して判定すると便利がよくあります。マルチキャストデーモンは、低速アプリケーション用にメッセージをバッファしますが、これにより、マルチキャストデーモンのデータのレイテンシーおよびメモリ使用率が増加します。これは、持続可能な状況ではありません。

マルチキャスト対応のコンシューマが時折遅延することが予期され、データが消失する可能性がある場合は、トピックの `maxbytes` プロパティおよび `maxmsgs` プロパティを使用してこれに対処することができます。これらのプロパティについての詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

## チャンネルパラメータの調整

ネットワーク機能およびマルチキャストレシーバの速度を決定したら、最大スループットを達成するためにチャンネルの `maxrate` プロパティを引き上げ（または引き下げ）を試行することができます。自分の環境で処理できるマルチキャストの最大速度を見つけるには、何よりも実験を繰り返すことが必要です。ネットワークが満杯になるとスループットが大幅に低下することを常に念頭においてください。

## オペレーティングシステムの調整

残念ながら、通常、オペレーティングシステムは `raw socket` では高いパフォーマンスには調整されません。可能なパフォーマンスの変更は幾つかあります。通常、これらの変更にはソケットバッファリングが含まれ、これによりスループットの大幅な向上を実現できます。

たとえば、Linux では、`/etc/sysctl.conf` ファイルのウィンドウのサイズを変更できます。

```
net.core.wmem_max=1073741824
net.core.rmem_max=1073741824
net.core.wmem_default=1073741824
net.core.rmem_default=1073741824
```

ただし、オペレーティングシステムのマルチキャスト向けの調整については、このドキュメントでは説明していません。TIBCO プロフェッショナルサービス部では、TIBCO Enterprise Message Service に固有の高度な調整に関する支援を提供できます。また、ネットワークパフォーマンス関連のオペレーティングシステムの一般的な調整に関するリソースは、インターネットに多数存在します。

## 開発環境および本番環境

マルチキャストの設定は、特定のネットワークに固有であり、設定では、ネットワークに固有のノードのトラフィックパターンおよび特徴を考慮する必要があります。結果的に、開発環境に適用されるパラメータの調整が本番環境では最適ではない可能性があります。また、その逆の場合もあります。ある環境から他の環境に移行する場合、アプリケーションおよび EMS アーキテクチャパターンが同一である可能性があります。ネットワークおよびアプリケーション機能については、このセクションで説明するステップを繰り返して再評価する必要があります。トピックおよびチャンネルの定義名を同じままにして、マルチキャストの速度、インタフェース、タイムアウトのパラメータを再評価します。

実装時に再評価する必要のあるチャンネルのプロパティは、次のとおりです。

- [232 ページの「maxrate」](#)
- [232 ページの「ttl」](#)
- [233 ページの「interface」](#)



## EMS のマルチキャストのトラブルシューティング

---

マルチキャスト実装の問題は、ユニキャストの類似の問題以上に解決が困難なことがよくあります。解決がより困難である理由には次のものが含まれます。

- 旧型ネットワーク機器がマルチキャスト実装を考慮しないで設計されている。たとえば、マルチキャストをフラッドさせることしかできないスイッチ、最新のマルチキャストルーティングプロトコルを実装していないルータなどです。
- 機器によっては同じ問題でも解決方法が異なる場合がある。たとえば、一部のスイッチでは IGMP スヌーピングを使用しますが、他のスイッチでは CGMP を使用します。
- マルチキャスト診断ツールがまだ使用できない。
- ネットワーク管理者のマルチキャスト実装の問題での経験が、ユニキャスト実装に関する経験ほど十分ではない。
- 帯域幅は、競合するユニキャストストリーム間で自動的に均等に共有されますが、目的のマルチキャスト帯域幅共有を実現するために管理者の介入が必要な場合があります。

### トラブルシューティングのヒント

このセクションでは、マルチキャスト実装で経験することがある困難に対応するうえで役立つ、トラブルシューティングのヒントを挙げます。

#### 一般的なヒント

実装で問題が発生する場合は、まず次の対策から実行します。

- 一般的には、「ボトムアップ」アプローチが最適です。つまり、ネットワークスタックの最も下のレイヤーから始めます。
- まず、EMS サーバーから、各スイッチおよびルータを経由してすべてのレシーバまでトレースします。受信アプリケーションをサーバーと同じハブ（スイッチまたはルータではない）に移動して、マルチキャスト接続できているかどうか確認します。マルチキャスト接続が機能していたら、より複雑なマルチキャストネットワークに移ります。

## 接続

EMS では、マルチキャスト接続の問題を検出します。接続の問題の検出には最大 64 秒かかる場合があります。次の推奨事項は接続の問題の解決に役立つ可能性があります。

- マルチキャスト接続の問題に取り組む前に、ネットワークでセNDERおよびすべてのレシーバ間に適切なユニキャスト接続があるかを確認します。
- IP マルチキャストがサポートされており、ルータまたはスイッチ、および使用されているすべてのネットワークインタフェース上で有効になっていることを確認します。
- ルータのアドレス範囲指定がマルチキャストパケットの転送を妨害していないか確認します。
- EMS サーバーでマルチキャストを有効にしないで、マルチキャストアプリケーションをテストすることにより、より一般的なトピックまたはアプリケーション設定の問題がメッセージの受信を妨害していないかを特定します。たとえば、不正なトピックをコンシュームしているコンシューマなどです。
- サーバーのマルチキャストおよびトピックトレースを有効化して、適切に設定されていることを確認し、メッセージがサーバーによってマルチキャストされていることを確認します。
- マルチキャスト デーモン のトレースメッセージを有効にして、設定の問題、警告、またはエラーを確認します。
- サーバーおよびマルチキャスト デーモン で正しいインタフェースを使用していることを確認します。マルチホームホストでは、デフォルトのインタフェースがサーバーからマルチキャストデータを受信できない可能性があります。
- チャネルの `t+1` が、データがすべてのスイッチおよびルータを移動するのに十分な大きさであるかを確認します。

## データ消失

次の推奨事項は、データ消失が発生した場合に役立つ可能性があります。

- データが配信されているかどうか、および過剰な消失が発生しているかどうかを確認するために、統計情報を有効化して確認します。消失が検出された場合、マルチキャストチャネルの `maxrate` プロパティを引き下げると状況が緩和されることがあります。
- マルチキャストストリームが、ネットワーク経由で可能な限り最も長いパスを使用してデスティネーションに到達できる十分な存続期間のあるメッセージが生成されていることを確認します。
- マルチキャスト速度の引き上げにともなって消失が増える場合は、ブロードキャスト速度を制限するように設定されている可能性のあるルータまたはスイッチを検索します。これらは、一般的にマルチキャスト速度も制限します。たとえば、Cisco Catalyst 5000 シリーズスイッチは、`set port broadcast` コマンドを使用して、秒あたりのパケットまたはブロードキャスト / マルチキャストトラフィックの割合を制限するように設定できます。

## アプリケーションおよびマルチキャスト デーモンのエラーおよび警告

次のヒントはマルチキャスト デーモン またはクライアントアプリケーションでエラーが発生した場合に役立ちます。

- 受信アプリケーションでマルチキャスト例外リスナーを登録します。これにより、アプリケーションにマルチキャストの警告およびエラーを検出、ログ記録および処理する方法が提供されます。

また、クライアントトレースがサーバーで有効化されている場合、マルチキャストイベントはクライアントで記録されますが、パフォーマンスが犠牲になり、他の問題を発生させる原因になります。このため、基本的な接続の問題のデバッグ以外で、または、TIBCO サポートによる指示のない限り、クライアントトレースの使用はお勧めしません。

- 通常、マルチキャストが有効になっているトピックのコンシューマでコンシューマの作成が失敗すると、マルチキャスト デーモンのログ（またはコンソール）だけでなく、サーバーログにもメッセージが書き込まれます。クライアントでも適切な例外またはリターンコードが呼び出しから生成されます。マルチキャスト以外の関連するその他の理由（セキュリティ、一般的な設定）が排除されたら、次の項目を確認してください。

— マルチキャスト デーモン が実行されているか。

— マルチキャスト デーモン が正しいポートで実行されているか。

— マルチキャスト デーモン でチャンネルの作成が失敗したか（これは、プロトコルレベルのマルチキャストの問題を示しています）。

- マルチキャスト デーモン で過剰な消失が検出されている場合、アプリケーションでマルチキャスト接続例外 `IO Failed` が生成されます。通常、これは、サーバーの送信速度が速すぎるため、チャンネルの `maxrate` を引き下げる必要があることを意味します。マルチキャスト デーモン は次のようなエラーを報告します。

```
2007-10-02 16:45:09.551 Multicast error: channel='mcast', Loss
Detected, status=IO failed
```

また、マルチキャスト統計情報では特定のチャンネルの `rcv_losses` が増大していることもわかります。

- コンシューマが `Timeout reached which may indicate a configuration or hardware problem` のようなメッセージを含む `TIBEMS_TIMEOUT` のマルチキャスト例外を受信する場合、これは、マルチキャスト接続が欠落していることを示しています。ユニキャスト接続がクライアントおよびサーバー間に存在し、マルチキャストチャンネルが設定されていても、マルチキャストデータをサーバーからローカルのマルチキャスト デーモン に取得することはできません。この問題の検出に 1 分以上かかる場合があります。
- マルチキャスト関連の統計情報を受信するため、サブスクライバの `$sys.monitor.multicast.stats` 監視メッセージのリスンを開始します。

## サーバーエラー

一般的に、サーバーエラーは自己記述型です。重要なのは、クライアントエラーがログ対象のサーバーに返されるため、マルチキャストエラーを一箇所で検索できることです。ただし、これらのエラーには特定のクライアントでの小規模な消失またはクライアントフェイルオーバーからのメッセージの消失は含まれません。

## 第 15 章

# TIBCO Rendezvous との相互運用

この章では、EMS と TIBCO Rendezvous の相互運用について説明します。

### トピック

---

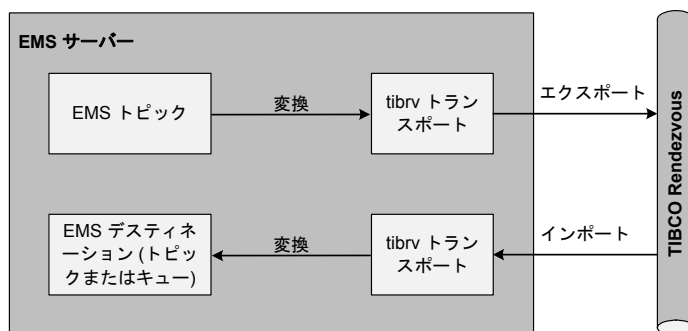
- 「概要」、390 ページ
- 「[Rendezvous のためのトランスポートの設定](#)」、392 ページ
- 「[トピック](#)」、398 ページ
- 「[キュー](#)」、400 ページ
- 「[インポートに関連する問題](#)」、402 ページ
- 「[エクスポートに関連する問題](#)」、404 ページ
- 「[メッセージの変換](#)」、405 ページ
- 「[Pure Java Rendezvous プログラム](#)」、411 ページ

## 概要

TIBCO Enterprise Message Service (リリース 4 以降) は、TIBCO Rendezvous (リリース 6.9 以降) とメッセージをやり取りできます。

- 範囲
- EMS は、EMS の「トピック」を通して、外部システムとの間でメッセージをインポートおよびエクスポートできます。
  - EMS は、外部システムから EMS の「キュー」にメッセージをインポートできます (ただし、キューはエクスポートを行えません)。

図 23 EMS サーバー内の Rendezvous トランスポート



## メッセージの変換

EMS と Rendezvous は、それぞれメッセージとデータに独自の形式を使用しています。tibemspd でメッセージのインポート / エクスポートを行う場合、メッセージとデータを適切な形式に変換します。詳細は、426 ページの「[メッセージの変換](#)」を参照してください。

## 設定

tibemspd では、四つの設定ファイルに定義およびパラメータ設定を行って、Rendezvous とメッセージを交換できるようにします。

- 有効化
- パラメータ `tibrv_transports` (`tibemspd.conf` 設定ファイル) は、Rendezvous とのメッセージのやり取りをグローバルに有効化または無効化します。デフォルトの値は、`disabled` です。これらのトランスポートを使用するには、このパラメータを明示的に「`enabled`」に設定する必要があります。

トランスポート	<p>トランスポート定義 (<a href="#">transports.conf</a> 設定ファイル) は、EMS と外部システム間の通信プロトコルを指定します。詳細は、392 ページの「<a href="#">Rendezvous のためのトランスポートの設定</a>」を参照してください。</p>
デスティネーション	<p>デスティネーション定義 (設定ファイル <a href="#">topics.conf</a> および <a href="#">queues.conf</a>) は、<a href="#">import</a> および <a href="#">export</a> プロパティを設定して、ひとつ以上のトランスポートを選択できます。</p> <ul style="list-style-type: none"><li>• <a href="#">import</a> を設定すると、<a href="#">tibemsd</a> は、<a href="#">Rendezvous</a> からこれらのトランスポートに送信されたメッセージをインポートし、EMS のデスティネーションにメッセージを配信します。</li><li>• <a href="#">export</a> を設定すると、<a href="#">tibemsd</a> は、EMS のデスティネーションに届いたメッセージを取り出し、選択されているトランスポート経由で、<a href="#">Rendezvous</a> へメッセージをエクスポートします。</li></ul> <p>詳細は、421 ページの「<a href="#">トピック</a>」および 422 ページの「<a href="#">キュー</a>」を参照してください。</p>
RVCМ リスナー	<p>保証メッセージ配信が設定されたトランスポート上のメッセージをエクスポートする場合、事前に <a href="#">tibrvcм.conf</a> ファイルに RVCМ リスナーを登録できます。</p> <p>詳細は、247 ページの「<a href="#">tibrvcм.conf</a>」および 404 ページの「<a href="#">保証メッセージ</a>」を参照してください。</p>

## Rendezvous のためのトランスポートの設定

トランスポートは、EMS と TIBCO Rendezvous との間でやり取りされるメッセージのフローを調整します。

timemsd は、他の Rendezvous クライアントと同じ方法で Rendezvous デーモンに接続します。トランスポート定義 ([transports.conf](#) ファイル) は、これらの接続の動作を設定します。これらのトランスポートを正しく設定する必要があります。

### Rendezvous メッセージのインポート方法

EMS サーバーは、その他すべての Rendezvous クライアントと同じ方法で Rendezvous デーモンに接続します。Rendezvous デーモンから受信したメッセージは、Rendezvous キューに保存されてから、コールバックにディスパッチされます。EMS サーバーは Rendezvous メッセージの JMS メッセージのコピーを作成し、EMS メッセージとしてメッセージのコピーの処理を開始します。トランスポートは、メッセージをインポートする方法を決定します。

トランスポートを介してインポートされる Rendezvous メッセージは、そのトランスポートに固有のキューに保持されます。各トランスポートは、異なる Rendezvous キューに関連付けられます。この Rendezvous キューには、必要な数だけ Rendezvous メッセージが保存されます。Rendezvous メッセージの受信速度が EMS サーバーが対応する EMS メッセージを処理できる速度を超えている場合、キューの保留メッセージ数が増加します。

トランスポートに定義されているインポート配信モードによっては、EMS メッセージはディスク上に保持されます。そのため、Rendezvous キューのバックログが増える可能性があります。つまり、EMS プロセスのメモリ使用率が増大します。このメモリ使用率の増大は、EMS サーバーの統計情報では一切考慮されていません。

### キューの制限ポリシー

Rendezvous キューの保留メッセージ数を制限するために、TIBCO Rendezvous クライアントアプリケーションの場合と同様に、トランスポートプロパティではキューの制限ポリシーを設定できます。トランスポートのキューの制限に到達していない場合、Rendezvous ライブラリでは、設定された数のメッセージが破棄されます。デフォルトのポリシーは、TIBRVQUEUE\_DISCARD\_NONE です。メッセージがまったく破棄されないことを意味します。TIBRVQUEUE\_DISCARD\_FIRST または TIBRVQUEUE\_DISCARD\_LAST を設定することにより、選択した破棄ポリシーを適用する前に、キューで保留できる Rendezvous メッセージの最大数を指定できます。制限に達したときに、破棄されるメッセージ数は、破棄量の値に基づきます。

制限に到達すると、Rendezvous メッセージが破棄されるため、EMS インポート配信モードに関わらず、EMS メッセージとしてインポートされません。前述のように、Rendezvous メッセージは、Rendezvous キューからディスパッチされた後に限り、EMS メッセージになります。キューの制限を超えると、信頼できる Rendezvous メッセージが消失します。



Rendezvous 保証メッセージは消失しませんが、メッセージフローが中断されません。送信に失敗したメッセージの再送信は Rendezvous ライブラリによって自動的に処理されますが、EMS サーバーでは制御できません。

また、キューの制限に到達すると、ログに記録される Rendezvous アドバイザリ (TIBCO Rendezvous のマニュアルの RVADV ログおよびコンソールトレースを参照) が生成され、トランスポートがキューの制限に到達したことを示します。このアドバイザリは、独立した制限のない Rendezvous キューに入ります。多数のアドバイザリが生成されると、この内部キューも増大し、制限ポリシーが環境に適していないことを通知します。

キューの制限ポリシーの設定は慎重に行ってください。EMS サーバーに大きな負担をかける Rendezvous プロデューサのリスクが低く抑えられ、制御された環境では、キューの制限ポリシーを設定する必要はありません。

## トランスポート定義

`transports.conf` には、ゼロ個以上のトランスポート定義が含まれています。各定義は、角かっこ ([ ]) で囲まれたトランスポートの名前で始まります。それに続く行で、トランスポートのパラメータを設定します。

表 62 Rendezvous : トランスポートのパラメータ (1 / 4 シート)

パラメータ	説明
<code>type</code>	必須。Rendezvous トランスポートの場合、値は <code>tibrv</code> または <code>tibrvcn</code> のいずれかです。
<b>Rendezvous パラメータ</b>	
<code>tibrv</code> トランスポートまたは <code>tibrvcn</code> トランスポートのいずれかで、これらのプロパティを使用します。	
これらのパラメータの構文とセマンティクスは、Rendezvous クライアントの対応するパラメータと同じです。詳細は、Rendezvous のマニュアルセットを参照してください。	
<code>service</code>	パラメータを指定しなかった場合のデフォルトの値は 7500 です。
<code>network</code>	このパラメータを指定しなかった場合、デフォルト値はホストコンピュータのプライマリネットワークです。

表 62 Rendezvous : トランスポートのパラメータ ( 2 / 4 シート)

パラメータ	説明
daemon	<p>指定しなかった場合、デフォルト値はローカルホストコンピュータ上の <code>rvd</code> プロセスです。EMS と Rendezvous 間でメッセージをトランスポートする場合、<code>rvd</code> プロセスが、EMS デーモン (<code>tibemsd</code>) と同じホスト上で実行されるように設定する必要があります。</p> <p>デフォルト以外のデーモンに接続するには、<code>hostname:protocol:port</code> と入力します。この三つの部分のいずれも省略可能です。デフォルトのホスト名はローカルのホストコンピュータです。デフォルトの <code>protocol</code> は <code>tcp</code> です。デフォルトの <code>port</code> は 7500 です。</p>

Rendezvous 保証メッセージ (RVCM) のパラメータ	
これらのパラメータは、 <code>tibrvcn</code> トランスポートの場合のみ使用します。	
これらのパラメータの構文とセマンティクスは、Rendezvous CM クライアントの対応するパラメータと同じです。詳細は、Rendezvous のマニュアルセットを参照してください。	
cm_name	対応する RVCM リスナーのトランスポートの名前。
rv_tport	必須。各 RVCM トランスポートは、通常の Rendezvous トランスポートのひとつを順番に使用します。このパラメータに、EMS 設定ファイル <code>transports.conf</code> で設定されている Rendezvous トランスポート (タイプ <code>tibrv</code> ) の名前を設定します。
ledger_file	元帳ファイルの名前。
sync_ledger	<code>true</code> または <code>false</code> 。 <code>true</code> を指定した場合、元帳を更新する処理は、変更がストレージメディアに書き込まれるまで制御が戻りません。
request_old	<code>true</code> または <code>false</code> 。 <code>true</code> を指定した場合、このトランスポートサーバーは、このトランスポートが利用できなかった間に、他の RVCM センダーから送信された、まだ確認応答されていないメッセージを要求します。
default_ttl	このパラメータは、このトランスポート上でエクスポートされるすべての CM メッセージに対して、デフォルトの CM 制限時間 (秒単位) を設定します。

表 62 Rendezvous : トランスポートのパラメータ (3 / 4 シート)

パラメータ	説明
<code>explicit_config_only</code>	<p><code>true</code> または <code>false</code>。<code>true</code> を指定した場合、事前に (<code>tibrvcn.conf</code> ファイルまたは <code>create rvcmlistener</code> コマンドで) EMS サーバーで設定されている RVCN リスナーのみ、<code>tibemsd</code> で保証配信への登録が許可されます。言い換えれば、<code>tibemsd</code> は、設定されていないリスナーからの登録要求を無視します。</p> <p><code>false</code> (デフォルト) の場合、<code>tibemsd</code> は、すべての RVCN リスナーからの登録要求を受け付けます。</p>
<b>EMS パラメータ</b>	
<code>tibrv</code> トランスポートまたは <code>tibrvcn</code> トランスポートのいずれかで、これらのプロパティを使用します。	
<code>topic_import_dm</code> <code>queue_import_dm</code>	<p>EMS 送信クライアントは、各メッセージで <code>JMSDeliveryMode</code> ヘッダフィールドを設定できます。Rendezvous クライアントはこのヘッダを設定できません。代わりに、これらのふたつのパラメータを使用して、<code>tibemsd</code> がこのトランスポート上にインポートする、すべてのトピックメッセージとキューメッセージの配信モードを決定します。</p> <p><code>TIBEMS_PERSISTENT</code>   <code>TIBEMS_NON_PERSISTENT</code>   <code>TIBEMS_RELIABLE</code></p> <p>このパラメータを指定しなかった場合、デフォルトは <code>TIBEMS_NON_PERSISTENT</code> です。</p>
<code>export_headers</code>	<p><code>true</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> ヘッダフィールドを設定します。</p> <p><code>false</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> ヘッダフィールドを設定しません。</p> <p>このパラメータを指定しなかった場合、デフォルト値は <code>true</code> です。</p>
<code>export_properties</code>	<p><code>true</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> プロパティを設定します。</p> <p><code>false</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> プロパティを設定しません。</p> <p>このパラメータを指定しなかった場合、デフォルト値は <code>true</code> です。</p>

表 62 Rendezvous : トランSPORTのパラメータ ( 4 / 4 シート)

パラメータ	説明
rv_queue_policy	<p>トランSPORTが受信 Rendezvous メッセージの保持に使用する Rendezvous キューの、キュー制限ポリシーを設定します。このパラメータは次の三つの部分があります。</p> <p><i>policy:max_msgs:qty_discard</i></p> <p>ここで、<i>policy</i> は以下で説明するキューの制限ポリシーのひとつで、<i>max_msgs</i> は、キューに保持できるメッセージの最大数で、それを超えると破棄されます。また、<i>qty_discard</i> は <i>max_msgs</i> に到達したときに EMS サーバーが破棄するメッセージ数です。</p> <p>キューの制限ポリシーは次のとおりです。</p> <ul style="list-style-type: none"><li>• TIBRVQUEUE_DISCARD_NONE : メッセージを破棄しない。キューが保持できるメッセージ数を制限しない場合は、このポリシーを使用します。</li><li>• TIBRVQUEUE_DISCARD_FIRST : キューの最初のメッセージを破棄する。キューの最初のメッセージは最も古いメッセージです。これが破棄されない場合、次のメッセージがキューからディスパッチされます。</li><li>• TIBRVQUEUE_DISCARD_LAST : キューの最後のメッセージを破棄する。最後のメッセージはキューで受信した最新のメッセージです。</li></ul> <p>たとえば、次のポリシーでは、キューのメッセージの合計数が 10,000 個に到達すると、Rendezvous ライブラリによってキュー内の最も古い 100 個のメッセージが破棄されます。</p> <p><code>rv_queue_policy=TIBRVQUEUE_DISCARD_FIRST:10000:100</code></p> <p><code>rv_queue_policy</code> を指定しない場合、デフォルトのキューの制限ポリシーは TIBRVQUEUE_DISCARD_NONE です。</p>
temp_destination_timeout	<p>最後にデスティネーションが使用されてから、サーバーがテンポラリデスティネーション (RV 受信箱用に作成) が保持する時間を指定します。これはマルチサーバー構成に使用します。たとえば、<code>rv-requester -&gt; serverA -&gt; serverB -&gt; rv-responder</code> の構成になっている場合、<code>serverB</code> に <code>temp_destination_timeout=60</code> を設定すると、<code>serverB</code> にテンポラリデスティネーションが 60 秒間保持されます。</p>

## 例

以下の例は、`transports.conf` ファイルの一部で、トランスポート定義の構文を示しています。

```
[RV01]
  type = tibrv
  topic_import_dm = TIBEMS_RELIABLE
  queue_import_dm = TIBEMS_PERSISTENT
  service = 7780
  network = lan0
  daemon = tcp:host5:7885
```

```
[RV02]
  type = tibrv
  service = 7890
  network = lan0
  daemon = tcp:host5:7995
  temp_destination_timeout = 60
```

```
[RVC01]
  type = tibrvcm
  export_headers = true
  export_properties = true
  rv_tport = RV02
  cm_name = RVC0Trans1
  ledger_file = ledgerFile.store
  sync_ledger = true
  request_old = true
  default_ttl = 600
```

次のふたつの例では、RVC03 はキューの制限ポリシーを定義していない RVC0 トランスポートです。ただし、キューの制限ポリシーが設定されている RV トランスポート RV03 を参照しています。Rendezvous メッセージが EMS でデステーションプロパティ `import=RVC03` が設定されているサブジェクトにパブリッシュされる場合、各トランスポートが独自のキューを使用しているため、Rendezvous メッセージはまったく破棄されません。10000 個のメッセージというキューの制限に到達した場合、RV03 トランスポートを使用して直接インポートされるメッセージのみが破棄される可能性があります。

```
[RV03]
  type = tibrv
  service = 7890
  network = lan0
  daemon = tcp:host5:7995
  rv_queue_policy = TIBRVQUEUE_DISCARD_LAST:10000:100
```

```
[RVC03]
  type = tibrvcm
  rv_tport = RV03
  cm_name = RVC0Trans2
  ledger_file = ledgerFile2.store
  sync_ledger = true
  request_old = true
  default_ttl = 600
```

## トピック

トピックはメッセージのエクスポートとインポートが可能です。したがって、`topics.conf` 設定ファイルのトピックの定義で、`import` と `export` のプロパティを設定して、外部トランスポートをひとつ以上選択できます。

- `import`
  - `import` を設定すると、`tibemsd` は、Rendezvous からこれらのトランスポートに送信されたメッセージをインポートし、EMS のデスティネーションにメッセージを配信します。
- `export`
  - `export` を設定すると、`tibemsd` は、EMS のデスティネーションに届いたメッセージを取り出し、選択されているトランスポート経由で、Rendezvous へメッセージをエクスポートします。



EMS サーバーは、同じトピックにインポートされたメッセージを再エクスポートしません。

(`topics.conf` の構文とセマンティクスについての一般的な情報は、[247 ページの「topics.conf」](#)を参照してください。管理ツールの `addprop topic` コマンドを使ってもトピックを設定できます。)

**例** たとえば、トピック `myTopics.news` を設定して RV01 と RV02 トランスポート上のメッセージをインポート (`import`) し、RV02 トランスポート上のメッセージをエクスポート (`export`) するには、`tibemsdadmin` コマンドを使って次のように入力します。

```
addprop topic myTopics.news import="RV01,RV02"
addprop topic myTopics.news export="RV02"
```

`myTopics.news` というサブジェクトを持つ Rendezvous メッセージが、RV01 および RV02 のトランスポート経由で、`tibemsd` に届きます。EMS クライアントは、`myTopics.news` をサブスクライブすることで、これらのメッセージを受け取ることができます。

`myTopics.news` に送信された EMS メッセージは、RV02 トランスポート経由で Rendezvous にエクスポートされます。対応するデーモンの Rendezvous クライアントは、`myTopics.news` をサブスクライブすることで、これらのメッセージを受け取ることができます。

## サブスクライバが存在する場合のみのインポート

接続中のトランスポート上で、トピックが `import` を指定すると、`tibemsd` は、そのトピックにサブスクライバが登録されている場合のみメッセージをインポートします。

## ワイルドカード

`import` および `export` プロパティに使用するワイルドカードは、EMS の構文とセマンティクス (Rendezvous の構文とセマンティクスと同じ) に従います (420 ページの「[デスティネーション名：構文とセマンティクス](#)」を参照)。

## 保証メッセージ

TIBCO Rendezvous 保証メッセージ (`tibrvc` トランスポート) を EMS トピックにインポート (`import`) およびエクスポート (`export`) することができます。Rendezvous 保証トランスポートはメッセージの配信を保証します。

**RVC**M 元帳 `tibrvc` トランスポートは、サブジェクトに関する情報を元帳ファイルに保存します。管理ツールのコマンドを使って元帳ファイルの中身を確認できます ([160 ページ](#)の「`show rvctransportledger`」を参照)。

元帳ファイルについての詳細は、TIBCO Rendezvous のマニュアルを参照してください。

**サブジェクトの競合** `RVC`M トランスポートからインポートするデスティネーションのサブスクライバには、`RVC`M リスナーを指示する制限と同じ制限が適用されます。TIBCO Rendezvous マニュアルで説明されているこれらの制限には、サブジェクトの競合も含まれます。

`RV` からメッセージをインポートすると、EMS サーバーはトランスポートごとにひとつの名前を使用して `RVC`M リスナーを作成します。これにより、対応する EMS サブスクライバに重複するトピックがある場合、サブジェクトの競合が発生する可能性があります。

## キュー

---

キューは、メッセージをインポート ([import](#)) ですが、エクスポート ([export](#)) できません。

### 設定

`queues.conf` 設定ファイルのキュー定義で、`import` プロパティを設定して、外部トランスポートをひとつ以上選択できます。

- [import](#) を設定すると、`tibemsd` は、Rendezvous からこれらのトランスポートに送信されたメッセージをインポートし、EMS のデスティネーションにメッセージを配信します。

(`queues.conf` の構文とセマンティクスについての一般的な情報は、[240 ページの「queues.conf」](#)を参照してください。管理ツールの `addprop queue` コマンドを使ってもキューを設定できます。)

例   たとえば、キュー `myQueue.in` を設定して RV01 と RV02 トランスポート上のメッセージをインポート (`import`) するには、`tibemsadmin` コマンドを使って次のように入力します。

```
addprop queue myQueue.in import="RV01,RV02"
```

`myQueue.in` サブジェクトの Rendezvous メッセージが、RV01 および RV02 のトランスポート経由で、`tibemsd` に届きます。EMS クライアントは、`myQueue.in` をサブスクライブすることで、これらのメッセージを受け取ることができます。

### インポート：開始および停止

接続中のトランスポート上のキューで [import](#) を指定すると、`tibemsd` は、キューのレシーバが存在しない場合でも、直ちにキューへのメッセージのインポートを開始します。

管理者によって静的に設定されたキューの場合は、管理者がキューを明示的に削除するまで、`tibemsd` はインポート処理を継続します。

### ワイルドカード

[import](#) プロパティに使用するワイルドカードは、EMS の構文とセマンティクス (Rendezvous の構文とセマンティクスとは異なる) に従います (420 ページの「[デスティネーション名：構文とセマンティクス](#)」を参照)。



EMS クライアントは、ワイルドカードキューをサブスクライブすることはできません。ただし、EMS サーバーでワイルドカードキューを定義して、プロパティを継承させることができます。たとえば、`foo.*` という名前の静的キューを設定して、そのプロパティを設定することで、`foo.bar` および `foo.baz` という名前の子のキューの両方にそれらのプロパティを継承させることができます。

`foo.*` をインポートするキューを定義すると、`tibemsd` は、このワイルドカードのパターンに一致する、`Rendezvous` 内のすべてのメッセージのインポートを開始します。メッセージが届くと、`tibemsd` は動的に子のキュー（`foo.bar` および `foo.baz`）を作成し、メッセージをこれらのキューに配信します。そのメッセージを取得するサブスクライバがない場合でも、`tibemsd` は動的に作成されたこれらの子のキューにメッセージを配信するので注意が必要です。

## インポートに関連する問題

---

このセクションでは、Rendezvous から EMS への、トピックまたはキューのメッセージのインポートに関連する問題を取り上げます。

### フィールド ID

Rendezvous メッセージをインポートして、変換するときは、tibemspd は Rendezvous プログラムアプリケーションの名前で識別される標準のメッセージフィールドタイプのみを処理することができます。フィールド ID を使用して識別されるカスタムフィールドおよびフィールドは EMS にインポートできません。

### JMSDestination

tibemspd が、Rendezvous メッセージをインポートして変換すると、EMS メッセージの **JMSDestination** フィールドが Rendezvous サブジェクトの値に設定されます。したがって、インポートされるデスティネーション名には固有の名前を付ける必要があります。トピックとキューが同じ名前を使用している場合、そのどちらか一方のみで **import** プロパティを指定できます。たとえば、トピック `foo.bar` とキュー `foo.bar` を定義すると、どちらか一方でのみ **import** プロパティを指定できます。

### JMSReplyTo

tibemspd が、Rendezvous メッセージをインポートして変換すると、EMS メッセージの **JMSReplyTo** フィールドが Rendezvous の応答サブジェクトの値に設定され、EMS クライアントがメッセージに応答できるようになります。

通常、この値には Rendezvous サブジェクトが示されます。メッセージを Rendezvous にエクスポートするのに使用される、対応する名前のトピックを生成するためには、tibemspd を明示的に設定する必要があります。

### JMSExpiration

tibemspd が、Rendezvous 保証メッセージをインポートして変換すると、EMS メッセージの **JMSExpiration** フィールドに保証メッセージの有効期限が設定されます。

メッセージがこの有効期限を過ぎた場合、セnderのプログラムは配信を保証しません。

**expiration** プロパティをデスティネーションで設定すると、メッセージプロデューサが設定した JMSExpiration の値が無効にされます。

## JMSTimestamp

`tibemsd` が、Rendezvous メッセージをインポートして変換すると、JMSTimestamp ヘッダフィールドを使用してメッセージの作成時刻が判定されます。JMSTimestamp フィールドが設定されていない場合、有効期限が不明な作成時刻に基づくことになるため、`tibemsd` は有効期限フィールドを無視します。

Rendezvous のセNDERは、メッセージの有効期限を有効にするために、JMSTimestamp と呼ばれるフィールドを作成する必要があります。

## 保証配信



Rendezvous から EMS への、エンドツーエンドの完全な配信を保証するには、以下の三つの条件をすべて満たしている必要があります。

- Rendezvous センダーは RVCN トランスポート上で、ラベル付きメッセージを送信する必要があります。詳細は、『TIBCO Rendezvous 概要』のマニュアルを参照してください。
- トランスポート定義で、`topic_import_dm` または `queue_import_dm` (状況に応じて選択) を `TIBEMS_PERSISTENT` に設定する必要があります。
- EMS トピックの持続キューまたはサブスクライバのいずれかが存在する必要があります。

## エクスポートに関連する問題

このセクションでは、EMS から Rendezvous へのメッセージのエクスポートに関連する問題を取り上げます。

### JMSReplyTo

**トピック** EMS メッセージの JMSReplyTo フィールドにトピックが含まれているとします。このようなメッセージを Rendezvous へエクスポートする場合、tibemspd がその応答トピックへの応答を Rendezvous からインポートするよう、明示的に設定する必要があります。

**テンポラリトピック** EMS メッセージの JMSReplyTo フィールドにテンポラリトピックが含まれているとします。tibemspd がこのようなメッセージを Rendezvous にエクスポートする場合、そのテンポラリトピックへの応答を「自動的に」Rendezvous からインポートするようになっています。したがって、明示的に設定する必要はありません。

### 保証メッセージ

**RVCМ 登録** RVCМ リスナーは、最初のラベル付きメッセージを受信すると、それ以降のメッセージを保証メッセージとして受け取るように登録します。登録処理が完了するまで、RVCМ リスナーはラベル付きメッセージを信頼メッセージとして受信します。tibrvcm トランスポート上でメッセージをエクスポートする場合、次のふたつのいずれかを実行し、エクスポートされるすべてのメッセージについて保証配信が行われるようにすることをお勧めします。

- EMS クライアントからメッセージを送信する前に、RVCМ リスナーを作成する。
- 管理ツール (126 ページの「[create rvcmlistener](#)」を参照) または設定ファイル tibrvcm.conf (247 ページの「[tibrvcm.conf](#)」を参照) を使用して、RVCМ リスナーをあらかじめ登録しておく。

### 保証配信



EMS から Rendezvous への、エンドツーエンドの完全な配信を保証するには、次の条件を満たすことが要求されます。

- EMS センダーは永続的メッセージを送信する必要がある。

## メッセージの変換

### JMS ヘッダフィールド

EMS は、10 種類の定義済みの JMS ヘッダフィールドをサポートします (17 ページの「[JMS メッセージのヘッダフィールド](#)」を参照)。

- 特別なケース      以下のヘッダフィールドは、特殊なケースです。
- JMS ヘッダ **JMSDestination** は Rendezvous のサブジェクトに対応しています。
  - JMS ヘッダ **JMSReplyTo** は Rendezvous の応答サブジェクトに対応しています。
  - JMS ヘッダ **JMSExpiration** は Rendezvous 保証メッセージの有効期限に対応しています。

インポート      Rendezvous メッセージを EMS メッセージにインポートする場合は、上記の特殊なケースを除いて、tibemsd は、JMS ヘッダフィールドを設定しません。

エクスポート      EMS メッセージを Rendezvous メッセージにエクスポートする場合、tibemsd は、(上記の特殊なケースを除き) すべての JMS ヘッダフィールドを Rendezvous メッセージ内のひとつのサブメッセージにまとめます。JMSHeaders フィールドにそのサブメッセージが含まれます。サブメッセージの各フィールドにより、JMS ヘッダフィールドの名前がその値にそれぞれマッピングされます。

tibemsd は null であるか、値が指定されていない JMS ヘッダフィールドは無視します。つまり、エクスポートされるメッセージから除外します。

トランスポートプロパティを `export_headers = false` と設定することで、すべてのエクスポートされるメッセージについて、ヘッダサブメッセージを完全に出力しないように tibemsd に指示することができます。

表 63 に、JMS ヘッダフィールドと Rendezvous データ型 (エクスポートされたメッセージ内の対応するフィールドのタイプ) との対応関係を示します。

表 63 Rendezvous : JMS ヘッダフィールドと RV データ型との対応関係  
(1 / 2 シート)

JMS ヘッダ名	Rendezvous タイプ
JMSDeliveryMode	TIBRVMSG_U8
JMSPriority	TIBRVMSG_U8
JMSTimestamp	TIBRVMSG_U64
JMSExpiration	TIBRVMSG_U64
JMSType	TIBRVMSG_STRING
JMSMessageID	TIBRVMSG_STRING

表 63 Rendezvous : JMS ヘッダフィールドと RV データ型との対応関係  
( 2 / 2 シート)

JMS ヘッダ名	Rendezvous タイプ
JMSCorrelationID	TIBRVMSG_STRING
JMSRedelivered	TIBRVMSG_BOOL
JMSDestination	TIBCO Rendezvous の send サブジェクト
JMSReplyTo	TIBCO Rendezvous の reply サブジェクト

JMS プロパティフィールド

インポート      Rendezvous メッセージを EMS メッセージにインポートした場合、tibemsd は以下の JMS プロパティを設定します。

- **JMS\_TIBCO\_IMPORTED** は true に設定され、メッセージが EMS クライアントから送信されたものでないことを示します。
- **JMS\_TIBCO\_MSG\_EXT** は true に設定され、メッセージにサブメッセージフィールドまたは配列フィールドが含まれている「可能性」があることを示します。

RVCМ のインポート      前述のふたつのフィールドに加え、tibemsd が tibrvcm トランスポート上の保証メッセージをインポートするときに、以下のプロパティが設定されることがあります (Rendezvous メッセージに対応する情報が設定されている場合)。

表 64 Rendezvous の対応するメッセージのプロパティ

プロパティ	説明
JMS_TIBCO_CM_PUBLISHER	メッセージを送信した TIBCO Rendezvous CM トランスポートの通信プログラム名 (すなわち、セNDER名) を表す文字列の値。
JMS_TIBCO_CM_SEQUENCE	TIBCO Rendezvous からインポートした RVCМ メッセージの CM シーケンス番号を表す long 値。

エクスポート      Rendezvous メッセージへ EMS メッセージをエクスポートする場合、tibemsd は、すべての JMS プロパティフィールドを Rendezvous メッセージ内のひとつのサブメッセージにまとめます。JMSProperties フィールドにそのサブメッセージが含まれます。サブメッセージの各フィールドにより、JMS プロパティフィールドの名前がその値にそれぞれマッピングされます。

`tibemspd` デーモンは、`null` であるか、値が指定されていない JMS プロパティフィールドは無視します。つまり、エクスポートされるメッセージから除外します。

トランスポートプロパティを `export_properties = false` と設定することで、エクスポートされるメッセージで、プロパティサブメッセージを完全に出力しないように `tibemspd` に指示することができます。

## メッセージ本文

`tibemspd` は、どのようなメッセージ本文タイプを持った JMS メッセージでも TIBCO Rendezvous にエクスポートできます。反対に、`tibemspd` は、どのようなメッセージタイプのメッセージでも TIBCO Rendezvous からインポートできます。

JMS の本文タイプについての詳細は、21 ページの「[JMS メッセージ本文](#)」を参照してください。

メッセージの構造についての詳細は、17 ページの「[JMS メッセージの構造](#)」を参照してください。

インポート Rendezvous メッセージをインポートすると、`tibemspd` は [表 65](#) に示したフィールドの表示に基づいて、そのメッセージを EMS メッセージの本文タイプに変換します。

表 65 Rendezvous : 対応するメッセージタイプ (インポート)

Rendezvous フィールド	EMS 本文タイプ
JMSBytes	JMSBytesMessage
JMSObject	JMSObjectMessage
JMSStream	JMSStreamMessage
JMSText	JMSTextMessage
どのフィールドも存在しない。	JMSMapMessage



フィールド名 `DATA` および `_data_` は、予約されています。特に EMS および Rendezvous アプリケーションのメッセージトランスポートメカニズムを統合する場合、これらのフィールド名をこのふたつのアプリケーションのいずれにも使用しないことを強くお勧めします。



EMS には、名前で識別される標準の Rendezvous フィールドのみをインポートできます。Rendezvous アプリケーションのフィールド ID で識別されるカスタムフィールドおよびフィールドはインポートできません。

エクスポート EMS メッセージをエクスポートすると、`tibemspd` はそのメッセージを以下の構造を持つ Rendezvous メッセージに変換します。

- `JMSHeaders` フィールドにサブメッセージが含まれます (405 ページの「[JMS ヘッダフィールド](#)」を参照)。トランスポートパラメータ `export_headers` に `false` を設定すると、このフィールドは省略されます。

- JMSProperties フィールドにサブメッセージが含まれます (406 ページの「JMS プロパティフィールド」を参照)。トランスポートパラメータ export\_properties に false を設定すると、このフィールドは省略されます。
- EMS メッセージのデータフィールドを変換する場合、JMS の本文タイプによって変換結果が異なります。表 66 にその対応関係を示します。

表 66 Rendezvous: 対応するメッセージタイプ (エクスポート)

JMS 本文タイプ	エクスポートによる変換。
BytesMessage	メッセージデータは、元の EMS メッセージの各バイトを含むバイト配列に変換されます。  JMSBytes フィールドがこのデータを受け取ります。 TIBRVMSG_OPAQUE タイプを持ちます。
ObjectMessage	メッセージデータは、シリアル化された Java オブジェクトを含むバイト配列に変換されます。  JMSObject フィールドがこのデータを受け取ります。 TIBRVMSG_OPAQUE タイプを持ちます。
StreamMessage	メッセージデータは、元の EMS メッセージのオブジェクトをエンコードしたバイト配列に変換されます。  JMSStream フィールドがこのデータを受け取ります。 TIBRVMSG_OPAQUE タイプを持ちます。
TextMessage	メッセージデータは、元の EMS メッセージのテキストに対応する UTF-8 の文字列に変換されます。  JMSText フィールドがこのデータを受け取ります。 TIBRVMSG_STRING タイプを持ちます。
MapMessage	メッセージデータフィールドは、Rendezvous メッセージの最上位のフィールドに直接マッピングされます。フィールドは、元の EMS メッセージと同じ名前を保持します。  16 ページの「EMS の機能を拡張した JMS メッセージ」も参照してください。



データ型

表 67 は、EMS のデータ型と Rendezvous のデータ型の対応関係を示します。対応する EMS のデータ型がない Rendezvous のデータ型を除き（これらのタイプは、表 67 の中央のカラムに片方向としてマークされています）、マッピングは双方向です。

表 67 Rendezvous: データ型の対応関係（1 / 2 シート）

EMS	マッピング 方向	Rendezvous
Boolean		TIBRVMSG_BOOL
Byte		TIBRVMSG_I8
Short	<—	TIBRVMSG_U8
Short		TIBRVMSG_I16
Integer	<—	TIBRVMSG_U16
Integer		TIBRVMSG_I32
Long	<—	TIBRVMSG_U32
Long		TIBRVMSG_I64
Long	<—	TIBRVMSG_U64
Float		TIBRVMSG_F32
Double		TIBRVMSG_F64
Short	<—	TIBRVMSG_IPPORT16
Integer	<—	TIBRVMSG_IPADDR32
MapMessage		TIBRVMSG_MSG
Long	<—	TIBRVMSG_DATETIME
byte[]		TIBRVMSG_OPAQUE
java.lang.String		TIBRVMSG_STRING
byte[]	<—	TIBRVMSG_XML
byte[]	<—	TIBRVMSG_I8ARRAY

表 67 Rendezvous: データ型の対応関係 （2 / 2 シート）

EMS	マッピング方向	Rendezvous
short[]	<—	TIBRVMSG_U8ARRAY
short[]		TIBRVMSG_I16ARRAY
int[]	<—	TIBRVMSG_U16ARRAY
int[]		TIBRVMSG_I32ARRAY
long[]	<—	TIBRVMSG_U32ARRAY
long[]		TIBRVMSG_I64ARRAY
long[]	<—	TIBRVMSG_U64ARRAY
float[]		TIBRVMSG_F32ARRAY
double[]		TIBRVMSG_F64ARRAY

## Pure Java Rendezvous プログラム

TIBCO Enterprise Message Service には、TIBCO Rendezvous アプリケーションにインクルードすることができる `tibrvjms.jar` ファイルが付属しています。この JAR ファイルには、`com.tibco.tibrv.TibrvJMSTransport` クラスの実装が含まれています。このクラスは `com.tibco.tibrv.TibrvNetTransport` クラスを拡張し、ユーザーの Pure Java Rendezvous プログラムが `rva` を介さずに、EMS サーバーと直接通信できるようにします。

アプリケーションには `tibrvjms.jar` および、`tibrvjweb.jar` または `tibrvj.jar` のいずれかを含める必要がありますが、`tibrvnative.jar` を含めることはできません。

`TibrvJMSTransport` クラスを使用するには、アプリケーションに `tibrvjms.jar` (EMS に付属) と `tibrvjweb.jar` または `tibrvj.jar` (TIBCO Rendezvous に付属) のいずれかが含まれている必要があります。アプリケーションには、`tibrvnative.jar` を含めることはできません。



`TibrvJMSTransport` を使用できるのは、Rendezvous アプリケーションのみです。このクラスは、EMS Java クライアントで使用するものではありません。

`TibrvJMSTransport` クラスを使用する Java Rendezvous アプリケーションを作成するには、TIBCO Rendezvous と EMS の両方を購入し、インストールおよび設定を済ませておく必要があります。

`TibrvJMSTransport` クラスは、Rendezvous の信頼通信のみを提供します。保証メッセージングなど、その他の種類の通信は、このトランスポートではサポートされていません。

このトランスポートを使用するアプリケーションは、メッセージのサブジェクトと同じトピック名を持つ EMS サーバー上のトピックにメッセージを送信できます。`TibrvJMSTransport` 経由で送信された Rendezvous メッセージを受信する EMS トピックは、`import` プロパティを指定する必要がありません。このトランスポートは、JMS キューにメッセージを送信するためには使用できません。

`TibrvNetTransport` および TIBCO Rendezvous Java プログラムでトランスポートを使用する方法についての詳細は、TIBCO Rendezvous のマニュアルを参照してください。[TibrvJMSTransport](#) のその他のメソッドについての詳細は、TIBCO Enterprise Message Service Java API リファレンスを参照してください。



## 第 16 章

# TIBCO SmartSockets との相互運用

この章では、TIBCO Enterprise Message Service と TIBCO SmartSockets の相互運用について説明します。

### トピック

---

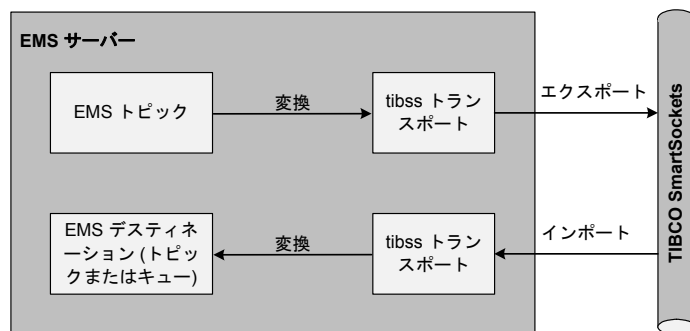
- 「概要」、414 ページ
- 「[SmartSockets のためのトランスポートの設定](#)」、415 ページ
- 「[トピック](#)」、421 ページ
- 「[キュー](#)」、422 ページ
- 「[インポートに関連する問題](#)」、424 ページ
- 「[エクスポートに関連する問題](#)」、425 ページ
- 「[メッセージの変換](#)」、426 ページ

## 概要

TIBCO Enterprise Message Service は TIBCO SmartSockets とメッセージを交換できます。

- 範囲
- EMS は、EMS の「トピック」を通して、外部システムとの間でメッセージをインポートおよびエクスポートできます。
  - EMS は、外部システムから EMS の「キュー」にメッセージをインポートできます（ただし、キューはエクスポートを行えません）。

図 24 EMS サーバー内の SmartSockets トランスポート



## メッセージの変換

EMS と SmartSockets は、それぞれメッセージとデータに独自の形式を使用しています。tibemspd でメッセージのインポート / エクスポートを行う場合、メッセージとデータを適切な形式に変換します。詳細は、426 ページの「[メッセージの変換](#)」を参照してください。

## 設定

tibemspd では、三つの設定ファイルに定義とパラメータを設定して、SmartSockets とメッセージを交換できるようにします。

- 有効化
- パラメータ `tibss_transports` (`tibemspd.conf` 設定ファイル) は、SmartSockets とのメッセージのやり取りをグローバルに有効化または無効化します。デフォルトの値は、`disabled` です。これらのトランスポートを使用するには、このパラメータを明示的に「`enabled`」に設定する必要があります。

パラメータ `tibss_config_dir` (設定ファイル `tibemsd.conf`) は、`tibemsd` 内で SmartSockets クライアントが必要とする SmartSockets ファイルの場所を指定します。

- |           |   |
|-----------|---|
| トランスポート   | トランスポート定義 ( <code>transports.conf</code> 設定ファイル) は、EMS と外部システム間の通信プロトコルを指定します。詳細は、415 ページの「 <a href="#">SmartSockets のためのトランスポートの設定</a> 」を参照してください。   |
| デスティネーション | <p>デスティネーション定義 (設定ファイル <code>topics.conf</code> および <code>queues.conf</code>) は、<code>import</code> および <code>export</code> プロパティを設定して、ひとつ以上のトランスポートを選択できます。</p> <ul style="list-style-type: none"> <li>• <code>import</code> を設定すると、<code>tibemsd</code> は、SmartSockets からこれらのトランスポートに送信されたメッセージをインポートし、EMS のデスティネーションにメッセージを配信します。</li> <li>• <code>export</code> を設定すると、<code>tibemsd</code> は、EMS のデスティネーションに届いたメッセージを取り出し、選択されているトランスポート経由で、SmartSockets へメッセージをエクスポートします。</li> </ul> <p>詳細は、421 ページの「<a href="#">トピック</a>」および 422 ページの「<a href="#">キュー</a>」を参照してください。</p> |

## サーバーの起動

`tibemsd` を実行する前に、SmartSockets RTserver を起動することをお勧めします。

## SmartSockets のためのトランスポートの設定

---

トランスポートは、TIBCO Enterprise Message Service と TIBCO SmartSockets との間でやり取りされるメッセージのフローを調整します。

`timemsd` は、他の SmartSockets クライアントと同じ方法で SmartSockets RTservers に接続します。トランスポート定義 (`transports.conf` ファイル) は、これらの接続の動作を設定します。これらのトランスポートを正しく設定する必要があります。

## トランスポート定義

`transports.conf` には、ゼロ個以上のトランスポート定義が含まれています。各定義は、角カッコ ([ ]) で囲まれたトランスポートの名前で始まります。それに続く行で、トランスポートのパラメータを設定します。

表 68 SmartSockets : トランスポートのパラメータ (1 / 4 シート)

パラメータ	説明
type	必須。SmartSockets トランスポートの場合、値は <code>tibss</code> です。
<b>SmartSockets パラメータ</b>	
これらのパラメータの構文とセマンティクスは、SmartSockets クライアントの対応するパラメータと同じです。詳細は、SmartSockets のマニュアルセットを参照してください。	
server_names	<p>SmartSockets RTservers への接続を指定します。複数の接続を指定する場合は、コンマで区切って指定します。</p> <p>リスト内の個々の接続は、<code>protocol:hostname:port</code> (ポート番号) の形式で指定します。この三つの部分のいずれも省略可能です。デフォルトのホスト名はローカルのホストコンピュータです。デフォルトのプロトコルとポート番号は、ハードウェアおよびオペレーティングシステムのプラットフォームによって異なります。Windows の場合のデフォルトは、プロトコルが <code>tcp</code>、ポート番号は <code>5101</code> です。</p> <p>複数のサーバーを列挙するとフォールトトレランスが指定され、<code>timemsd</code> は、列挙された順番にサーバーへの接続を試みます。</p> <p>このパラメータを指定しない場合、デフォルトで EMS サーバーは、ローカルホストコンピュータ (EMS サーバーと同じコンピュータ) 上の RTserver へ接続を試みます。この場合、デフォルトのプロトコルとポート番号が使用されます。</p>
username password	<code>timemsd</code> はこのふたつのパラメータを使って、各 SmartSockets サーバーに対して自身の認証を行います。
project	<p>SmartSockets はプロジェクトを使って、サブジェクトの名前空間の直交性を確保します。</p> <p>このパラメータを指定しなかった場合、デフォルトのプロジェクトは <code>rtworks</code> です。</p>
delivery_mode	<p>このトランスポート経由で SmartSockets サーバーへメッセージを配信するときのサービスレベル (通信品質) を指定します。</p> <p><code>best_effort</code>   <code>gmd_all</code>   <code>gmd_some</code>   <code>ordered</code></p> <p>このパラメータを指定しなかった場合、デフォルトは <code>best_effort</code> です。</p>



表 68 SmartSockets : トランスポートのパラメータ ( 2 / 4 シート)

パラメータ	説明
lb_mode	<p>SmartSockets サーバーは、複数のクライアント間にメッセージを配信して、メッセージの負荷を分散させます。このパラメータは、このトランスポート経由で SmartSockets サーバーへエクスポートされるメッセージの負荷分散処理を指定します。</p> <p><code>none</code>   <code>round_robin</code>   <code>weighted</code>   <code>sorted</code></p> <p>このパラメータを指定しなかった場合、デフォルトは <code>none</code> です。</p>
override_lb_mode	<p><code>enable</code> を設定すると、負荷分散が他のクライアントに振り分けられている場合でも、RTserver はこのクライアント接続ですべてのメッセージを配信します。たとえば、多数の注文処理クライアントに注文メッセージの負荷が分散されているような場合でも、メッセージログ機能は、サブセットではなく、すべての注文メッセージを必要とします。</p> <p><code>disable</code> を設定すると、このクライアント (EMS サーバー) で負荷分散が行われている (たとえば、他の EMS サーバーと負荷を共有している) ことを RTserver に伝えます</p> <p>このパラメータを指定しなかった場合、デフォルトは <code>enable</code> です。</p>
gmd_file_delete	<p>SmartSockets クライアントは、保証メッセージ配信 (GMD) 用のデータをストアファイルに格納しています。</p> <p><code>disable</code> を指定すると、tibemspd は既存の GMD ストアファイルを開きます。</p> <p><code>enable</code> を設定すると、tibemspd は、このトランスポートの作成時に GMD ストアファイルを削除し、新しいストアファイルを作成します。</p> <p>このパラメータを指定しなかった場合、デフォルトは <code>disable</code> です。</p>
import_ss_headers	<p>EMS プロパティへの SmartSockets メッセージヘッダのインポートを制御します。</p> <p>パラメータの値には、<code>none</code>、<code>type_num</code>、または <code>all</code> のいずれかを指定できます。詳細は、427 ページの「<a href="#">SmartSockets メッセージのプロパティ</a>」を参照してください。</p> <p>このパラメータを指定しなかった場合、デフォルト値は <code>none</code> です。</p>

表 68 SmartSockets : トランSPORTのパラメータ ( 3 / 4 シート)

パラメータ	説明
preserve_gmd	<p>このパラメータは、SmartSockets へエクスポートした GMD メッセージを SmartSockets が配信できない場合の EMS サーバーの動作を指定します。SmartSockets から未配信のメッセージが返されると、EMS サーバーは、そのメッセージを EMS の未配信メッセージキューに保存するか、破棄します。</p> <ul style="list-style-type: none"><li>• <b>always</b> を指定すると、EMS サーバーは、すべての未配信 GMD メッセージを EMS の未配信メッセージキューに保存します。</li><li>• <b>receivers</b> を指定すると、EMS サーバーは、ひとつ以上の GMD レシーバが存在するにも関わらず SmartSockets が配信できなかった未配信 GMD メッセージのみを保存します。したがって、GMD レシーバが存在しなかったために SmartSockets がメッセージを配信できなかった場合、EMS サーバーはその未配信 GMD メッセージを保存しません。</li><li>• <b>never</b> を指定すると、EMS サーバーは、すべての SmartSockets の未配信メッセージを破棄します。</li></ul> <p>このパラメータを指定しなかった場合、デフォルト値は <b>never</b> です。</p> <p>このパラメータが適用されるのは、トランSPORTの <b>delivery_mode</b> パラメータが <b>gmd_all</b> または <b>gmd_some</b> のいずれかに指定されている場合です。</p> <p>EMS サーバーが GMD メッセージを保存する際は、返された SmartSockets メッセージを以下のルールに従って EMS メッセージに変換します。</p> <ul style="list-style-type: none"><li>• メッセージのインポートの一般ルールに従う (426 ページの「<a href="#">メッセージの変換</a>」を参照)。</li><li>• <b>import_ss_headers</b> パラメータの値を破棄し、代わりにすべての SmartSockets ヘッダを (import_ss_headers の値に <b>all</b> が指定されていたかのように) インポートする。ヘッダについては、427 ページの「<a href="#">SmartSockets メッセージのプロパティ</a>」のリストを参照してください。</li><li>• <b>JMS_TIBCO_SS_EXPIRATION</b> の値に現在の時刻 (SmartSockets サーバーが未配信メッセージを EMS サーバーに返した時刻) を設定する。(GMD メッセージの期限は切れないため、時刻を設定しない場合は、このヘッダは未使用のままになります。)</li></ul>

表 68 SmartSockets : トランスポートのパラメータ (4 / 4 シート)

パラメータ	説明
<b>EMS パラメータ</b>	
topic_import_dm queue_import_dm	<p>EMS 送信クライアントは、各メッセージで <code>JMSDeliveryMode</code> ヘッダフィールドを設定できます。SmartSockets クライアントはこのヘッダを設定できません。代わりに、これらのふたつのパラメータを使用して、<code>tibemsd</code> がこのトランスポート上にインポートする、すべてのトピックメッセージとキューメッセージの配信モードを決定します。</p> <p><code>TIBEMS_PERSISTENT</code>   <code>TIBEMS_NON_PERSISTENT</code>   <code>TIBEMS_RELIABLE</code></p> <p>このパラメータを指定しなかった場合、デフォルトは <code>TIBEMS_NON_PERSISTENT</code> です。</p>
export_headers	<p><code>true</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> ヘッダフィールドを設定します。</p> <p><code>false</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> ヘッダフィールドを設定しません。</p> <p>このパラメータを指定しなかった場合、デフォルト値は <code>true</code> です。</p>
export_properties	<p><code>true</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> プロパティを設定します。</p> <p><code>false</code> を指定した場合、<code>tibemsd</code> は、エクスポートされるメッセージに <code>JMS</code> プロパティを設定しません。</p> <p>このパラメータを指定しなかった場合、デフォルト値は <code>true</code> です。</p>

**例**

以下の例は、`transports.conf` ファイルの一部で、トランスポート定義の構文を示しています。

```
[SS01]
type = tibss
server_names = rtHost1
username = emsServer6
password = myPasswd
project = sales_order_entry

[SS02]
type = tibss
server_names = tcp:rtHost2A:5555, ssl:rtHost2B:5571
username = emsServer6
password = myPasswd
project = mfg_process_control
override_lb_mode = enable
delivery_mode = gmd_some
```

## デスティネーション名：構文とセマンティクス

スラッシュおよびドットの区切り文字

EMS のデスティネーション名と SmartSockets のサブジェクトとの対応関係は 1 対 1 の単純なもので、双方向の関係です。

EMS のデスティネーション名はドット (.) で区切られたトークンで構成されます。一方、SmartSockets のサブジェクトは、UNIX ディレクトリのパス名のようにスラッシュ (/) で区切られたトークンで構成されます。

たとえば、EMS の `foo.bar.baz` というデスティネーション名は、SmartSockets のサブジェクト名では `/foo/bar/baz` と表されます (SmartSockets のサブジェクト名はスラッシュから始まる必要がありますが、EMS のデスティネーション名はドットで始まる必要はありません。先頭にドットがあると、その前の要素が空であることを示します)。

スラッシュとドットは、EMS と SmartSockets で相補的な役割を持ちます。EMS ではスラッシュは通常の文字ですが、ドットは区切り文字として使われます。SmartSockets ではスラッシュは区切り文字ですが、ドットは通常の文字として使われます。EMS と SmartSockets 間で名前を変換するには、これらの文字をそれぞれ置き換えます。たとえば、EMS の `foo/bar.baz` というデスティネーション名は、SmartSockets のサブジェクト名では `/foo.bar/baz` と表されます。ただし、混乱を避けるため、スラッシュとドットは通常の文字として使用しないことをお勧めします。

アスタリスク (\*) のワイルドカード

EMS も SmartSockets も共にアスタリスク (\*) をワイルドカードとして認識しますが、それぞれのセマンティクスは異なります。これに関しては 1 対 1 の対応関係にはなりません。

EMS では、アスタリスクは名前のトークン全体に一致させることができますが、トークンの一部に一致させることはできません。SmartSockets では、アスタリスクはトークンの一部に一致させることができます。たとえば、`/foo/b*/baz` は、`/foo/bar/baz` および `/foo/box/baz` のいずれにも一致します。

SmartSockets のワイルドカードに慣れているが、EMS のワイルドカードについてはよく知らない場合は、73 ページの「[ワイルドカード](#)」を参照してください。

大なり記号 (>) のワイルドカード

EMS では、大なり記号 (>) は後続の任意の個数のトークンに一致するワイルドカードです。SmartSockets では、3 個のドット (...) で同じ意味を表します。

## トピック

トピックはメッセージのエクスポートとインポートが可能です。したがって、`topics.conf` 設定ファイルのトピックの定義で、`import` と `export` のプロパティを設定して、外部トランスポートをひとつ以上選択できます。

- import**
  - `import` を設定すると、`tibemsd` は、`SmartSockets` からこれらのトランスポートに送信されたメッセージをインポートし、EMS のデスティネーションにメッセージを配信します。
- export**
  - `export` を設定すると、`tibemsd` は、EMS のデスティネーションに届いたメッセージを取り出し、選択されているトランスポート経由で、`SmartSockets` へメッセージをエクスポートします。



EMS サーバーは、同じトピックにインポートされたメッセージを再エクスポートしません。

(`topics.conf` の構文とセマンティクスについての一般的な情報は、[247 ページの「topics.conf」](#)を参照してください。管理ツールの `addprop topic` コマンドを使ってもトピックを設定できます。)

### 例

たとえば、トピック `myTopics.news` を設定して三つのトランスポート経由でメッセージをインポートおよびエクスポートするには、`tibemsd` コマンドを使って次のように入力します。

```
addprop topic myTopics.news import="SS01,SS02"
addprop topic myTopics.news export="SS01,SS02,SS03"
```

サブジェクト `/myTopics/news` を含む `SmartSockets` メッセージが、`SS01` および `SS02` のトランスポート経由で、`tibemsd` に届きます。EMS クライアントは、`myTopics.news` をサブスクライブすることで、これらのメッセージを受け取ることができます。

`myTopics.news` に送信された EMS メッセージは、`SS01`、`SS02`、および `SS03` の三つのトランスポート経由で `SmartSockets` にエクスポートされます。対応する `RTservers` の `SmartSockets` クライアントは、`/myTopics/news` をサブスクライブすることで、これらのメッセージを受け取ることができます。

### サブスクライバが存在する場合のみのインポート

接続中のトランスポート経由でのインポートがトピックに指定されると、`tibemsd` は、そのトピックにサブスクライバが登録されている場合のみメッセージをインポートします。

## ワイルドカード

`import` および `export` プロパティに使用するワイルドカードは、EMS の構文とセマンティクス（SmartSockets の構文とセマンティクスではなく）に従います（420 ページの「[デスティネーション名：構文とセマンティクス](#)」を参照）。

## キュー

---

キューは、メッセージをインポートできますが、エクスポートできません。

## 設定

`queues.conf` 設定ファイルのキュー定義で、`import` プロパティを設定して、外部トランスポートをひとつ以上選択できます。

- `import` を設定すると、`tibemspd` は、SmartSockets からこれらのトランスポートに送信されたメッセージをインポートし、EMS のデスティネーションにメッセージを配信します。

（`queues.conf` の構文とセマンティクスについての一般的な情報は、[240 ページの「`queues.conf`」](#)を参照してください。管理ツールの `addprop queue` コマンドを使ってもキューを設定できます。）

例 たとえば、キュー `myTopics.news` を設定して `SS01` と `SS02` トランスポート上のメッセージをインポートするには、`tibemspd` コマンドを使って次のように入力します。

```
addprop queue myQueue.in import="SS01,SS02"
```

`/myQueue/in` サブジェクトの SmartSockets メッセージが、`SS01` および `SS02` のトランスポート経由で、`tibemspd` に届きます。EMS クライアントは、`myQueue.in` をサブスクライブすることで、これらのメッセージを受け取ることができます。

## インポート：開始および停止

接続されたトランスポート経由でのインポートがキューに指定されると、`tibemspd` は、キューのレシーバが存在しない場合でも、直ちにキューへのメッセージのインポートを開始します。

管理者によって静的に設定されたキューの場合は、管理者がキューを明示的に削除するまで、`tibemspd` はインポート処理を継続します。

## ワイルドカード

import プロパティに使用するワイルドカードは、EMS の構文とセマンティクス (SmartSockets の構文とセマンティクスとは異なる) に従います (420 ページの「[デスティネーション名：構文とセマンティクス](#)」を参照)。

EMS クライアントは、ワイルドカードキューをサブスクライブすることはできません。ただし、EMS サーバーでワイルドカードキューを定義して、プロパティを継承させることができます。たとえば、`foo.*` という名前の静的キューを設定して、そのプロパティを設定することで、`foo.bar` および `foo.baz` という名前の子のキューの両方にそれらのプロパティを継承させることができます。

`foo.*` をインポートするキューを定義すると、`tibemsd` は、このワイルドカードのパターンに一致する、`SmartSockets` 内のすべてのメッセージのインポートを開始します。メッセージが届くと、`tibemsd` は動的に子のキュー (`foo.bar` および `foo.baz`) を作成し、メッセージをこれらのキューに配信します。そのメッセージを取得するサブスクライバがない場合でも、`tibemsd` は動的に作成されたこれらの子のキューにメッセージを配信するので注意が必要です

## インポートに関連する問題

このセクションでは、SmartSockets から EMS への、トピックまたはキューのメッセージのインポートに関連する問題を取り上げます。

### インポートしたデスティネーション名は固有の名前である必要がある



トピックとキューが同じ名前を使用している場合、そのどちらか一方のみで `import` プロパティを指定できます。たとえば、トピック `foo.bar` とキュー `foo.bar` を定義すると、どちらか一方でのみ `import` プロパティを指定できます。

### JMSReplyTo

`tibemspd` がメッセージをインポートし、SmartSockets メッセージに変換すると、EMS メッセージの `JMSReplyTo` フィールドが SmartSockets の `reply_to` ヘッダに設定され、EMS クライアントがメッセージに応答できるようになります。

通常、この値には SmartSockets サブジェクトが示されます。メッセージを SmartSockets にエクスポートするのに使用される、対応する名前のトピックを生成するためには、`tibemspd` を明示的に設定する必要があります。

### 保証配信



SmartSockets から EMS へのエンドツーエンドの完全な配信を保証するには、以下の三つの条件をすべて満たすことが要求されます。

- SmartSockets センダーは保証メッセージ配信（GMD）のメッセージを送信する必要があります。
- トランSPORT定義で、`topic_import_dm` または `queue_import_dm`（状況に応じて選択）を `TIBEMS_PERSISTENT` に設定する必要があります。
- EMS トピックまたはキューに持続サブスクライバが存在する必要があります。

エクスポートの保証については、425 ページの「[保証配信](#)」を参照してください。



## エクスポートに関連する問題

このセクションでは、EMS から SmartSockets へのメッセージのエクスポートに関連する問題を取り上げます。

### JMSReplyTo

**トピック** EMS メッセージの JMSReplyTo フィールドにトピックが含まれているとします。このようなメッセージを SmartSockets へエクスポートする場合、tibemsd がその応答トピックへの応答を SmartSockets からインポートするよう、明示的に設定する必要があります。

**テンポラリトピック** EMS メッセージの JMSReplyTo フィールドにテンポラリトピックが含まれているとします。tibemsd がこのようなメッセージを SmartSockets にエクスポートする場合、そのテンポラリトピックへの応答を「自動的に」SmartSockets からインポートするようになっています。したがって、明示的に設定する必要はありません。

### ワイルドカードサブスクリプション

**アスタリスク (\*)  
ワイルドカード** EMS も SmartSockets も共にアスタリスク (\*) をワイルドカードとして認識しますが、それぞれのセマンティクスは異なります。EMS はアスタリスクを全体的な要素として認識し、要素全体に一致させます。一方、SmartSockets はアスタリスクを部分的な要素として認識し、要素内の一部の文字列に一致させます。

SmartSockets クライアントが `foo.bar*` をサブスクライブし、tibemsd がスーパーセット `foo.*` をエクスポートするように設定すると、RTserver は対象のセットを絞り、サブスクライブに一致するメッセージのみを配信します。EMS と SmartSockets のワイルドカードの違いについての詳細は、420 ページの「[デステーション名：構文とセマンティクス](#)」を参照してください。

### 保証配信



EMS から SmartSockets へのエンドツーエンドの完全な配信を保証するには、以下のふたつの条件をすべて満たすことが要求されます。

- EMS センダーは永続的メッセージを送信する必要がある。
- トランスポートの定義で、`delivery_mode` を `gmd_some` または `gmd_all` のいずれかに（適切なほうに）指定する必要がある。

EMS の未配信メッセージキューへの未配信 GMD メッセージの保存については、418 ページの「[preserve\\_gmd](#)」を参照してください。インポートの保証については、424 ページの「[保証配信](#)」を参照してください。

## メッセージの変換

### JMS ヘッダフィールド

EMS は、10 種類の定義済みの JMS ヘッダフィールドをサポートします (17 ページの「[JMS メッセージのヘッダフィールド](#)」を参照)。

#### ふたつの特別な ケース

以下のふたつのヘッダフィールドは、特殊なケースです。

- JMS ヘッダ `JMSDestination` は SmartSockets の `dest` に対応しています。
- JMS ヘッダ `JMSReplyTo` は SmartSockets の `reply_to` に対応しています。

#### インポート

SmartSockets メッセージを EMS メッセージにインポートする場合は、上記の特殊なケースを除いて、`tibemspd` は、JMS ヘッダフィールドを設定しません。

#### エクスポート

EMS メッセージを SmartSockets メッセージにエクスポートする場合、`tibemspd` は、(上記の特殊なケースを除き)すべての JMS ヘッダフィールドを SmartSockets メッセージ内のひとつのサブメッセージにまとめます。JMSHeaders フィールドにそのサブメッセージが含まれます。サブメッセージの各フィールドにより、JMS ヘッダフィールドの名前がその値にそれぞれマッピングされます。

`tibemspd` は `null` であるか、値が指定されていない JMS ヘッダフィールドは無視します。つまり、エクスポートされるメッセージから除外します。

トランスポートプロパティを `export_headers = false` と設定することで、すべてのエクスポートされるメッセージについて、ヘッダサブメッセージを完全に出力しないように `tibemspd` に指示することができます。

### JMS プロパティフィールド

#### インポート

SmartSockets メッセージを EMS メッセージにインポートした場合、`tibemspd` は以下の JMS プロパティを設定します。

- [JMS\\_TIBCO\\_IMPORTED](#) は `true` に設定され、メッセージが EMS クライアントから送信されたものでないことを示します。
- [JMS\\_TIBCO\\_MSG\\_EXT](#) は `true` に設定され、メッセージにサブメッセージフィールドまたは配列フィールドが含まれている「可能性」があることを示します。
- [JMS\\_TIBCO\\_SS\\_SENDER](#) は、SmartSockets `sender` ヘッダフィールド (SmartSockets 構文) の値に設定されます。

さらに、`tibemspd` は、SmartSockets メッセージプロパティを EMS プロパティにマッピングします。詳細は、427 ページの「[SmartSockets メッセージのプロパティ](#)」を参照してください。

**エクスポート** SmartSockets メッセージへ EMS メッセージをエクスポートする場合、tibemsd は、すべての JMS プロパティフィールドを SmartSockets メッセージ内のひとつのサブメッセージにまとめます。JMSProperties フィールドにそのサブメッセージが含まれます。サブメッセージの各フィールドにより、JMS プロパティフィールドの名前がその値にそれぞれマッピングされます。

tibemsd は、null であるか、値が指定されていない JMS プロパティフィールドは無視します。つまり、エクスポートされるメッセージから除外します。

トランスポートプロパティを `export_properties = false` と設定することで、エクスポートされるメッセージで、プロパティサブメッセージを完全に出力しないように tibemsd に指示することができます。

SmartSockets メッセージのプロパティ

リリース 4.1.0 以降では、インポート時に tibemsd は SmartSockets メッセージのヘッダを EMS メッセージのプロパティにマッピングします。表 69 に対応関係を示します。最初のカラムは EMS のプロパティを、2 番目のカラムは対応するヘッダを取得する SmartSockets のメソッドを示します。

**インポート** トランスポートパラメータ `import_ss_headers` はインポートの動作を制御します。表 69 の 3 番目のカラムは、その行のメッセージプロパティを tibemsd がインポートするときに使用するパラメータの値を示します。417 ページの「`import_ss_headers`」を参照してください。

**エクスポート** EMS クライアントプログラムは、インポートされたメッセージを SmartSockets へ再度エクスポートする際に、メッセージ内のこれらの値を変更することがあります（ただし、ネイティブの EMS メッセージをエクスポートした場合、これらのプロパティは SmartSockets に継承されません）。

これらのプロパティのエクスポートは、トランスポートパラメータ（419 ページの「`export_properties`」）の値に依存します。

EMS メッセージを SmartSockets メッセージにエクスポートした場合、逆に tibemsd はこれらのプロパティをマッピングします。ほとんどの場合、マッピングは対称的であるため、エクスポートによるマッピングで、SmartSockets の元のヘッダに戻されます。ただし、マッピングが対称的にならない例外が三つあります（`JMS_TIBCO_SS_SENDER`、`JMS_TIBCO_SS_MESSAGE_ID`、および `JMS_TIBCO_SS_SEQ_NUM`）。これらをエクスポートすると、SmartSockets メッセージ内の JMSProperties フィールドのサブフィールドにマッピングされます。表 69 の四番目のカラムに、この非対称なマッピングが示されています。

表 69 SmartSockets : 対応するメッセージプロパティ（インポートとエクスポート）（1 / 2 シート）

EMS のプロパティ	SmartSockets のメソッド	インポート	エクスポート（非対称）
JMS_TIBCO_SS_SENDER	TipcMsgGetSender	none type_num all	非対称。

表 69 SmartSockets：対応するメッセージプロパティ（インポートとエクスポート）（2 / 2 シート）

EMS のプロパティ	SmartSockets のメソッド	インポート	エクスポート（非対称）
JMS_TIBCO_SS_TYPE_NUM	TipcMsgGetType	type_num all	
JMS_TIBCO_SS_DELIVERY_MODE	TipcMsgGetDeliveryMode	all	
JMS_TIBCO_SS_LB_MODE	TipcMsgGetLbMode	all	
JMS_TIBCO_SS_EXPIRATION	TipcMsgGetExpiration	all	
JMS_TIBCO_SS_PRIORITY	TipcMsgGetPriority	all	
JMS_TIBCO_SS_SENDER_TIMESTAMP	TipcMsgGetSenderTimestamp	all	
JMS_TIBCO_SS_CORRELATION_ID	TipcMsgGetCorrelationId	all	
JMS_TIBCO_SS_USER_PROP	TipcMsgGetUserProp	all	
JMS_TIBCO_SS_MESSAGE_ID	TipcMsgGetMessageId	all	非対称。
JMS_TIBCO_SS_SEQ_NUM	TipcMsgGetSeqNum	all	非対称。

メッセージ本文

tibemsd は、どのようなメッセージ本文タイプを持った JMS メッセージでも TIBCO SmartSockets にエクスポートできます。反対に、tibemsd は、どのようなメッセージタイプのメッセージでも TIBCO SmartSockets からインポートできます。

JMS の本文タイプについての詳細は、21 ページの「[JMS メッセージ本文](#)」を参照してください。

メッセージの構造についての詳細は、17 ページの「[JMS メッセージの構造](#)」を参照してください。

- インポート
- SmartSockets メッセージをインポートすると、tibemsd は、以下の EMS メッセージ本文タイプのいずれかにメッセージを変換します。
- SmartSockets メッセージに「名前のない」フィールドのみが含まれている場合、JMSStreamMessage に変換されます。ストリームには、SmartSockets メッセージ内の表示と同じ順序で名前のないフィールドの値が含まれます。
  - SmartSockets メッセージに名前の付いたフィールドがひとつ以上含まれている場合は、JMSMapMessage に変換されます。マップメッセージには名前の付いたフィールドが含まれます。フィールドの順序は確定されません。

**エクスポート** EMS メッセージをエクスポートすると、tibemspd はそのメッセージを以下の構造を持つ 6 種類の SmartSockets メッセージタイプ（表 70 を参照）のひとつに変換します。

- **JMSHeaders** という名前のフィールドが最初のフィールドになります（トランスポートパラメータ `export_headers` を `false` に設定すると、このフィールドは省略されます）。このフィールドにはサブメッセージが含まれます（426 ページの「**JMS ヘッダフィールド**」を参照）。
- **JMSProperties** という名前のフィールドが 2 番目のフィールドになります（トランスポートパラメータ `export_properties` を `false` に設定すると、このフィールドは省略されます）。このフィールドにはサブメッセージが含まれます（426 ページの「**JMS プロパティフィールド**」を参照）。
- データフィールドは、JMS ヘッダとプロパティに従います（これらが設定されている場合）フィールドの名前とタイプについての詳細は、表 70 の 3 番目のカラムを参照してください。

表 70 SmartSockets: 対応するメッセージタイプ（エクスポート）

JMS メッセージタイプ	SmartSockets のメッセージタイプ	データフィールド
JMSBytesMessage	T_MT_JMS_BYTES	名前のない T_MSG_FT_BINARY タイプの フィールド
JMSMapMessage	T_MT_JMS_MAP	名前の付いたフィールド。 順序不同。
JMSObjectMessage	T_MT_JMS_OBJECT	名前のない T_MSG_FT_BINARY タイプの フィールド
JMSStreamMessage	T_MT_JMS_STREAM	名前のないフィールド。順 番に表示。
JMSTextMessage	T_MT_JMS_TEXT	名前のない T_MSG_FT_STR タイプのフィールド
その他の JMS メッセージタイプ	T_MT_INFO	データフィールドなし

データ型

表 71 は、EMS のデータ型と SmartSockets のデータ型の対応関係を示します。対応する EMS のデータ型がない SmartSockets の一部のデータ型を除き（これらのタイプは、表 71 の中央のカラムに片方向としてマークされています）、マッピングは双方向です。

表 71 SmartSockets: データ型の対応関係（1 / 2 シート）

EMS	マッピング方向	SmartSockets
Boolean		T_MSG_FT_BOOL
Byte		T_MSG_FT_BYTE
Character		T_MSG_FT_CHAR
Short		T_MSG_FT_INT2
Integer		T_MSG_FT_INT4
Long		T_MSG_FT_INT8
Float		T_MSG_FT_REAL4
Double		T_MSG_FT_REAL8
Double	<—	T_MSG_FT_TIMESTAMP
String		T_MSG_FT_STR
String	<—	T_MSG_FT_XML
String	<—	T_MSG_FT_UTF8
Byte Array		T_MSG_FT_BINARY
Short Array	<—	T_MSG_FT_BOOL_ARRAY
Short Array		T_MSG_FT_INT2_ARRAY
Integer Array		T_MSG_FT_INT4_ARRAY
Long Array		T_MSG_FT_INT8_ARRAY
Float Array		T_MSG_FT_REAL4_ARRAY

表 71 SmartSockets: データ型の対応関係 (2 / 2 シート)

EMS	マッピング方向	SmartSockets
Double Array		T_MSG_FT_REAL8_ARRAY
Double Array	←	T_MSG_FT_TIMESTAMP_ARRAY
Stream Message		T_MSG_FT_MSG
Map Message		(428 ページの「インポート」を参照)。

デスティネーション名

メッセージをインポートまたはエクスポートすると、tibemsd は、自動的にデスティネーション名を変換します (420 ページの「スラッシュおよびドットの区切り文字」を参照)。

インポートする場合、SmartSockets の subject と reply\_to フィールド内の名前を変換します。エクスポートする場合、EMS の JMSDestination と JMSReplyTo フィールド内の名前を変換します。





## 第 17 章

## サーバーアクティビティの監視

システム管理者は TIBCO Enterprise Message Service サーバーを監視し、管理する必要があります。サーバーが提供するログ記録、監視、統計情報などの機能を使用して、システム管理者は効率的にシステムのアクティビティを表示し、システムのパフォーマンスを追跡することができます。

トピック

---

- [「ログファイルとトレース」、434 ページ](#)
- [「メッセージのトレース」、440 ページ](#)
- [「サーバーイベントの監視」、442 ページ](#)
- [「サーバーの統計情報の操作」、448 ページ](#)

## ログファイルとトレース

TIBCO Enterprise Message Service サーバーを設定して、ログにさまざまな情報を書き出すことができます。サーバーのパラメータとコマンドを設定して、ログファイルの場所を指定したり、ログファイルに記録する情報を定義したりすることができます。ログはファイルまたはシステムコンソールのいずれか、または両方に出力することができます。

### ログファイルの設定

`tibemsd.conf` ファイル内の `logfile` 設定パラメータを設定して、ログファイルの場所と名前を指定します。

ログファイルをバックアップし、最大サイズに達したら空になるように指定することもできます。これにより、ログファイルを使い回し、ログファイルのサイズが無限に大きくなるのを防ぐことができます。`logfile_max_size` 設定パラメータを設定して、ログファイルの最大サイズを指定できます。パラメータにゼロを指定すると、ファイルサイズの制限は設定されません。単位には、KB、MB、または GB を指定します。

ログファイルが最大サイズに達すると、ファイルは現在のログファイルと同じ名前前で、シーケンス番号が付いたバックアップファイルにコピーされます。サーバーはディレクトリを検索し、最初に利用可能なコピーのシーケンス番号を決定します。たとえば、ログファイル名が `tibems.log` の場合、最初のコピーは `tibems.log.1`、2 番目のコピーは `tibems.log.2` というようなシーケンス番号になります。必要に応じて、ログディレクトリからファイルを移動することもできます。この場合、次のログファイルのシーケンス番号は、最初に利用可能なログファイルディレクトリ内のバックアップコピーのシーケンス番号に基づいて決定されます。



ログファイルを削除または移動する場合は、ログファイル内のすべてのログファイルを削除または移動することをお勧めします。このようにすると、サーバーはログファイルのシーケンス番号をまた 1 から始めることができます。

また、`tibemsadmin` の `rotatelog` コマンドを使用すると、動的にログファイルを強制的にバックアップしてから切り替えることができます。`rotatelog` コマンドについての詳細は、122 ページの「[コマンドのリスト](#)」を参照してください。

ログファイルに関連する他の設定パラメータについては、210 ページの「[ログファイルパラメータのトレース](#)」を参照してください。

## サーバー上でのトレース

TIBCO Enterprise Message Service サーバーを設定すると、トレースメッセージを生成できます。これらのメッセージに、さまざまな領域の機能（例：アクセス制御、管理、ルーティング）で実行するアクションの説明を記述できます。また、これらのメッセージを使って、サーバー上で、またはサーバーによって実行されるアクティビティに関する情報や、障害の発生および不正なアクションに関する警告を得ることもできます。

トレースメッセージは、ログファイルまたはシステムコンソールのいずれか、または両方に出力することができます。トレースの設定には、以下の方法があります。

- `tibemspd.conf` ファイルの `log_trace` や `console_trace` パラメータを設定する（137 ページの「表 16」を参照）。
- サーバーの起動時に、`-trace` オプションを指定する。
- サーバーの実行時に `set server` コマンドを使用する。

`log_trace` および `console_trace` パラメータを使用して、ログファイルおよびシステムコンソールに出力するメッセージのタイプを設定できます。



トレースメッセージをログファイルに出力する場合は、`logfile` 設定パラメータも設定する必要があります。`log_trace` をだけを指定して、`logfile` 設定パラメータに有効なファイルを設定しなかった場合、設定したトレースオプションは保存されますが、サーバーが有効なログファイルと共に起動されるまで、それらの設定は使用されません。

ログファイルまたはシステムコンソールでのトレースを設定する場合、生成可能なトレースメッセージのタイプを設定するためのさまざまなオプションがあります。436 ページの「表 72」に利用可能なトレースオプションを示します。

表 72 サーバーのトレースオプション

トレースオプション	説明
DEFAULT	トレースオプションにデフォルトセットを設定します。デフォルトセットには次の値が含まれます。 <ul style="list-style-type: none"><li>• INFO</li><li>• WARNING</li><li>• ACL</li><li>• LIMITS</li><li>• ROUTE</li><li>• ADMIN</li><li>• RVADV</li><li>• CONNECT_ERROR</li><li>• CONFIG</li><li>• MSG</li></ul>
ACL	許可されないアクションを実行しようとするメッセージを出力します。たとえば、パブリッシュするアクセス権限がユーザーに与えられていないセキュアトピックにメッセージをパブリッシュしようとするような場合です。
ADMIN	管理機能が実行されるときにメッセージを出力します。
AUTH	外部 LDAP システムを使用してサーバーがユーザーを認証するとメッセージを出力します。
CONFIG	EMS サーバーの起動時に設定ファイルおよびそのコンテンツに関する情報を出力します。
CONNECT	ユーザーがサーバーへの接続を試みるとメッセージを出力します。
CONNECT_ERROR	接続にエラーが発生するとメッセージを出力します。
DBSTORE	データベースストアが作成されると、一般的なデータベースストア情報およびエラーと共にメッセージを出力します。

表 72 サーバーのトレースオプション

トレースオプション	説明
DEST	動的なデスティネーションが作成されるとメッセージを出力します。
FLOW	サーバーがデスティネーションのフロー制御を実行または停止するとメッセージを出力します。
INFO	設定ファイルの作成、持続データベースファイルのオープン、およびメッセージのパーズなど、さまざまな内部管理機能をサーバーが実行する際に、メッセージを出力します。また、メッセージ ID によるメッセージ追跡を有効または無効にする場合もメッセージを出力します。
JAAS	拡張可能なセキュリティモジュールに関連するメッセージを出力します。  ユーザー名およびパスワードが認証用の <code>LoginModule</code> に渡されるとき、およびユーザーおよびアクションが認証用のアクセス権限モジュールに渡されるときに、メッセージが出力されます。
JVM	JVM 設定の起動情報、および <code>System.out</code> を使用する JVM で実行されているカスタムモジュールからの出力を表示します。
JVMERR	<code>System.err</code> を使用する JVM で実行されるカスタムモジュールからの出力を表示します。
LDAP_DEBUG	LDAP が認可に使用されるか、グループ情報の収集に使用されるとメッセージを出力します。
LIMITS	デスティネーションの最大サイズなど、制限値を超えるとメッセージが出力されます。
LOAD	動的に読み込まれたライブラリのパスを出力します。たとえば、 <code>tibemsd</code> は <code>Zlib</code> 、 <code>SmartSockets</code> 、および <code>SSL</code> ライブラリを読み込みます。
MEMORY	サーバーのメモリ不足の状況により予約メモリがトリガーされると、サーバートレース情報を出力します。

表 72 サーバーのトレースオプション

トレースオプション	説明
MSG	メッセージのトレースメッセージの出力を指定します。デスティネーションまたは個々のメッセージについて、メッセージのトレースを有効または無効にします。メッセージまたはデスティネーションに対してメッセージのトレースが有効になっていない場合、ログファイルまたはシステムコンソールでのトレースにこのオプションを指定しても、トレースメッセージは出力されません。メッセージトレースについての詳細は、440 ページの「 <a href="#">メッセージのトレース</a> 」を参照してください。
MULTICAST	メッセージコンシューマがマルチキャスト対応のトピックをサブスクライブするか、サブスクライブを試みると、一般的なマルチキャスト情報およびエラーと共にメッセージを出力します。
PRODCONS	クライアントがプロデューサまたはコンシューマを作成または閉じるとメッセージを出力します。
ROUTE	ルートが作成されたとき、またはルート接続が確立されたときにメッセージを出力します。
ROUTE_DEBUG	ルートのステータスおよびルートに関するエラーメッセージを出力します。
RVADV	TIBCO Rendezvous のアドバイザリメッセージを受信するとそれを出力します。
SS	SmartSockets ブリッジに関連するトレースメッセージを出力します。
SSL	証明書の内容など、SSL プロセスの詳細なメッセージを出力します。
SSL_DEBUG	SSL 接続の確立をトレースするメッセージを出力します。
TX	クライアントがトランザクションを実行するとメッセージを出力します。
WARNING	何らかの障害が発生するとメッセージを出力します。通常は、ユーザーが不正なアクションを実行しようすることが原因になります。たとえば、ユーザーがワイルドカードのデスティネーション名にパブリッシュしようとする、メッセージが出力されます。

トレースを指定する場合、各トレースオプションをコンマで区切ります。トレースオプションは、次の三つの形式で指定できます。

- 符号なし: プリフィクス文字が指定されていないトレースオプション。既存のすべてのトレースオプションを置換する。
- `+:+` で始まるトレースオプション。指定したオプションを現在のトレースオプションセットに追加する。
- `-:-` で始まるトレースオプション。指定したオプションを、現在のトレースオプションセットから削除する。

### 例:

次のコード例では、アクセス制御違反に関するメッセージのみを表示するようにトレースログを設定します。

```
log_trace=ACL
```

次のコード例では、SSL メッセージに加えて、すべてのデフォルトのトレースメッセージを表示するように設定します。ただし、ADMIN メッセージを除きます。

```
log_trace=DEFAULT, -ADMIN, +SSL
```

次のコード例では、TIBCO Rendezvous アドバイザリメッセージが到着したときに、トレースメッセージをコンソールに送信します。

```
console_trace=RVADV
```

## メッセージのトレース

他のサーバーのアクティビティをトレースするだけでなく、メッセージの処理時に、その状態をトレースすることもできます。メッセージのトレース情報は、トレースが実行されるデスティネーションまたはメッセージに対してのみ生成されます。デスティネーションの場合、トレースメッセージの生成を有効にするには、**trace** プロパティを指定します。個々のメッセージの場合は、**JMS\_TIBCO\_MSG\_TRACE** プロパティを設定して、デスティネーションの設定に関わらず、このメッセージをトレースするように指定します。以降のセクションでは、デスティネーションとメッセージのトレースプロパティについて説明します。

メッセージのトレース情報は、システムコンソールまたはログファイルのいずれかに出力することができます。**MSG** トレースオプションを設定して、メッセージのトレース情報を表示し、**DEFAULT** (デフォルト) のトレースオプションに **MSG** オプションが含まれるように指定します。トレースオプションについての詳細は、435 ページの「サーバー上でのトレース」を参照してください。

メッセージのトレース情報を表示するには、デスティネーションまたはメッセージのいずれかのトレースプロパティを設定し、さらに、システムコンソールまたはログファイルの **MSG** または **DEFAULT** トレースオプションを設定する必要があります。



EMS のトレース機能では、表示できない文字をトレース出力から除外できません。アプリケーションがメッセージ内に表示できない文字を使用している場合 (ヘッダまたはデータのいずれの場合も)、メッセージトレースは、予期しない結果になることがあります。

### デスティネーションのメッセージトレースの有効化

デスティネーションの **trace** プロパティを設定して、そのデスティネーションのトレース情報を生成するように指定します。

この **trace** プロパティには、オプションとして **trace=body** を指定することができます。**trace=body** を設定すると、トレースメッセージにメッセージ本文が含まれます。EMS サーバーでは、最大 1 KB のメッセージ文字列フィールドを、8 KB の合計メッセージサイズを出力します。トレースメッセージは、メッセージ全体が出力されたかどうかを示します。

**body** オプションを指定せずに **trace** プロパティを設定すると、メッセージシーケンスとメッセージ ID のみを含むトレースメッセージを生成します。

デスティネーションのメッセージトレースを有効にすると、メッセージ処理で発生する以下の各イベントについてトレース情報を生成します。

- メッセージがデスティネーションに受信される
- メッセージがコンシューマに送信される
- メッセージが外部システムからインポートされる、または外部システムへエクスポートされる



- メッセージが確認応答される
- メッセージがデスティネーションブリッジ経由で送信される
- メッセージがルーティングされる

要求メッセージへの応答は、応答デスティネーションには **trace** プロパティが設定されている場合にのみトレースされます。同様に、エクスポートされたメッセージへの応答は、**trace** プロパティが設定されている場合にのみトレースされます。

## メッセージのメッセージトレースの有効化

個々のメッセージをトレースするには、メッセージの **JMS\_TIBCO\_MSG\_TRACE** プロパティを設定します。プロパティの値には、**null** (Java/.NET では「**null**」、C の場合は「**NULL**」) または文字列「**body**」のいずれかを指定します。**null** を設定すると、メッセージ ID とメッセージシーケンスのみを含むメッセージのトレース情報を生成します。「**body**」を設定すると、メッセージ本文を含むメッセージのトレースエントリを生成します。

メッセージの **JMS\_TIBCO\_MSG\_TRACE** プロパティを設定すると、メッセージが通過するデスティネーションに **trace** プロパティが設定されているかどうかに関わらず、処理に従ってメッセージのトレース情報が生成されます。メッセージのトレースメッセージは、プロデューサがメッセージを送信したときと、コンシューマがメッセージを受信したときに生成されます。

## サーバーイベントの監視

---

TIBCO Enterprise Message Service サーバーは、内部システムイベントの発生について、トピックメッセージをパブリッシュすることができます。たとえば、ユーザーがサーバーに接続したときや接続を切断したときに、サーバーからメッセージをパブリッシュすることができます。システムイベントのメッセージには、メッセージのプロパティに格納されたイベントについての詳細情報が含まれます。このセクションでは、サーバーが提供する監視機能についてその概要を説明します。監視のトピックおよび各トピックのメッセージプロパティの説明については、521 ページの付録 A、「メッセージの監視」のリストを参照してください。

### システム監視トピック

TIBCO Enterprise Message Service サーバーは、システムイベントの発生時に、さまざまなトピックにメッセージをパブリッシュすることができます。イベントクラスにはさまざまなタイプがあり、各クラスは関連するイベントをグループ化しています。たとえば、接続、管理、およびルートなどのイベントクラスがあります。各イベントクラスは、さらに細かく各クラスに属するイベントに分けられます。たとえば、接続クラスには、接続と切断のふたつのイベントがあります。これらのイベントクラスを使って、システムイベントは意味のあるカテゴリにグループ化されています。

システムイベントのトピック名は、`$sys.monitor.` という文字列で始まります。名前の残りの部分には、イベントクラス、さらにイベント名が続きます。たとえば、クライアントがサーバーの接続を切断すると、サーバーからトピック

`$sys.monitor.connection.disconnect` へメッセージがパブリッシュされます。システムイベントのトピックの名前付けスキーマに基づいて、あるクラスのすべてのイベントにワイルドカードサブスクリプションを作成できます。たとえば、クライアントがサーバーに接続または接続を切断したときにメッセージを受け取るようにするには、トピック `$sys.monitor.connection.*` のサブスクライバを作成します。

監視トピックは、サーバーによって作成および管理されます。監視トピックは、`topics.conf` ファイルにはリストされません。監視トピックをサブスクライブすることはできますが、監視トピックを作成することはできません。

### メッセージの監視

デスティネーションが行うメッセージの送信、受信、または確認応答などの処理を監視することができます。また、期限切れ、破棄、または `maxRedelivery` エラーなどによって途中で終了したメッセージも監視できます。

メッセージを監視するには、トピック `$sys.monitor` を次の形式で指定します。

`$sys.monitor.D.E.destinationName`

D にはデスティネーションのタイプ、E には監視するイベント、*destinationName* にはそのメッセージを監視するデスティネーションの名前を指定します。表 73 に、D と E に指定可能な値を示します。

表 73 メッセージ監視修飾子（1 / 2 シート）

修飾子	値	説明
D	T	監視対象のデスティネーションはトピックです。監視メッセージ内のメッセージ本文は <b>byte</b> 型配列です。監視メッセージを表示する場合に、必要に応じて、メッセージ本文を再作成するには、 <b>createFromBytes()</b> メソッドを使用します。
	t	監視対象のデスティネーションはトピックです。監視メッセージ内にメッセージ本文を指定しません。
	Q	監視対象のデスティネーションはキューです。監視メッセージ内のメッセージ本文は <b>byte</b> 型配列です。監視メッセージを表示する場合に、必要に応じて、メッセージ本文を再作成するには、 <b>createFromBytes()</b> メソッドを使用します。
	q	監視対象のデスティネーションはキューです。監視メッセージ内にメッセージ本文を指定しません。

表 73 メッセージ監視修飾子（2 / 2 シート）

修飾子	値	説明
E	s	サーバーが以下のオブジェクトにメッセージを送信すると監視メッセージが生成されます。 <ul style="list-style-type: none"><li>• コンシューマ</li><li>• ルート</li><li>• トランスポート経由による外部のシステム</li></ul>
	r	指定したデスティネーションがメッセージを受信すると監視メッセージが生成されます。次のようなメッセージを受信すると生成されます。 <ul style="list-style-type: none"><li>• プロデューサから送信されたメッセージ。</li><li>• ルートから送信されたメッセージ。</li><li>• ブリッジ経由で他のデスティネーションから転送されたメッセージ</li><li>• トランスポート経由で外部システムにインポートされたメッセージ</li></ul>
	a	メッセージが確認応答されると監視メッセージが生成されます。
	p	メッセージが、期限切れ、破棄、または <code>maxRedelivery</code> エラーなどによって途中で終了した場合、監視メッセージが生成されます。
	*	指定したデスティネーション宛てのメッセージが送信、受信、または確認応答されると監視メッセージが生成されます。

たとえば、`$sys.monitor.T.r.corp.News` は、`corp.News` という名前のトピックに対するすべての受信メッセージを監視するトピックです。受信したメッセージのメッセージ本文は、このトピックの監視メッセージに含まれます。トピック `$sys.monitor.q.*.corp.*` は、名前 `corp.*` に一致するすべてのキューに対して、すべてのメッセージイベント（送信、受信、確認応答）を監視します。メッセージの本文は、このトピックのメッセージには含まれません。

このタイプの監視トピックに送信されるメッセージには、イベントの説明、およびメッセージの送信元（プロデューサ、ルート、外部システムなど）の情報が含まれます。さらに、*D* の値によっては、メッセージ本文が含まれる場合もあります。メッセージの監視で指定可能なプロパティについての詳細は、521 ページの [付録 A、「メッセージの監視」](#) を参照してください。

メッセージを監視するトピックは、明示的にサブスクライブする必要があります。つまり、`$sys.monitor.>` をサブスクライブすることで `$sys.monitor` から始まる名前のトピックをすべてサブスクライブすることになります。

`$sys.monitor.T.*.foo.bar` などの特定のメッセージを監視するトピックはサブスクライブしません。ただし、別のサブスクライバがそのメッセージを監視するトピックに興味を示した場合、このサブスクライバもこれらのメッセージを受信します。

メッセージ監視トピック名の *destinationName* の部分にワイルドカードを指定することで、そのワイルドカードに一致するすべてのデスティネーションのメッセージ監視トピックをサブスクライブすることができます。たとえば、トピック `$sys.monitor.T.r.>` をサブスクライブすると、すべてのトピックが受け取るすべてのメッセージを監視できます。システムパフォーマンスの理由から、あまり多くのメッセージ監視トピックをサブスクライブしないようにしてください。詳細は、446 ページの「[監視トピックのパフォーマンスへの影響](#)」を参照してください。

## 監視トピックの表示

監視トピックの扱いは、他のトピックと同様です。監視トピックを表示するには、目的のトピックをサブスクライブするクライアントアプリケーションを作成します。

監視トピックには、潜在的に機密システム情報が含まれるため、クライアントが監視トピックにアクセスするときは認証やアクセス権限が常にチェックされます。つまり、サーバーの認証が無効なときでも、有効なユーザー名とパスワードでログインしていなければ、また、ユーザーにトピックの表示権限がなければ、クライアントは監視トピックにアクセスできません。

`admin` ユーザーと `$admin` グループに属するユーザーには、監視トピックのサブスクリプションも含め、サーバーのアクションを実行するアクセス権限が付与されています。それ以外のユーザーが監視トピックのサブスクライバを作成できるようにするには、監視トピックを表示できるアクセス権限を明示的に、ユーザーに付与する必要があります。たとえば、`$admin` グループのメンバーではない `BOB` というユーザーに、すべての接続イベントを監視できることを許可するには、管理ツールを使って次のコマンドを作成し、接続イベントの監視に必要なアクセス権限を `BOB` に付与します。

```
grant topic $sys.monitor.connection.* BOB subscribe
```

これで、ユーザー `BOB` のクライアントアプリケーションからトピック

`$sys.monitor.connect.*` のサブスクライバを作成したり、接続イベントおよび接続の切断イベントを監視したりできるようになります。



名前が `$sys.monitor` から始まるトピックは、`$sys.monitor` から始まる親トピック以外の親トピックのアクセス権限を継承しません（つまり、`*.*` や `*.>` のワイルドカードを使用しても、`$sys.monitor` の親を指定できません）。

したがって、ユーザーに `>` をサブスクライブできるアクセス権限を与えても、ユーザーは `$sys.monitor` で始まる名前のトピックをサブスクライブすることはできません。このようなトピックをユーザーがサブスクライブできるようにするには、`$sys.monitor` で始まる名前のトピック（または `$sys.monitor.admin.*` などのパターンが付いた親トピック）へアクセスできる権限を明示的に付与する必要があります。

監視トピックは **MapMessage** タイプのメッセージをパブリッシュします。イベントについての詳細情報は、メッセージのプロパティ内に格納されます。各システムイベントには、それぞれ固有のプロパティがあります。個々の監視トピックおよびその監視トピックにパブリッシュされるメッセージのメッセージプロパティについては、521 ページの **付録 A、「メッセージの監視」** を参照してください。クライアントアプリケーションは、トピックに送信された他のタイプのメッセージを扱うのと同じように、監視メッセージの全体または一部を受信、表示できます。ただし、標準メッセージとは異なる監視メッセージの処理がいくつかあります。

- 監視メッセージは他のサーバーにルーティングできません。
- 監視メッセージは永続的にディスクに保存されません。
- 監視メッセージをプロセスメモリからディスクにスワップすることはできません。

監視メッセージをサブスクライブするアプリケーションの数に制限はありません。監視トピックの数だけアプリケーションを作成し、各監視トピックに個別にサブスクライブすることができます。また、ひとつのアプリケーションを作成して、そのアプリケーションから必要な限りすべての監視トピックをサブスクライブすることもできます。また、監視トピックのサブスクライバは、メッセージセレクタを使用して監視メッセージをフィルタし、必要なメッセージだけをアプリケーションが受信するように設定することができます。

## 監視トピックのパフォーマンスへの影響

TIBCO Enterprise Message Service サーバーは、現在サブスクライバが存在する監視トピックのみメッセージを生成します。監視トピックをサブスクライブするアプリケーションがない場合、監視メッセージは生成されません。監視メッセージを生成するとシステムリソースが消費されるので、環境に応じて監視対象の内容を考慮する必要があります。システムパフォーマンスは、監視トピックに対するサブスクリプションの数やこれらのトピックに対するメッセージの頻度によって影響を受けます。

開発やシステムのテストが目的の場合は、すべてのシステムイベントを監視するとよいでしょう。通常、開発やシステムテストの環境ではメッセージの数量は大きくならないため、監視によってシステムの問題に関する情報を得ることができます。

実稼動システムですべてのイベントを監視すると、システムパフォーマンスの低下を招く恐れがあります。したがって、システムの実稼動環境では、`$sys.monitor.>` のトピックサブスクライバを作成しないようにしてください。また、将来のリリースで監視イベントが追加され、監視トピックの数が増える可能性もあります。システム実稼動環境での監視トピックのサブスクリプションは、特定の監視トピック、または必要な監視トピックの特定のクラスへのワイルドカードサブスクリプションのみに制限することをお勧めします。

さらに、各監視トピックにメッセージが配信される頻度も考慮する必要があります。トピックやルートを作成、アクセス権限の変更などのシステム管理イベントは頻繁には行われないため、これらのタイプのサブスクリプションを作成しても、ほとんどの場合、システムパフォーマンスにはあまり影響を与えません。

また、メッセージセレクトラを使って管理メッセージを制限すると、システムパフォーマンスが、わずかですが向上することがあります。サブスクライバのメッセージセレクトラに一致しないメッセージは、サーバーから送信されません。送信されることがなくても、メッセージは常に生成されます。したがって、そのトピックのすべてのメッセージがサブスクライバのメッセージセレクトラのフィルタに一致しない場合でも、監視トピックのサブスクライバによってシステムにオーバーヘッドが生じることがあります。

## サーバーの統計情報の操作

TIBCO Enterprise Message Service サーバーでは、サーバー全般に関してだけではなく、各プロデューサ、コンシューマ、またはルートで送受信されるメッセージの数量、メッセージのサイズ、およびメッセージの統計情報を追跡できます。収集される統計情報の種類、平均値を求める間隔、および各タイプのメッセージ量の詳細を設定することができます。

統計情報の追跡は、サーバーの設定ファイルで設定できます。または、管理ツールのコマンドを使用するか、管理 API を使用するアプリケーションを作成することで、設定を動的に変更することができます。

統計情報は管理ツールで表示できますが、より詳細な統計情報が分析できるよう、管理 API を使用する独自のアプリケーションを作成することもできます。

このセクションでは、設定ファイルと管理ツールのコマンドを使用した統計情報の設定および表示の方法について説明します。管理 API についての詳細は、オンラインマニュアルの `com.tibco.tibjms.admin` の説明を参照してください。



TIBCO Enterprise Message Service サーバーは、送受信されるメッセージの数量を追跡しますが、追跡するのは、プロデューサ、コンシューマ、またはルートで送受信されるメッセージのみです。サーバーはシステムメッセージも送信しますが、これらはメッセージの数には含まれません。

ただし、サーバーが内部使用するために、サーバーがメッセージに少量のデータを埋め込むことがあります。このオーバーヘッド部分のデータは、メッセージサイズの合計の計算に入れられます。そのため、メッセージによっては、予想していたものよりメッセージのサイズが大きくなることがあります。

## サーバーの全般統計情報

サーバーは常に、ある種のサーバーの全般統計情報を収集しています。この統計情報には、メッセージの着信および発信率（1 秒あたりに転送されるメッセージの数で表す）、メッセージによるメモリ使用率とディスク使用率、デスティネーション数、接続数、および持続サブスクリプション数が含まれます。これらの統計情報の収集にはシステムリソースはほとんど消費されないため、統計情報はいつでも表示することができます。サーバーの全般統計情報を表示するには、`show server` コマンドを実行します。

デフォルトでは、サーバーの全般統計情報は 1 秒間隔で収集されます。システム稼動状況の平均統計情報は、比較的長い間隔で表示することができます。`server_rate_interval` 設定パラメータで、サーバー統計情報の収集間隔を制御します。このパラメータは設定ファイルで設定できますが、`set server` コマンドを使って動的に設定することもできます。このパラメータには、ゼロより大きい正の整数しか設定できません。



## 統計情報収集の有効化

各プロデューサ、コンシューマ、デスティネーション、およびルートについて、全体的な統計情報とそれぞれのデスティネーションごとの統計情報を収集することができます。統計情報の収集を有効にする場合は、`statistics` パラメータを `enabled` に設定します。このパラメータは設定ファイルで設定できますが、`set server` コマンドを使って動的に設定することもできます。

`statistics` パラメータを使用すれば、統計情報収集の有効化および無効化をグローバルに指定することができます。各オブジェクトの統計情報は、そのオブジェクトが存在する限りサーバーのメモリに保持されています。各オブジェクトの統計情報の合計値をゼロにリセットするには、統計情報の収集を無効にして、再度有効にします。サーバーの統計情報は、またサーバーがシャットダウンして再起動した場合や、フォールトトレランスのフェイルオーバーが発生した場合にもリセットされます。

各プロデューサ、コンシューマ、デスティネーション、およびルートについて、送受信されたメッセージの総計およびメッセージの合計サイズなどの全般統計情報は保持されます。また、プロデューサとコンシューマについては、メッセージの送受信に使用した各デスティネーションに関するそれぞれの統計情報が保持されます。

メッセージの送受信率とメッセージのサイズは、一定の間隔で計算されます。デフォルトでは、3 秒間隔で平均値が計算されます。この値を増減するには、`rate_interval` パラメータの値を変更します。このパラメータは設定ファイルで設定できますが、`set server` コマンドを使って動的に設定することもできます。このパラメータをゼロに設定すると、一定の間隔での統計情報の追跡が無効になり、デスティネーション、ルート、プロデューサ、またはコンシューマの統計情報の合計のみが保持されます。

プロデューサ、コンシューマ、デスティネーション、およびルートに関する統計情報の合計の収集には、システムリソースはあまり消費されません。通常の場合、統計情報の収集および平均値の計算を有効にしても、システムパフォーマンスには影響を与えません。

## 詳細統計情報

場合によっては、デフォルトで収集される統計情報では不十分な場合があります。たとえば、トピックサブスクライバがワイルドカードトピックにサブスクライブすると、ワイルドカードのパターンに一致する各トピックの情報を合計した統計情報が保持されます。この場合、さらに詳しい情報を収集し、サブスクライバが受信した実際のトピックごとの統計情報を追跡することができます。

以下の場合、詳細統計情報の収集が必要になります。

- ワイルドカードトピックをサブスクライブするトピックサブスクライバ。
- メッセージの作成時にデスティネーションを指定しないメッセージプロデューサ。これらのメッセージプロデューサは、どのデスティネーションに対してもメッセージをプロデュースすることができ、デスティネーション名はメッセージの送信時に指定されます。
- ルートで、多数のトピックに関してメッセージが送受信される。

- チャネルでも、多数のトピックに関してメッセージが送信される。

詳細統計情報を有効にするには、`detailed_statistics` パラメータに、受信する統計情報のタイプを設定します。パラメータには、次の値を指定できます。

- `NONE` : 詳細統計情報の収集を無効にします。
- `CONSUMERS` : ワイルドカードのトピック名でトピックサブスクライバの詳細統計情報を有効にします。
- `PRODUCERS` : デスティネーションを指定しないプロデューサの詳細統計情報を有効にします。
- `ROUTES` : ルートの詳細統計情報を有効にします。
- `CHANNELS` : チャネルの詳細統計情報を有効にします。

`detailed_statistics` パラメータを `NONE` に設定するか、`CONSUMERS`、`PRODUCERS`、`ROUTES`、または `CHANNELS` の値を任意に組み合わせることができます。詳細統計情報の収集タイプを複数指定する場合は、値をコンマで区切って指定します。`detailed_statistics` パラメータは設定ファイルで設定できますが、`set server` コマンドを使って動的に設定することもできます。たとえば、`set server` コマンドを使って次のように入力すると、プロデューサとルートの詳細情報統計の追跡が有効になります。

```
set server detailed_statistics = PRODUCERS,ROUTES
```

詳細統計情報の収集はメモリを多く消費するため、大量の統計情報を収集するとシステムパフォーマンスが低下することがあります。詳細統計情報を収集する際のシステムリソースの消費を制御するために、ふたつのパラメータが用意されています。まず、統計情報を保持する期間を制御することができ、次に、詳細統計情報の収集に使用するメモリの最大使用量を設定することができます。アプリケーションプログラムによってデスティネーションが動的に多数作成されるような場合は、詳細統計情報を収集しないことをお勧めします。

`statistics_cleanup_interval` パラメータを使って、詳細統計情報の保持期間を制御します。このパラメータは設定ファイルで設定できますが、`set server` コマンドを使って動的に設定することもできます。デフォルトでは、15 秒間分の統計情報が保持されます。たとえば、トピック `foo.*` のトピックサブスクライバがある場合、そのサブスクライバはトピック `foo.bar` のメッセージを受信します。15 秒以内に新しいメッセージがトピック `foo.bar` に届かないと、そのコンシューマ (サブスクライバ) 宛てのトピック `foo.bar` についての統計情報は削除されます。このパラメータをゼロに設定すると、すべての統計情報を永久に保持できます。ただし、オブジェクトの統計情報が保持されるのは、オブジェクトが存在している間だけです。したがって、メッセージコンシューマがクライアントアプリケーションを終了すると、そのコンシューマに関するすべての詳細統計情報がメモリから削除されます。

`max_stat_memory` パラメータを設定して、詳細統計情報に使用されるメモリの最大サイズを制御します。このパラメータは設定ファイルで設定できますが、`set server` コマンドを使って動的に設定することもできます。デフォルトでは、このパラメータはゼロに設定されており、詳細統計情報のメモリ制限はありません。単位を指定しない場合、このパラメータの値はバイト単位です。単位としては、KB、MB、または GB をオプションで指定できます。指定したメモリ制限に達すると、サーバーは詳細統計情報の収集を中止します。使用されているメモリの容量が減少すると（たとえば、`statistics_cleanup_interval` パラメータの設定によって古い統計情報が削除されると）、サーバーは統計情報の収集を再開します。

## 統計情報の表示

統計情報の収集を有効にした場合、管理ツールで `show stat` コマンドを使用すると、プロデューサ、コンシューマ、ルート、およびデスティネーションの統計情報を表示できます。

`show stat` コマンドを指定すると、デスティネーション名、ユーザー名、接続 ID、またはこれらの条件のいずれかの組み合わせで、統計情報の表示をフィルタできます。オプションで `total` キーワードを指定すると、統計情報の合計のみを表示することができます（この場合、詳細統計情報は出力されません）。また、オプションで「`wide`」キーワードを指定すると、デスティネーションまたはルートの統計情報を表示できます。このキーワードを指定すると、インバウンドとアウトバウンドのメッセージ統計情報を同じ行に表示します（1 行に 100 文字以上表示できます）。

以下に、詳細情報統計の追跡が有効になっている場合に、ルートの統計情報がどのように表示されるかの例を示します。

```
tcp://server1:7322> show stat route B
Inbound statistics for route 'B':
```

	Total Count		Rate/Second	
Destination	Msgs	Size	Msgs	Size
<total>	189	37.9 Kb	10	2.0 Kb
Topic: dynamic.0	38	7.6 Kb	2	0.4 Kb
Topic: dynamic.1	38	7.6 Kb	2	0.4 Kb
Topic: dynamic.2	38	7.6 Kb	2	0.4 Kb
Topic: dynamic.3	38	7.6 Kb	2	0.4 Kb
Topic: dynamic.4	37	7.4 Kb	2	0.4 Kb

```
Outbound statistics for route 'B':
```

	Total Count		Rate/Second	
Destination	Msgs	Size	Msgs	Size
<total>	9538	1.9 MB	10	2.1 Kb
Topic: dynamic.0	1909	394.9 Kb	2	0.4 Kb
Topic: dynamic.1	1908	394.7 Kb	2	0.4 Kb
Topic: dynamic.2	1907	394.5 Kb	2	0.4 Kb
Topic: dynamic.3	1907	394.5 Kb	2	0.4 Kb
Topic: dynamic.4	1907	394.5 Kb	2	0.5 Kb

`show stat` コマンドおよびその構文についての詳細は、[161 ページ](#)の「`show stat`」を参照してください。



## 第 18 章

## SSL プロトコルの使用

SSL (Secure Sockets Layer) は、セキュアな認証を提供し、暗号されたデータをインターネットや内部ネットワークで転送するためのプロトコルです。ほとんどの Web ブラウザは SSL をサポートしており、多くの Web サイトや Java アプリケーションは SSL プロトコルを使用して、クレジットカード番号などのユーザーの個人情報を取得しています。

SSL プロトコルは複雑なため、この章では SSL のすべてを説明することはできませんが、代わりに、TIBCO Enterprise Message Service サーバーおよびサーバーと通信するクライアントアプリケーションで SSL を設定する方法を説明します。SSL についての詳細は、<http://wp.netscape.com/eng/ssl3/> サイトの SSL 仕様を参照してください。

## トピック

- 「TIBCO Enterprise Message Service での SSL のサポート」、454 ページ
- 「デジタル証明書」、455 ページ
- 「証明書と鍵のファイル名」、457 ページ
- 「サーバーでの SSL の設定」、459 ページ
- 「EMS クライアントでの SSL の設定」、460 ページ
- 「暗号スイートの指定方法」、464 ページ
- 「認証にのみ SSL を使用する」、470 ページ
- 「FIPS 準拠の有効化」、471 ページ

## TIBCO Enterprise Message Service での SSL のサポート

TIBCO Enterprise Message Service は、SSL をサポートしています。SSL は、ネットワーク接続で転送されるデータを公開鍵と秘密鍵を使って暗号化し、以下のネットワークコンポーネント間の通信セキュリティを強化します。

- EMS クライアントと `tibemsd` サーバー間
- `tibemsdadmin` ツールと `tibemsd` サーバー間
- ふたつのルーティング先サーバー間
- ふたつのフォールトトレランスサーバー間

SSL は、EMS サーバーで利用可能な認証用の他のメカニズムと連携してセキュアな通信を提供します。サーバーで `authorization` が有効になっている場合、接続には 2 フェーズの認証プロセスが使用されます。最初に、クライアントとサーバー間の SSL ハンドシェイクでセキュアな接続を初期化します。次に、指定されたユーザー名とパスワードを使用して、EMS サーバーがクライアントの認証情報をチェックします。接続しているクライアントで有効なユーザー名とパスワードの組み合わせが指定されなかった場合、SSL 67 が成功した場合でも接続は失敗します。



認可が有効になっている場合、セキュアな SSL 接続でも常にユーザー名とパスワードが確認されます。

### 実装

TIBCO Enterprise Message Service サーバーと C クライアントライブラリは、OpenSSL を使用して SSL をサポートします。詳細は、[www.openssl.org](http://www.openssl.org) を参照してください。

EMS Java クライアントは、Sun JavaSoft 社が提供する JSSE または Entrust 社が提供する SSL 実装のいずれかを使用できます。JSSE は EMS Java インストールに付属しています。Entrust を使用したい場合は、Entrust SSL を別途購入し、インストールする必要があります。

EMS .NET 2.0 クライアントは、Microsoft の SSL 実装を使用しています。Microsoft の SSL 実装は、OpenSSL と互換性があります。クライアントで必要とされる証明書は、ファイルまたは Microsoft 証明書ストアのいずれかに格納されています。ただし、証明書ファイルが使用されている場合でも、Microsoft では Microsoft 証明書ストアにルート証明書がインストールされている必要があります。

通常、EMS のディストリビューションには、その製品のリリース時に入手可能な OpenSSL および OpenLDAP の最新バージョンのビルドが組み込まれています。正確なバージョン番号については、TIBCO Enterprise Message Service の TIBCO Software Inc. エンドユーザーライセンス契約に記載された「第三者ソフトウェアライセンス契約」を参照してください。

## デジタル証明書

デジタル証明書は、身元を示すデータ構造のファイルです。この証明書を使用して、EMS は、サーバーとクライアントの身元を確認します。サーバーとクライアントが、SSL 上でデータを交換するためには、必ずしも互いを検証する必要はありませんが、証明書を使用すれば、より高いレベルのセキュリティを実現できます。

デジタル証明書は、信頼できるサードパーティの認証局（CA : Certificate Authority）または企業内のセキュリティ管理者のいずれかによって発行されます。通常、ネットワーク上の各ユーザーとサーバーには、データが正しい送信元から送られ、正しいデスティネーションで受信されたことを確認できるように、固有の証明書が必要です。

SSL をサポートするため、EMS サーバーにはデジタル証明書が必要です。オプションで、EMS クライアントに証明書を発行することもできます。クライアントの証明書を検証するようにサーバーが設定されている場合、クライアントには証明書が必要になり、その証明書はサーバーによって検証される必要があります。同様に、サーバーの証明書を検証するように EMS クライアントを設定することができます。サーバーおよびクライアントの ID がいったん確認されると、SSL プロトコルで、暗号化されたデータをサーバーとクライアント間で転送できるようになります。

デジタル証明書は、証明書の所有者（ユーザーまたはサーバー）を特定できる公開部分と、所有者の秘密を守る秘密鍵のふたつの部分で構成されています。

デジタル証明書の公開部分には、以下のようなさまざまな情報が含まれます。

- 所有者の名前およびサブジェクトの身元を確認するために必要なその他の情報。この情報には、デジタル証明書を使用している Web サーバーの URL や電子メールアドレスなどが含まれます。
- サブジェクトの公開鍵。
- デジタル証明書を発行した CA の名前。
- シリアル番号。
- 開始日と終了日で定義された証明書の有効期間。

最もよく使用されているデジタル証明書は ITU-T X.509 です。TIBCO Enterprise Message Service は、X.509 バージョン 3（X.509v3）に準拠したデジタル証明書をサポートしています。Verisign や Entrust など、ほとんどの CA はこの標準に準拠しています。

## デジタル証明書形式

TIBCO Enterprise Message Service は以下のデジタル証明書形式をサポートしています。

- PEM (Privacy Enhanced Mail)
- DER (Distinguished Encoding Rules)
- PKCS#7
- PKCS#12
- Java KeyStore (クライアントのデジタル証明書用)
- Entrust Store (クライアントのデジタル証明書用)

## 秘密鍵形式

TIBCO Enterprise Message Service は以下の秘密鍵形式をサポートしています。

- PEM (Privacy Enhanced Mail)
- DER (Distinguished Encoding Rules)
- PKCS#8
- PKCS#12

EMS サーバーは OpenSSL を使用して秘密鍵を読み込みます。EMS サーバーは PEM、DER、PKCS8、および PKCS12 形式をサポートしていますが、Java KeyStore または Entrust ストアファイルは読み込めません。



## 証明書と鍵のファイル名

ID (デジタル証明書)、秘密鍵、発行者 (証明書チェーン)、または信頼する CA リストを指定するすべてのパラメータについて、有効なファイルを指定する必要があります。すべてのファイル形式がクライアントとサーバーでサポートされるわけではありません。各パラメータの説明では、そのパラメータがサポートしているファイル形式を説明します。

表 74 に有効なファイル形式を示します。

表 74 ファイル形式

拡張子	説明
.pem	PEM でエンコードされた証明書と鍵 (証明書と秘密鍵を同じファイルに格納できる)。
.der	DER でエンコードされた証明書。
.p8	PKCS#8 ファイル
.p7b	PKCS#7 ファイル
.p12	PKCS12 ファイル (証明書と秘密鍵を同じファイルに格納できる)。
.jks	Java KeyStore ファイル
.epf	Entrust ストアファイル

証明書が置かれているディレクトリは `EMS_install_dir/certs` です。EMS のインストール時に、サンプルの設定ファイルが使用する証明書と秘密鍵のサンプルがこのディレクトリにインストールされています。

サンプル証明書には以下の種類があります。

- PEM および PKCS8 形式で暗号化されたルートの自己署名証明書および対応する秘密鍵：

```
server_root.cert.pem
server_root.key.pem
server_root.key.p8
```

- PEM および PKCS8 形式で暗号化されたサーバーの証明書および対応する秘密鍵：この証明書は、`server_root.cert.pem` で発行され、サーバーが使用します。

```
server.cert.pem
server.key.pem
server.key.p8
```

- PEM および PKCS8 形式で暗号化されたルートの自己署名証明書および対応する秘密鍵：

```
client_root.cert.pem  
client_root.key.pem  
client_root.key.p8
```

- PEM および PKCS8 形式で暗号化されたクライアントの証明書および対応する秘密鍵：この証明書は、`client_root.cert.pem` で発行され、クライアントが使用します。

```
client.cert.pem  
client.key.pem  
client.key.p8
```

- `client.cert.pem` クライアント証明書、`client.key.pem` クライアント秘密鍵、および `client_root.cert.pem` 発行者証明書を含む PKCS12 ファイル：

```
client_identity.p12
```

## サーバーでの SSL の設定

SSL を使用するには、`tibemspd` の各インスタンスに、デジタル証明書と秘密鍵が必要です。オプションで、サーバーが証明書チェーンまたは信頼する証明書を必要とする場合があります。

クライアントからの SSL 接続をリスンするようにサーバーを設定するには、`tibemspd.conf` ファイルの `listen` パラメータを使用します。ポートが SSL 接続を受け入れるように指定するには、`listen` パラメータを次のように指定します。

```
listen = ssl://localhost:7243
```

### SSL パラメータ

`tibemspd.conf` ファイル内に複数の SSL サービスを設定できます。最低限の設定として、パラメータ (`ssl_server_identity`) をひとつ以上設定する必要があります。ただし、サーバーの証明書ファイルに秘密鍵が含まれていない場合は、`ssl_server_key` に秘密鍵を設定する必要があります。

`tibemspd.conf` ファイルに設定できる SSL サービスについての詳細は、214 ページの「[SSL サーバーのパラメータ](#)」を参照してください。

### コマンドラインオプション

サーバーが受け付ける SSL のコマンドラインオプションがいくつかあります。

`tibemspd` サーバーを起動する際に、以下のオプションを指定できます。

- `-ssl_trace`: 読み込んだ証明書のトレースを有効にします。これにより、サーバーの起動時に、読み込んだ証明書を説明するメッセージをシステムコンソールに表示します。
- `-ssl_debug_trace`: 実稼動システムでの使用のためではなく、デバッグで使用するために詳細な SSL トレースを有効にします。
- `-ssl_password`: 秘密鍵のパスワードを指定します。または、`tibemspd.conf` ファイルの `ssl_server_password` パラメータにこのパスワードを指定することができます。これらのいずれの方法でもパスワードを指定しない場合、`tibemspd` の起動時にパスワードの入力が求められます。詳細は、`ssl_password` 設定パラメータの説明を参照してください。

## EMS クライアントでの SSL の設定

---

EMS サーバーに対して SSL 接続を行う場合、クライアントは CLASSPATH に JAR ファイルをインクルードする必要があります。

- `tibcrypt.jar`
- `slf4j-api-1.4.2.jar`
- `slf4j-simple-1.4.2.jar`

これらの JAR は TIBCO Enterprise Message Service のインストールに付属されており、`EMS_HOME\lib` ディレクトリに格納されています。

**Entrust** EMS クライアントで Entrust を使用するには、Entrust ライブラリを別途購入し、インストールする必要があります。Entrust ライブラリを使用するには、CLASSPATH で `tibcrypt` JAR ファイルの「前に」それらのライブラリを指定しなければなりません。JDK で Entrust を使用するには、Sun の Web サイトから無制限の強度ポリシー（`unlimited strength policy`）の JAR ファイルをダウンロードし、ローカルの JDK インストールにインストールする必要があります。インストールと設定についての詳細は、Entrust のマニュアルを参照してください。

## クライアントデジタル証明書

デジタル証明書によるクライアント認証が EMS サーバーによって必要とされる場合（`tibemspd.conf` ファイルの `ssl_require_client_cert` パラメータの説明を参照）、クライアントは自分のクライアント証明書と秘密鍵をいっしょに、以下のいずれかのファイル形式のファイルに保存することができます。

- PKCS#12
- Java KeyStore
- Entrust Store

クライアント証明書ファイルとは別に、秘密鍵ファイルを保存することもできます。この場合、次のいずれかの形式で証明書と秘密鍵を格納する必要があります。

- PEM
- PKCS#8

クライアントのデジタル証明書と秘密鍵に使用するファイル形式は、クライアントが使用する SSL ベンダーによって異なります。JSSE と Entrust はさまざまなファイル形式をサポートしており、形式の異なるファイルの組み合わせも可能です。ファイル形式についての詳細は、各 SSL ベンダーのマニュアルを参照してください。

## SSL の設定

EMS サーバーに接続しているクライアントは、以下のような方法で SSL の特性を設定できます。

- 接続ファクトリを作成して適切な SSL パラメータを指定し、JNDI を使用して接続ファクトリを照合します。接続ファクトリのサーバー URL に SSL プロトコルを指定し、接続ファクトリに適切なパラメータを指定する必要があります。  
  
あらかじめ設定されている接続ファクトリは、多くの使用状況に対応できるように設定されています。SSL パラメータを含む接続ファクトリを EMS で作成する方法についての詳細は、[セキュア接続のための接続ファクトリの作成とセキュア照合の実行](#)を参照してください。
- 接続ファクトリを動的に作成し（[接続ファクトリの動的作成](#)を参照）、[TibjmsSSL](#) クラス（Java）、[tibemsSSLParams](#) タイプ（C）、または [EMSSSL](#) クラス（C#）のいずれかを使用して、グローバルな SSL パラメータをローカルで設定します。

接続ファクトリ内の SSL パラメータを指定すると、[TibjmsSSL](#) クラスに設定されたすべてのグローバル SSL パラメータが無視されます。

### 接続ファクトリの設定

接続ファクトリの設定には、EMS 管理ツールまたは EMS 管理 API を使用できます。[第 6 章、「EMS 管理ツールの使用」](#)を参照してください。

接続ファクトリを設定する場合、[tibemsd.conf](#) で設定できるサーバーのパラメータと同様に、複数の SSL パラメータを指定できます。



接続ファクトリを設定する場合、EMS は SSL パラメータに指定されているファイル名を検証しません。JNDI 実装を使って接続ファクトリが照合されるときに、EMS サーバーは、ファイルの参照を解決しようと試みます。ファイルがサポートされているタイプに一致しない、またはファイルが見つからない場合、JNDI 照合は失敗し、[ConfigurationException](#) が発行されます。



接続ファクトリには（セキュリティ上の理由により）[ssl\\_password](#) が含まれないため、EMS サーバーはユーザー認証用の「接続の作成」呼び出しで指定されたパスワードを使用します。接続の作成パスワードが [ssl\\_password](#) と異なる場合、接続の作成は失敗します。

[表 75](#) に、接続ファクトリに設定可能なパラメータの説明と各パラメータについての補足情報を簡単に示します。各パラメータについての詳細は、[177 ページの「tibemsd.conf」](#)の該当するパラメータの説明を参照してください。

表 75 ConnectionFactory SSL パラメータ (1 / 2 シート)

パラメータ	説明
ssl_vendor	クライアントが使用している SSL 実装のベンダー名。
ssl_identity	クライアントのデジタル証明書。  このデジタル証明書のファイル形式についての詳細は、457 ページの「証明書と鍵のファイル名」を参照してください。
ssl_issuer	クライアントのデジタル証明書の発行者の証明書チェーン。CA ルート証明書を含む、チェーン全体を指定します。クライアントはチェーン内の証明書を、このパラメータに表示される順番で読み取ります。  <b>例</b> ssl_issuer = certs¥CA_root.pem ssl_issuer = certs¥CA_child1.pem ssl_issuer = certs¥CA_child2.pem  このデジタル証明書のファイル形式についての詳細は、457 ページの「証明書と鍵のファイル名」を参照してください。
ssl_private_key	クライアントの秘密鍵。秘密鍵が ssl_identity 内のデジタル証明書に含まれている場合は、このパラメータの設定を省略できます。  このデジタル証明書のファイル形式についての詳細は、457 ページの「証明書と鍵のファイル名」を参照してください。
ssl_trusted	サーバー証明書の発行者が信頼する CA ルート証明書のリスト。CA ルート証明書を指定します。  このデジタル証明書のファイル形式についての詳細は、457 ページの「証明書と鍵のファイル名」を参照してください。
ssl_verify_host	クライアントがサーバーの証明書を検証すべきかどうかを指定します。このパラメータの値は、enabled または disabled です。デフォルトは enabled で、クライアントがサーバーの証明書を検証する必要があることを意味します。  このパラメータが disabled に設定されている場合は、クライアントはサーバーとのセキュアな接続を確立しますが、サーバーの身元は検証しません。

表 75 ConnectionFactory SSL パラメータ (2 / 2 シート)

パラメータ	説明
ssl_verify_hostname	<p>クライアントがサーバーの証明書の CN フィールドに指定された名前を検証すべきかどうかを指定します。このパラメータの値は、<b>enabled</b> および <b>disabled</b> です。デフォルトは <b>enabled</b> で、クライアントが接続先のホストの名前、または <b>ssl_expected_hostname</b> パラメータに指定されている名前を、サーバーの証明書の値に照らし合わせて検証する必要があることを意味します。名前が一致しない場合、クライアントは接続を拒否します。</p> <p>このパラメータが <b>disabled</b> に設定されている場合は、クライアントはサーバーとのセキュアな接続を確立しますが、サーバーの名前は検証しません。</p>
ssl_expected_hostname	<p>サーバーの証明書の CN フィールドに含まれる名前として、クライアントが想定する名前。このパラメータが設定されていない場合は、必要な名前は、サーバーのホスト名になります。</p> <p>このパラメータの値は、<b>ssl_verify_hostname</b> パラメータが <b>enabled</b> に設定されている場合に使用されます。</p>
ssl_ciphers	<p>クライアントが使用可能な暗号スイートを指定します。</p> <p>暗号スイートの名前を複数指定する場合は、コロンで名前を区切ります。スイート名には、<b>OpenSSL</b> の暗号スイート名または、よりわかりやすい長い名前のいずれかを指定できます。</p> <p>詳細は、464 ページの「<a href="#">暗号スイートの指定方法</a>」を参照してください。</p>
ssl_auth_only	<p>SSL を使用してサーバーとクライアント間のすべての通信を暗号化するか、クライアント認証のみを暗号化するかを指定します。</p> <p>このパラメータが <b>enabled</b> に設定されている場合、クライアントは認証のみに SSL を使用するよう要求します。サーバーはその後のデータ交換に TCP 通信を使用します。このパラメータが <b>disabled</b> に設定されている場合、または値が指定されていない場合、クライアントとサーバー間のすべての通信は SSL で暗号化されている必要があります。</p> <p>この機能についての詳細は、470 ページの「<a href="#">認証にのみ SSL を使用する</a>」を参照してください。</p>
ssl_rand_egd	<p>EGD (Entropy Gathering Daemon) がインストールされている場合、その場所のパスを指定します。このデーモンは、クライアントが使用するランダムなデータを生成します。</p>

## 暗号スイートの指定方法

EMS サーバーの `tibemsd.conf` ファイルで、`ssl_server_ciphers` 設定パラメータを設定して暗号スイートを指定します。サーバー設定ファイルについての詳細は、175 ページの第 7 章、「設定ファイルの使用」を参照してください。

接続ファクトリと接続するクライアントについては、接続ファクトリの `ssl_ciphers` パラメータを設定して暗号スイートを指定します。詳細は、460 ページの「EMS クライアントでの SSL の設定」を参照してください。

### 暗号スイートの構文

EMS は、OpenSSL を使用して SSL をサポートします。したがって、暗号スイート名の指定には、その暗号スイートの OpenSSL 名が使用できます。

暗号スイートを指定する場合、複数の暗号スイートを指定するには、通常、各スイート名をコロン (:) で区切ります。または、スペースとコンマを使用して各スイート名を区切ります。

### Java クライアントの構文

Java クライアントで、暗号スイートのリストを指定する構文は、他の場所で、暗号スイートのリストを指定する構文とは異なります。Java クライアントでは、修飾子 (例: スイートの追加に使用する「+」) に続けて暗号スイート名を指定します。暗号スイート名は、大文字と小文字を区別します。表 76 に、Java クライアントの `ConnectionFactory` に暗号スイート名を指定する際に使用できる修飾子を示します。

表 76 Java クライアントの暗号スイートの修飾子

修飾子	説明
+	暗号リストに暗号を追加する。
-	暗号リストから暗号を削除する。
>	暗号リストの最後に暗号を移動する。
<	暗号リストの先頭に暗号を移動する。
ALL	暗号リスト内のすべての暗号 (NULL 暗号を除く) を指定する。このキーワードを使用すると、すべての暗号を追加または削除できます。  ひとつ以上の暗号スイートを指定する必要があります。暗号スイートがひとつも存在しない場合、SSL 接続の初期化に失敗します。したがって、-ALL を指定する場合は、続けてリストに必要な暗号を追加する必要があります。



次のコード例は、Java クライアントの `ssl_ciphers` 接続ファクトリパラメータに暗号スイートを指定します。

```
-ALL:+RC4-MD5:+DES-CBC-SHA:<DES-CBC3-SHA
```

次のコード例は、フルネームを使用して暗号スイートを指定します。

```
-ALL:+SSL_RSA_WITH_RC4_128_MD5:+SSL_RSA_WITH_DES_CBC_SHA:<SSL_RSA_WITH_3DES_EDE_CBC_SHA
```

その他の暗号スイート指定の構文

Java クライアントの接続ファクトリ以外の場所で、暗号スイートのリストを指定する場合は、OpenSSL の構文を使用します。特に、C クライアントと `ssl_server_ciphers` 設定パラメータは OpenSSL 構文を必要とします。

OpenSSL 構文では、暗号スイート名を指定すると、その暗号スイートが暗号リストに追加されます。各暗号スイート名の先頭には、修飾子を指定することができます。暗号スイート名は、大文字と小文字を区別します。表 77 に、OpenSSL 構文で使用可能な修飾子を示します。

表 77 暗号スイートの OpenSSL の修飾子（1 / 2 シート）

修飾子	説明
¥	リストの最初にこのオプションを入力した場合、EMS は空のリストで開始し、このスラッシュの後に暗号を追加する。  暗号リストの先頭に / を付けない場合、EMS によって暗号リストの先頭に OpenSSL 暗号文字列 DEFAULT が付けられます。  この修飾子は、リストの先頭にのみ使用できます。/ がその他の場所に使用された場合、暗号スイートの構文は無効になり、エラーが発生します。
+	暗号リストの最後に暗号を移動する。  この修飾子は、既存の暗号を移動するために使用します。新しい暗号のリストへの追加には使用できません。
-	暗号リストから暗号を削除する。このオプションを使用した場合、その暗号は、後で暗号リストに追加できます。
!	暗号リスト内の暗号を永久に無効化する。暗号を削除し、さらに削除した暗号を、今後リストに追加しない場合に、このオプションを使用します。この修飾子は、他のすべての修飾子よりも前に置かれます。

表 77 暗号スイートの OpenSSL の修飾子（2 / 2 シート）

修飾子	説明
ALL	暗号リスト内のすべての暗号（NULL 暗号を除く）を指定する。 このキーワードを使用すると、すべての暗号を追加または削除 できます。  少なくともひとつの暗号スイートが指定する必要があります。 暗号スイートがひとつも存在しない場合、SSL 接続の初期化に 失敗します。したがって、-ALL を使用した場合は、ひとつ以上 の暗号をリストに追加する必要があります。
@STRENGTH	鍵の長さを基準に、暗号リストを並べ替える。

次のコード例は、ssl\_server\_ciphers 設定パラメータに暗号スイートを指定します。  
ssl\_server\_ciphers = -ALL:RC4-MD5:DES-CBC-SHA:DES-CBC3-SHA  
次の例は、RC4-MD5 を無効にし、それ以外のすべての暗号を追加します。  
ssl\_server\_ciphers = !RC4-MD5:ALL

デフォルトの暗号  
リスト      EMS サーバーと C クライアントライブラリでは、デフォルトの暗号リストがハー  
ドコーディングされます。このリストは「ALL:!ADH:RC4+RSA:+SSLv2:@STRENGTH」  
と表すことができます。

サポートされる暗号スイート

通常、EMS サーバーと C クライアントライブラリでは、IDEA、RC-5、および CAST  
を除き、OpenSSL がサポートするすべての暗号スイートがサポートされます。暗  
号スイートの全容については、OpenSSL のマニュアルを参照してください。

Java クライアントでサポートされる暗号スイート

Java クライアントでサポートされる暗号スイートは、表 78 に示したスイートの  
みです。参考のため、表では各暗号スイートについて EMS 標準のスイート名と  
OpenSSL のスイート名を併記してあります。

表 78 サポートされる暗号スイート（1 / 4 シート）

スイート名 (OpenSSL 名)	エク ス ポ ー ト	鍵交換	認証	暗号化法	キーサ イズ	MAC
SSL_RSA_WITH_RC4_128_MD5 (RC4-MD5)		RSA	RSA	RC4	128	MD5

表 78 サポートされる暗号スイート (2 / 4 シート)

スイート名 (OpenSSL 名)	エクス ポート	鍵交換	認証	暗号化法	キーサ イズ	MAC
<b>SSL_RSA_WITH_RC4_128_SHA (RC4-SHA)</b>						
		RSA	RSA	RC4	128	SHA1
<b>SSL_RSA_WITH_DES_CBC_SHA (DES-CBC-SHA)</b>						
		RSA	RSA	DES	56	SHA1
<b>SSL_RSA_WITH_3DES_EDE_CBC_SHA (DES-CBC3-SHA)</b>						
		RSA	RSA	3-DES	168	SHA1
<b>SSL_RSA_EXPORT_WITH_RC4_40_MD5 (EXP-RC4-MD5)</b>						
	はい	RSA (512)	RSA	RC4	40	MD5
<b>SSL_RSA_EXPORT_WITH_DES_40_CBC_SHA (EXP-DES-CBC-SHA)</b>						
	はい	RSA (512)	RSA	DES	40	SHA1
<b>SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA (EDH-DSS-DES-CBC3-SHA)</b>						
		DH	DSS	3-DES	168	SHA1
<b>SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA (EDH-RSA-DES-CBC3-SHA)</b>						
		DH	RSA	3-DES	168	SHA1
<b>SSL_DHE_DSS_WITH_DES_CBC_SHA (EDH-DSS-DES-CBC-SHA)</b>						
		DH	DSS	DES	56	SHA1

表 78 サポートされる暗号スイート (3 / 4 シート)

スイート名 (OpenSSL 名)	エクス ポート	鍵交換	認証	暗号化法	キーサ イズ	MAC
SSL_DHE_RSA_WITH_DES_CBC_SHA (EDH-RSA-DES-CBC-SHA)						
		DH	RSA	DES	56	SHA1
SSL_DHE_DSS_EXPORT_WITH_DES_40_CBC_SHA (EXP-EDH-DSS-DES-CBC-SHA)						
	はい	DH (512)	DSS	DES	40	SHA1
SSL_DHE_RSA_EXPORT_WITH_DES_40_CBC_SHA (EXP-EDH-RSA-DES-CBC-SHA)						
	はい	DH (512)	RSA	DES	40	SHA1
TLS_RSA_WITH_AES_128_CBC_SHA (AES128-SHA)						
		RSA	RSA	AES	128	SHA1
TLS_RSA_WITH_AES_256_CBC_SHA (AES256-SHA)						
		RSA	RSA	AES	256	SHA1
TLS_DHE_DSS_WITH_AES_128_CBC_SHA (DHE-DSS-AES128-SHA)						
		DH	DSS	AES	128	SHA1
TLS_DHE_DSS_WITH_AES_256_CBC_SHA (DHE-DSS-AES256-SHA)						
		DH	DSS	AES	256	SHA1
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (DHE-RSA-AES128-SHA)						
		DH	RSA	AES	128	SHA1

表 78 サポートされる暗号スイート（4 / 4 シート）

スイート名 (OpenSSL 名)	エクス ポート	鍵交換	認証	暗号化法	キーサ イズ	MAC
<b>TLS_DHE_RSA_WITH_AES_256_CBC_SHA</b> (DHE-RSA-AES256-SHA)						
		DH	RSA	AES	256	SHA1
<b>SSL_RSA_WITH_NULL_MD5</b> (NULL-MD5)						
JSSE のみ。Entrust ではこのスイートはサポートされません。						
		DH	RSA	なし	—	MD5
<b>SSL_RSA_WITH_NULL_SHA</b> (NULL-SHA)						
JSSE のみ。Entrust ではこのスイートはサポートされません。						
		DH	RSA	なし	—	SHA1

### .NET クライアントでサポートされる暗号スイート

.NET クライアントでサポートされる暗号スイートは、以下のスイートのみです。

- RC4-MD5
- RC4-SHA
- DES-CBC3-SHA
- DES-CBC-SHA
- EXP-RC2-CBC-MD5
- EDH-DSS-DES-CBC3-SHA
- EDH-DSS-DES-SHA
- EXP-RC4-MD5
- AES128-SHA



Windows Server 2008 および Windows 7 などの一部の新しい Windows プラットフォームでは、弱い暗号（EXP-RC4-MD5 など）はサポートされていません。これらのプラットフォームは強い暗号（AES128-SHA など）をサポートしています。特定の Windows プラットフォームでサポートされている暗号についての完全な詳細は、MSDN のマニュアルを参照するか、Microsoft サポートにお問い合わせください。

## 認証にのみ SSL を使用する

---

EMS サーバーでの SSL の使用は、標準的な使い方としてはセキュアデータの交換時に使用しますが、クライアントを認証するためだけに SSL を使用することもできます。このセクションでは、クライアント認証での SSL の使用について説明します。

**使用例** 一部のアプリケーションでは、強力な暗号化された認証が必要とされますが、メッセージの暗号化は必要とされません。

このような場合、アプリケーションの設計者は、NULL 暗号で SSL を設定する方法を選択することもできます。しかし、この方法で SSL 呼び出しを行うと内部的なオーバーヘッドが発生し、メッセージの速度とスループットが低下します。

パフォーマンスの最適化を図るには、クライアントの認証のみに SSL を使用し、それ以降のデータ交換には通常の TCP 通信プロトコルを使用して SSL 呼び出しを行わない方法が適切です。メッセージ処理のパフォーマンスには影響を与えません。

**前提条件** クライアント認証のみに SSL を使用するには、以下の三つの条件をすべて満たす必要があります。

- サーバーとクライアントのどちらにもリリース 4.2 以降のバージョンを使用する必要があります（この条件を満たさない場合、EMS は、接続している間、すべての通信に SSL プロトコルを使用します）。
- サーバーは、`tibemspd.conf` 設定ファイル内の `ssl_auth_only` パラメータを明示的に「enable」に設定する必要があります。
- クライアントプログラムは、クライアント認証のみに SSL を使用する接続を要求する必要があります。クライアントが接続ファクトリでこの要求を指定するには、`ssl_auth_only` パラメータを有効にするか、以下のメソッドまたは関数を呼び出します。

— Java : `TibjmsSSL.setAuthOnly`

— C : `tibemspdSSLParams_SetAuthOnly`

— C# : `EMSSSL.SetAuthOnly`

**関連トピック** サーバーパラメータ : [218 ページの「ssl\\_auth\\_only」](#)  
クライアントパラメータ : [463 ページの「ssl\\_auth\\_only」](#)

## FIPS 準拠の有効化

TIBCO Enterprise Message Service では、連邦情報処理規格（FIPS: Federal Information Processing Standard）140-2 に準拠した実行を有効にできます。

### EMS サーバーの有効化



EMS サーバーは、Windows、Linux、および Solaris 10 (x86) プラットフォームでのみ FIPS 準拠をサポートしています。UNIX では、`tibemsd64`、64 ビットバージョンのサーバーのみがサポートされています。32 ビットはサポートされていません。

EMS サーバーで FIPS 140-2 動作を有効にするには、以下を実行します。

- 主要設定ファイルの `fips140-2` パラメータを `true` に設定する。
- 以下に示す互換性のないパラメータがサーバーの設定ファイルに含まれていないことを確認する。

`fips140-2` が有効になっている場合、EMS サーバーは起動時に FIPS 140-2 に準拠して初期化されます。初期化に成功した場合、このモードで動作していることを示すメッセージが EMS サーバーに表示されます。初期化に失敗した場合、(`startup_abort_list` の設定に関わらず) サーバーは終了します。

#### 互換性のないパラメータ

FIPS 準拠モードで動作するには、`tibemsd.conf` ファイルに以下のパラメータが含まれていないことが必要です。

- `ssl_dh_size`
- `ssl_server_ciphers`
- `ldap_tls_rand_file`
- `ldap_tls_cipher_suite`
- `ft_ssl_ciphers`

また、`routes.conf` ファイルに次のパラメータが含まれていないことが必要です。

- `ssl_ciphers`

## EMS クライアントの有効化

Java および C クライアントアプリケーションは、FIPS 準拠モードで動作できます。

- **Java クライアント** Java クライアントが EMS に付属された JSSE ではなく Entrust の SSL 実装を使用している場合、FIPS 140-2 準拠モードで動作できます。

Java クライアントで FIPS 140-2 動作を有効にするには、以下を実行します。

- EMS メソッドを呼び出す前に、`com.tibco.security.FIPS` プロパティを `true` に設定する。
- インストール済み JDK に対応する Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files をダウンロードしてインストールします。これらのファイルは、Sun Microsystems の Web サイトから入手可能です。

Entrust の使用についての詳細は、460 ページの「[EMS クライアントでの SSL の設定](#)」を参照してください。

- **C クライアント** C クライアントが動的な EMS ライブラリにリンクされている場合、FIPS 140-2 準拠モードで動作できます。静的ライブラリでは FIPS 準拠モードは使用できません。

C クライアントで FIPS 140-2 動作を有効にするには、準拠している OpenSSL ライブラリを使用し、EMS 関数を呼び出す前にライブラリを初期化して FIPS 140-2 動作を有効にします。



C ライブラリは、Windows、Linux、および Solaris 10 (x86) プラットフォームでのみ FIPS 準拠をサポートしています。UNIX では、64 ビットの C ライブラリのみがサポートされています。32 ビットはサポートされていません。



## 第 19 章      フォールトトレランス

この章では、TIBCO Enterprise Message Service のフォールトトレランス機能について説明します。

### トピック

---

- 「フォールトトレランスの概要」、474 ページ
- 「共有ステートフェイルオーバーのプロセス」、476 ページ
- 「非共有ステートフェイルオーバーのプロセス」、480 ページ
- 「共有ステート」、483 ページ
- 「フォールトトレランスサーバーの設定」、487 ページ
- 「Central Administration でのフォールトトレランスの設定」、490 ページ
- 「フォールトトレランスサーバー用のクライアントの設定」、492 ページ
- 「非共有ステート接続用のクライアントの設定」、495 ページ

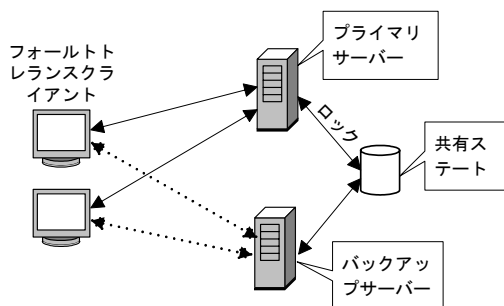
## フォールトトレランスの概要

TIBCO Enterprise Message Service サーバーをフォールトトレランスで運用するには、サーバーをプライマリサーバーとバックアップサーバーのペアで構成します。プライマリサーバーはクライアントの接続を受け入れ、クライアントとやり取りしてメッセージを配信します。プライマリサーバーに障害が発生すると、代わってバックアップサーバーが処理を引継ぎます（三つ以上のサーバーを使用したフォールトトレランス構成はサポートされていません）。

### 共有ステート

フォールトトレランスでペアにされた各サーバーは、クライアント接続と永続的メッセージに関する情報で構成されている共有ステートにアクセスできます。この共有ステートの情報により、バックアップサーバーはクライアント接続とメッセージの処理に関する役割を適切に引き継ぐことができます。図 25 は、EMS のフォールトトレランス構成を示しています。

図 25 共有ステートを使用したプライマリサーバーとバックアップサーバー



### ロック処理

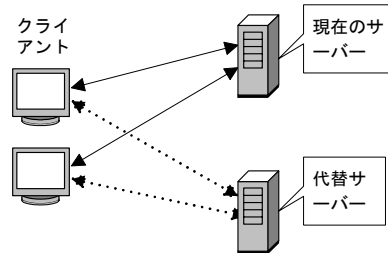
プライマリサーバーが正常に動作している間は、プライマリサーバーの役割がバックアップサーバーに引き継がれないよう、プライマリサーバーは共有ステートをロックします。障害の発生によってプライマリサーバーが停止すると、共有ステートのロックは解除され、バックアップサーバーがロックを取得します。

## 非共有ステートファイルオーバー

ステートを共有しないバックアップサーバーを含めることもできます。共有ステートと同様、現在のサーバーに障害が発生すると 2 番目のサーバーが接続とメッセージの処理に関する役割を引き継ぎます。ただし、共有ステートとは異なり非共有ステートは EMS クライアントによって制御され、非共有ステートフェイルオーバーは共有ステートフェイルオーバーほどフォールトトレラントではありません。ステートはサーバー間で共有されないため、フェイルオーバープロセスでメッセージの消失、重複、または不規則な配信が発生する可能性があります。

図 26 は、EMS の非共有ステートフォールトトレランス構成を示しています。

図 26 非共有ステートを使用した現在のサーバーと 2 番目のサーバー



## 設定ファイル

障害の発生によってプライマリサーバーが停止すると、バックアップサーバーはプライマリサーバーのステータスを引き継ぎ、処理を再開します。バックアップサーバーが新しいプライマリサーバーになる前に、バックアップサーバーはプライマリサーバーのすべての設定ファイルを読み込み直します。両方のサーバーが設定ファイルを共有している場合、古いプライマリサーバーのときに加えられた管理上の変更は、新しいプライマリサーバーに引き継がれます。



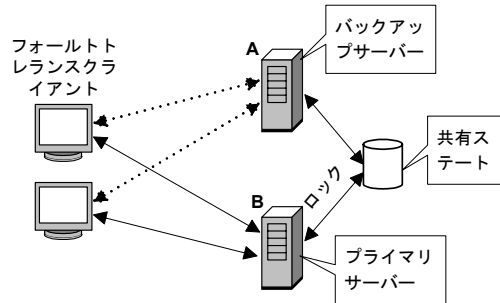
フォールトトレランスの各サーバーが設定ファイルを共有している場合、設定の変更は現在のプライマリサーバーのみに限定する必要があります。個別にバックアップサーバーの設定を変更すると、共有している設定ファイルを上書きするため、予期しない設定の誤りが起きる場合があります。



## 役割の交代

サーバー B が新しくプライマリサーバーになると、サーバー A はバックアップサーバーとして再起動できます。これにより両方のサーバーの役割が入れ替わります。

図 28 回復したサーバーのバックアップサーバーへの移行



## クライアントの移管

バックアップサーバー B へのフェイルオーバーが設定されているサーバー A のクライアントは、サーバー B が新しいプライマリサーバーになると、自動的にサーバー B に移管されます。サーバー B は共有ストレージからクライアントの現在のステートを読み取り、クライアントに永続的メッセージを配信します。

## クライアントへの通知

共有ステートフェイルオーバーの発生時に、クライアントアプリケーションが通知を受け取るように設定できます。

### Java

Java クライアントプログラムで通知を受け取るようにするには、システムプロパティ `tibco.tibjms.ft.switch.exception` に任意の値を設定し、`ExceptionListener` を定義してフェイルオーバーの通知を処理できるようにします。『TIBCO Enterprise Message Service Java API リファレンス』の `com.tibco.tibjms.Tibjms` クラスを参照してください。

### C

C クライアントで通知を受け取るようにするには、`tibems_setExceptionOnFTSwitch(TIBEMS_TRUE)` を呼び出し、例外コールバックを登録して、再接続に成功した通知を受け取ることができるようにします。

**C#**

.NET クライアントプログラムで通知を受け取るようにするには、`Tibems.SetExceptionOnFTSwitch(true)` を呼び出し、例外リスナーを定義して、フェイルオーバーの通知を処理できるようにします。『TIBCO Enterprise Message Service .NET API リファレンス』の 294 ページのメソッド `Tibems.SetExceptionOnFTSwitch` を参照してください。

**メッセージの再配信**

永続的	障害が発生すると、PERSISTENT の配信モードが設定されているメッセージのうち、障害発生の前に確認応答に成功していなかったメッセージは再配信されます。
同期モード	持続サブスクライバを使用すると、たとえ障害の発生中であっても、EMS は PERSISTENT 配信モードが設定されていて <code>store</code> にプロパティ <code>mode=sync</code> で書き込まれたメッセージが失われないことを保証します。
配信に成功済みのメッセージ	確認応答またはコミットに成功していたメッセージは再配信されません。これは、JMS 1.1 の仕様です。
トピック	すべてのトピックサブスクライバは、フェイルオーバーの後も正常に処理を継続します。

**トランザクション**

トランザクションが「アクティブ」になっているのは、ひとつ以上のメッセージがセッションによって送信または受信され、トランザクションのコミットが成功していない場合と考えられます。

フェイルオーバー後にアクティブなトランザクションにコミットを試みると、`javax.jms.TransactionRolledBackException` が発行されます。トランザクション処理を行うクライアントはこの例外を処理し、トランザクション中に送信されたメッセージを再送信する必要があります。バックアップサーバーは、ロールバックしたトランザクション中にセッションへ配信されたメッセージを自動的に再配信します。

**キュー**

キューレシーバの場合、レシーバに送信されたメッセージのうち、フェイルオーバーの前に確認応答されていなかったメッセージは、フェイルオーバーの直後に他のレシーバに送信されることがあります。

レシーバが、フェイルオーバー後にメッセージの確認応答を試みると、`javax.jms.IllegalStateException` を受け取ることがあります。この例外は、他のキューレシーバにすでに送信されているメッセージに対して確認応答が試行されたことを示します。この例外が発行されるのはこのシナリオの場合だけですが、

セッションの終了後や接続が切断しているときに確認応答が行われた場合にも発行されます。フェイルオーバー時にレシーバがひとつしか存在しない場合、この例外が発行されることはありませんが、排他キューに対して複数のレシーバが起動されていた場合、そのキューに対してこの例外が発行されることがあります。

キューレシーバが `javax.jms.IllegalStateException` 例外を受け取ったときの適切な対処方法は、`Session.recover()` メソッドを呼び出すことです。この場合、メッセージの再配信を処理できるよう、アプリケーションプログラムを設定しておく必要があります。フェイルオーバー後に他のキューレシーバに再配信可能なキューメッセージは、メッセージのヘッダフィールド `JMSRedelivered` が必ず `true` に設定されます。再配信のときに同じメッセージを重複して処理しないよう、アプリケーションプログラムは、このヘッダフィールドを必ずチェックしなければなりません。



確認応答されたメッセージは再配信されません (JMS 1.1 仕様に準拠)。上記のケースは、フェイルオーバーのためにアプリケーションがメッセージを確認応答できない場合です。

## ハートビートパラメータ

プライマリサーバーのハートビートが停止すると、バックアップサーバーは、アクティブ化間隔（最後のハートビートの検出以降の経過時間）のあいだ待機してから、共有ストレージからステート情報を取得し、プライマリサーバーの役割を引き継ぎます。

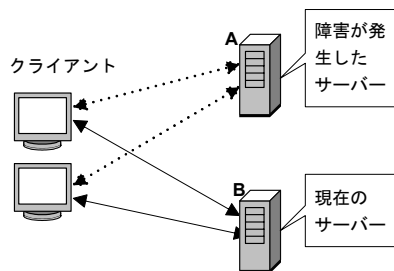
デフォルトのハートビート間隔は 3 秒で、デフォルトのアクティブ化間隔は 10 秒です。アクティブ化間隔は、ハートビート間隔の 2 倍以上にする必要があります。どちらの間隔も、秒単位で指定します。これらの間隔は、サーバーの設定ファイルで設定できます。詳細は、203 ページの「[フォールトトレランスのパラメータ](#)」を参照してください。

## 非共有ステートフェイルオーバーのプロセス

このセクションでは、非共有ステートフェイルオーバーのシーケンスについて説明します。設定情報についての詳細は、495 ページの「[非共有ステート接続用のクライアントの設定](#)」を参照してください。

**検出** 非共有ステートフェイルオーバーは、EMS クライアントによって開始されます。非共有ステートのクライアントセットアップでサーバー A への接続が失われたことが検出された場合、接続ファクトリで定義されたようにサーバー B への接続を試みます。

図 29 非共有ステートフェイルオーバー



**応答** 非共有ステート接続のクライアントは、A との接続が失われた場合は自動的に B に接続します。

非共有ステートのクライアントセットアップでサーバー A への接続が失われたことが検出された場合、サーバー B への新しい接続が作成されます。クライアントの接続によるすべてのランタイムオブジェクトは再作成されます。このオブジェクトにはセッション、デスティネーション、メッセージプロデューサ、およびメッセージコンシューマが含まれます。

非共有ステートは接続ファクトリで定義されるため、接続がアクティブである間は B が現在のサーバーとなります。B への接続が失われた場合、クライアントは接続ファクトリで定義された別のサーバーへの接続を試みます。

**メッセージの消失** B はフェイルオーバーの前に配信されていない永続的メッセージや確認応答されていない永続的メッセージにはアクセスできないため、フェイルオーバープロセスで一部のメッセージが消失したり、不規則に配信されたりする場合があります。メッセージの消失を回避するには、共有ステートフェイルオーバーを使用します。

**サポートされていない機能** 以下の機能と Java クラスは、非共有ステート接続ではサポートされていません。

- XA トランザクション
- トピックの持続サブスクライバ



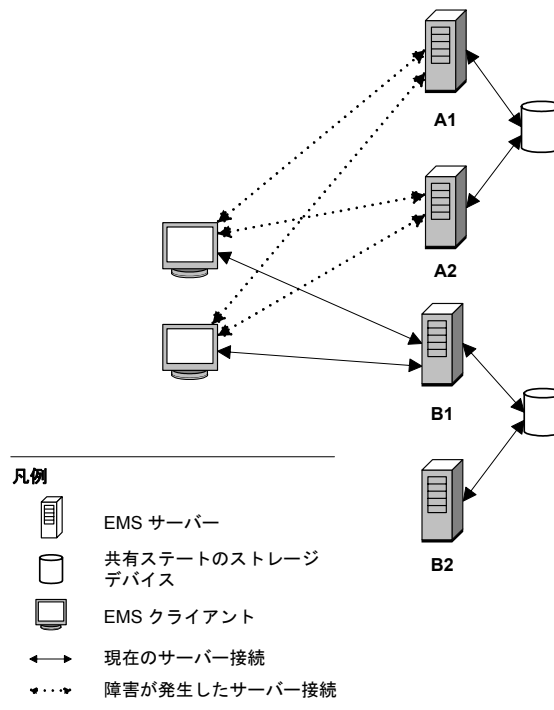
- ConnectionConsumer
- ServerSession
- ServerSessionPool
- QueueRequestor
- TopicRequestor

## デュアルステートフェイルオーバー

非共有ステート接続ファクトリには、バックアップサーバーのリスト内に共有ステートサーバーのペアを含めることができます。共有ステートサーバーと非共有ステートサーバーの両方が含まれている場合、フェイルオーバープロセスは両方のフェイルオーバータイプの組み合わせになります。

図 30 は、デュアルステートフェイルオーバーのプロセスを示しています。

図 30 デュアルステートフェイルオーバーのプロセス



この例では、サーバー A1 と A2 がステートを共有しています。また、サーバー B1 と B2 もステートを共有しています。ただし、A1 と A2 は、B1 と B2 とはステートを共有していません。

EMS クライアントは、非共有ステート接続ファクトリを使用して接続を作成しています。最初にサーバー A1 との接続を試みます。A1 への接続に失敗した場合、フェイルオーバープロセスは 476 ページの「[共有ステートフェイルオーバーのプロセス](#)」で説明されているように進行します。

A1 が再起動される前に、A2 で障害が発生するとします。この場合クライアントは B1 への接続を作成し、接続によるすべてのランタイムオブジェクトを再作成します（前述の[非共有ステートフェイルオーバーのプロセス](#)を参照）。これで B1 が現在のサーバーになります。B1 と B2 はステートを共有するため、B1 に障害が発生した場合は B2 が現在のサーバーになります。

## 共有ステート

最も堅牢なフェイルオーバー保護を実現するには、プライマリサーバーとバックアップサーバーが同じステートを共有する必要があります。サーバーの共有ステートには、次の三つのカテゴリの情報が含まれます。

- 永続的メッセージデータ（キューとトピック）
- プライマリサーバーのクライアント接続
- メッセージ配信に関するメタデータ

フェイルオーバー時に、バックアップサーバーはすべての共有ステート情報を再度読み取ります。

### 共有ステートの実装

共有ストレージデバイスを使用して共有ステートを実装することをお勧めします。共有ステートは、プライマリサーバーとバックアップサーバーの両方からアクセスできなければなりません。

### サポート条件

ハードウェアとソフトウェアの組み合わせによる共有ストレージの実装には、いくつかのオプションがあります。EMS のストレージソリューションには、[表 79](#) に示した四つの条件をすべて満たすことが要求されます。



共有ストレージに使用するストレージとオペレーティングシステムの各ベンダーに「必ず」相談し、採用したストレージソリューションがこの四つの条件を満たすようにしてください。

表 79 フォールトトレランスの共有ストレージ条件

条件	説明
書き込み順序	ストレージソリューションでは、共有ストレージに書き込まれるデータブロックの順番は、データバッファに入れられるデータの順番と一致する必要があります。  (たとえば、ディスクの使用効率を上げるために、データブロックの書き込み順序を変更しているソリューションは、この条件を満たしません。)
同期書き込みの永続性	同期書き込み呼び出しは、制御が戻った時点で、すべてのデータが持続性のある永続的なストレージに書き込まれている必要があります。

表 79 フォールトトレランスの共有ストレージ条件

条件	説明
分散ファイルロック	EMS の各サーバーは、共有ストレージを排他的にロックすることを要求し、それを取得する必要があります。ストレージソリューションでは、プライマリとバックアップのふたつのサーバーに同時にロックを割り当ててはいけません（485 ページの「ソフトウェアオプション」を参照）。  EMS の各サーバーは、このロックを手がかりに、どちらのサーバーがプライマリサーバーであるかを判断します。
一意の書き込み所有権	ファイルをロックできる EMS サーバープロセスはひとつだけであり、そのプロセスだけがファイルに書き込むことができます。システムがロックを他のサーバーに移管すると、以前の所有者によってキューに挿入された保留中の書き込みは失敗します。

ハードウェアオプション

一般的に入手可能な共有ストレージ用のハードウェアとして、次のタイプがあります。

- デュアルポート SCSI デバイス
- SAN（Storage Area Network）
- NAS（Network Attached Storage）

SCSI と SAN	デュアルポート SCSI および SAN によるソリューションは、書き込み順序と同期書き込みの永続性の条件を満たしています（残りのふたつの条件は、クラスタリングソフトウェアを使用して満足する必要があります）。四つの条件がすべて満たされていることを、必ずベンダーに確認してください。
NAS	NAS によるソリューションでは、分散ファイルロックの条件を満たすのに、CFS ではなく CS が必要とされます（以下を参照）。  NAS によっては、条件を満たさないソリューションがあります。四つの条件をすべて満たすことができるかどうか、必ずベンダーに確認してください。
NFS を採用した NAS	NFS ファイルシステムを採用している NAS の場合、そのソリューションが条件を満たしているかどうかを特定するのは極めて困難です。TIBCO で調査した結果、以下のことが言えます。 <ul style="list-style-type: none"><li>• NFS v2 と NFS v3 は、条件をまったく満たしていません。</li><li>• NFS v4 と TCP の組み合わせは、条件を満たす場合があります。NAS のベンダーに相談し、NAS 内の NAS サーバーが条件を満たすことを確認してください。オペレーティングシステムのベンダーに相談し、（サーバーホストコンピュータのオペレーティングシステム内の）NFS クライアントが条件を満たすことを確認してください。両方のベンダーによって、各コンポーネントが連携動作し、条件を満たすことが保証されれば、その共有ストレージソリューションが EMS をサポートすることになります。</li></ul>

EMS による共有ストアファイルのロック方法についての詳細は、114 ページの「[EMS による共有ストアファイルへのアクセスの管理方法](#)」を参照してください。

## ソフトウェアオプション

一般的に入手可能なソフトウェアとして、次のタイプがあります。

- クラスタサーバー (CS)

クラスタサーバーは、EMS サーバーのプロセスとホストコンピュータを監視し、常に、ひとつのサーバープロセスが正常に動作するようにします。プライマリサーバーが停止した場合、プライマリサーバーは CS サーバーによって再起動され、再起動に失敗した場合は、代わりにバックアップサーバーが起動されます。

- クラスタ化ファイルシステム (CFS)

CFS では、ふたつの EMS サーバーのプロセスを同時に処理できます。両方のサーバーに共有ファイルシステムを同時にマウントできます。ただし、CFS は一度にひとつのサーバープロセスにしかロックを割り当てることができません。また、CFS はオペレーティングシステムのファイルデータのキャッシュ処理を管理しているので、バックアップサーバーから、ファイルシステムの最新の状態を参照でき、古いキャッシュデータが参照されることはありません。

デュアルポート SCSI または SAN ハードウェアの場合は、CS または CFS のいずれも [分散ファイルロック](#) の条件を満たします。NAS ハードウェアの場合、この条件を満たすのは CS だけです (通常、CFS ソフトウェアは条件を満たしません)。もちろん、四つの条件がすべて満たされることを、必ずベンダーに確認する必要があります。

## 共有ステートに保存されたメッセージ

PERSISTENT の配信モードが設定されているメッセージは保存されるため、プライマリサーバーに障害が発生した場合に、それらのメッセージを取得できます。NON\_PERSISTENT 配信モードが設定されているメッセージは、プライマリサーバーに障害が発生すると、取得できません。

フェイルオーバー時のメッセージの回復についての詳細は、478 ページの「[メッセージの再配信](#)」を参照してください。

## ストレージファイル

デフォルトでは、tibemspd サーバーは共有ステートを保存するために三つのファイルベースのストアを作成します。

- `$sys.failsafe`: このストアは、同期 I/O 呼び出しを使用して永続的メッセージを保持します。
- `$sys.nonfailsafe`: このファイルは、非同期 I/O 呼び出しを使用してメッセージを保存します。

- `$sys.meta` : このストアは、持続サブスクライバ、フォールトトレランス接続、およびその他のメタデータに関するステート情報を保持します。

これらのストアは、ストア設定ファイルのパラメータを使用して完全にカスタマイズできます。これらのファイルとデフォルトの設定についての詳細は、[243 ページの「stores.conf」](#)を参照してください。

ふたつのサーバーが同じストアファイルを使用しないように、どちらのサーバーも、処理中のストアファイルへのアクセスを制限します。EMS による共有ストアファイルの管理方法についての詳細は、[114 ページの「EMS による共有ストアファイルへのアクセスの管理方法」](#)を参照してください。



これらのデフォルトのファイルは変更または修正可能です。詳細は、[30 ページの「デフォルトのストアファイル」](#)を参照してください。

## ストレージパラメータ

フォールトトレランス機能を設定しない場合でも、EMS ストレージファイルに適用される設定パラメータがいくつかあります ([196 ページの「ストレージファイルのパラメータ」](#)を参照)。

## フォールトトレランスサーバーの設定

### 共有ステート

EMS サーバーをフォールトトレランスのバックアップサーバーに設定する場合、主要設定ファイル（またはサーバーのコマンドライン）で、以下のパラメータを設定します。

- **server**     プライマリサーバーとバックアップサーバーの両方の設定ファイルに同じサーバー名を指定します。
- **ft\_active**     プライマリサーバーの設定ファイルで、バックアップサーバーの URL を設定します。バックアップサーバーの設定ファイルに、プライマリサーバーの URL を設定します。

バックアップサーバーは起動時に、プライマリサーバーへの接続を試みます。プライマリサーバーへの接続を確立すると、バックアップサーバーはスタンバイモードに入ります。プライマリサーバーへの接続が確立できない場合、バックアップサーバーが（アクティブモードに入って）プライマリサーバーの役割を引き継ぎます。

スタンバイモードのバックアップサーバーは、クライアントからの接続要求を受け付けません。バックアップサーバーを管理する場合、**admin** ユーザーは EMS 管理ツールを使って、バックアップサーバーに接続できます。

### 認可とフォールトトレランスサーバー

EMS の認可処理は、フォールトトレランスな方法でやり取りします。

**authorization** を有効にし、ふたつの EMS サーバーをフォールトトレランスに設定する場合、フォールトトレランスペアの両方のサーバーを以下のように設定する必要があります。

- 各サーバーの **tibemsd.conf** ファイルに同じサーバー名とパスワードを設定する必要があります（**server** と **password** のパラメータの値は両サーバーとも同じにします）。
- 各サーバーの **users.conf** ファイル内のユーザー名とパスワードは、**tibemsd.conf** ファイル内の **server** と **password** の各パラメータの値と一致する必要があります。



ふたつの EMS サーバーが、**users.conf** ファイルを共有しない場合は、EMS サーバーと同じ名前のユーザーを作成し、そのユーザーのパスワードの値に、「サーバー」のパスワードを設定します。

たとえば、ふたつの EMS サーバー（サーバー 1 とサーバー 2）の名前が「EMS-SERVER」で、使用するパスワードが「mySecret」になっているとします。また、これらのサーバーは **users.conf** ファイルを共有していないとします。ユーザー名とパスワードを設定するには、117 ページの「[EMS 管理ツールの使用](#)」の説明に従って、サーバーごとに EMS 管理ツールを起動し、以下の設定を行います。

アクティブサーバー（サーバー 1）から、次のように入力します。

```
set server password=mySecret
create user EMS-SERVER password=mySecret
```

バックアップサーバー（サーバー 2）から、次のように入力します。

```
set server password=mySecret
create user EMS-SERVER password=mySecret
```

アクティブサーバー（サーバー 1）から、次のように入力します。

```
set server authorization=enabled
```

バックアップサーバー（サーバー 2）から、次のように入力します。

```
set server authorization=enabled
```

## SSL

SSL を使用すると、フォールトトレランスペアのサーバー間の通信をセキュリティ保護できます。

主要設定ファイル ([tibemsd.conf](#)) のパラメータは、この動作に影響を与えます。関連するパラメータには、すべて先頭に `ft_ssl` のプリフィクスが付いています。

別のサーバーへのセキュアな接続を初期化するサーバーは、`ft_ssl` パラメータを使用してその別のサーバーへのセキュアな接続のプロパティを決定します。受信側のサーバーは、その受信接続を独自の `ssl_` パラメータに対して検証します。[ft\\_ssl](#) パラメータについての詳細は、203 ページの「[フォールトトレランスのパラメータ](#)」を参照してください。`ssl_` パラメータについての詳細は、214 ページの「[SSL サーバーのパラメータ](#)」を参照してください。

関連トピック    453 ページの [第 18 章、「SSL プロトコルの使用」](#)

## 再接続のタイムアウト

フェイルオーバー時にバックアップサーバーがプライマリサーバーの役割を引き継ぐと、クライアントはバックアップサーバー（新しいプライマリサーバー）に接続し直し、現在のメッセージステートの処理を続行します。クライアントからの再接続要求を受け付ける前に、バックアップサーバーは共有ステートファイルからメッセージステートを読み込みます。

`ft_reconnect_timeout` 設定パラメータで指定された時間内に再接続できないクライアントについては、ステート情報をクリーンアップするようにサーバーに指定できます。`ft_reconnect_timeout` 時間は、サーバーが共有ステートを完全に復元したときに開始します。従って、この値には、ストアファイルを復元するのに必要な時間は含まれていません。詳細は、[204 ページの「ft\\_reconnect\\_timeout」](#)を参照してください。



## 非共有ステート

ステートを共有しないフォールトトレランスのペアを設定する場合、両方のサーバーが同じ設定を使用していることを確認する必要があります。これは、以下の設定で特に重要です。

- **デスティネーション**：両方のサーバーが同じデスティネーションをサポートしている必要があります。
- **ルート**：メッセージは、サーバー間で同等または同一のルートを使用してエンドポイントに到達できる必要があります。
- **アクセス制御**：`users.conf`、`groups.conf`、および `acl.conf` ファイルの設定が一致するように、アクセス制御が両方のサーバーで完全に等しくセットアップされている必要があります。
- **SSL**：SSL が導入されている場合、両方のサーバーが同じ証明書を使用している必要があります。

## Central Administration でのフォールトトレランスの設定

Central Administration では、フォールトトレランスでペアにされた両方のサーバーの管理に同じ JSON 設定ファイルを使用します。プライマリサーバーの役割とセカンダリサーバーの役割は、サーバーの起動時に決定します。

`listen` パラメータと `ft_active` パラメータは個別に設定されますが、このふたつ以外のすべての設定は両方の EMS サーバーで共有されます。

- プライマリサーバーは、メインの [Server Properties] (サーバーのプロパティ) ページにある [Primary Listens] (プライマリリスン) セクションで定義されたポートでクライアント接続をリスンします。フェイルオーバー後、[Fault Tolerance] (フォールトトレランス) プロパティページの [FT Active] (FT アクティブ) ラジオボタンを使用してフラグが付けられた、[Secondary Listens URL] (セカンダリリスン URL) でセカンダリサーバーをリスンします。
- セカンダリサーバーは、メインの [Server Properties] (サーバーのプロパティ) ページにある [FT Active] (FT アクティブ) ラジオボタンでフラグが付けられた、[Primary Listens URL] (プライマリリスン URL) を使用してプライマリサーバーをリスンします。セカンダリサーバーがアクティブになると、セカンダリサーバーは [Fault Tolerance] (フォールトトレランス) ページで定義された [Secondary Listen URLs] (セカンダリリスン URL) を使用してクライアント接続をリスンします。

Central Administration についての詳細は、TIBCO Enterprise Message Service Central Administration を参照してください。

**手順** Central Administration を使用してフォールトトレランスサーバーのペアを設定するには、以下の手順に従います。

1. プライマリサーバーを通常どおり設定します。
2. [Server Properties] (サーバーのプロパティ) ページで必要なリスン URL の横にある [FT Active] (FT アクティブ) ラジオボタンをクリックし、セカンダリサーバーがプライマリサーバーをリスンする URL を指定します。
3. [Fault Tolerance] (フォールトトレランス) プロパティページで、バックアップサーバーがプライマリサーバーになった場合にクライアント接続をリスンするために使用する、[Secondary Listens URLs] (セカンダリリスン URL) を設定します。
4. 障害が発生してセカンダリサーバーがアクティブになった場合に、プライマリサーバーがセカンダリサーバーをリスンする URL を指定します。必要なセカンダリリスン URL の横にある [FT Active] (FT アクティブ) ラジオボタンをクリックします。
5. [Fault Tolerance] (フォールトトレランス) ページで、フォールトトレランスの残りのプロパティを設定します。
6. 設定の変更を適用します。

7. 103 ページの「[フォールトトレランスサーバーのペアの起動](#)」で説明されている方法を使用して、プライマリおよびセカンダリ EMS サーバーを起動します。

**設定エラー** EMS サーバーの起動時に、プライマリ `tibemsd` のセカンダリリスンリスト内にある URL またはセカンダリ `tibemsd` のプライマリリスンリスト内にある URL によって、フォールトトレランスメカニズムがトリガーされます。

フォールトトレランスがトリガーされると、EMS サーバーで [FT Active] (FT アクティブ) スイッチがピアのリスト内の URL に割り当てられていないことが検出された場合にエラーが生成されます。`startup_abort_list` パラメータ内に `CONFIG_ERRORS` が存在する場合、`tibemsd` は起動を中止します。存在しない場合、`tibemsd` はフォールトトレランスをキャンセルし、ピアをチェックせずに起動します。その結果、2 番目に起動された EMS サーバーでファイルロックエラーが発生します。

## フォールトトレランスサーバー用のクライアントの設定

フェイルオーバー時にバックアップサーバーがプライマリサーバーの役割を引き継ぐと、クライアントはバックアップサーバー（新しいプライマリサーバー）に接続し直します。クライアントが接続を変更できるようにするには、接続の作成時に両方のサーバーの URL を指定する必要があります。

複数のサーバーを指定する場合は、各 URL をコンマで区切って指定します。どちらの URL にも同じ通信プロトコル（tcp または ssl）を指定する必要があります。たとえば、最初のサーバー（tcp://server0:7222）と（最初のサーバーが利用できない場合の）2 番目のサーバー（tcp://server1:7344）を指定するには、次のように入力します。

```
serverUrl=tcp://server0:7222, tcp://server1:7344
```

クライアントは、入力した URL の順番でサーバーに接続を試みます。最初の URL への接続が失敗すると、次の URL への接続が試行されます。すべての URL への試行が終了するまで、クライアントは順番に URL への接続を試みます。最初に失敗した接続がリストの最初の URL でなかった場合、接続の試行はリストの先頭へ戻ります（これによって、すべての URL が試行されます）。どの URL への試行も成功しなかった場合、接続は失敗です。

EMS 名前付けサービスでフォールトトレランスの URL を照合する方法についての詳細は、354 ページの「[フォールトトレランス照合の実行](#)」を参照してください。



クライアントの再接続ロジックは、サーバーへの接続時に複数の URL が指定されることによってトリガされます。バックアップサーバーが存在しない場合、障害の発生後に利用可能になったサーバーに自動的にクライアントが接続されるようにするには、クライアントは少なくともふたつの URL を指定する（通常は、同じサーバーを参照する）必要があります。



メッセージが非永続モードまたは信頼モードで送信された場合、通常のコンシューマはサーバーによる確認応答を待機しません。ただし、フォールトトレランスコンシューマはサーバーの応答を待機します（DUPS\_OK\_ACKNOWLEDGE または EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE 以外の確認応答モードを使用している場合）。これは共有ステート設定の場合に該当します。非共有ステート設定ではフェイルオーバー時にメッセージの消失、重複、および不規則な送信が許容されるため、サーバーの応答を待機しません。

### 三つ以上の URL の指定

サーバーには、プライマリとバックアップのふたつしかありませんが、クライアントは三つ以上の接続 URL を指定することができます。たとえば、各サーバーに複数のリスンアドレスが設定されている場合、クライアントは、同じサーバーに接続し直す場合に、別のアドレス（つまり、異なるネットワークインタフェース）を使用できます。

## 再接続障害パラメータの設定

EMS では、初回の接続試行と再接続の試行とに別々のパラメータを設定できます。初回の接続試行のパラメータを設定する方法については、324 ページの「[接続の試行回数、タイムアウト、および遅延時間のパラメータ設定](#)」を参照してください。このセクションでは、フォールトトレランスによるスイッチオーバーの後に再接続が行われるよう、設定できるパラメータを説明します。

初回の接続試行パラメータと再接続の試行パラメータを分けている理由は、オペレーティングシステムによって、特定の時間に EMS サーバーが処理できる接続試行の回数が制限されているからです（たとえば Unix の場合、この制限は `ulimit` の設定で調整できます）。通常の場合、各接続試行は分散されるため、サーバーが受け取る接続要求はキューの最大値を超えることはありません。しかし、フォールトトレランス スwitchオーバー時には、各クライアントは自動的に、ほぼ一斉にバックアップサーバーへの再接続を試みます。接続数が多い場合、各クライアントが接続し直すのにかかる時間は、初回の接続にかかる時間よりも長くなる場合があります。

デフォルトでは、クライアントは 500 ミリ秒の間隔（遅延時間）でサーバーへの再接続を 4 回試みるように設定されています。この設定を変更するには、[factories.conf](#) ファイルまたはクライアントの接続ファクトリ API を使用して、以下の例のように設定します。

以下の各コード例は、試行回数 10、遅延時間 1000 ミリ秒、タイムアウト 1000 ミリ秒の再接続の設定例です。

### Java

`TibjmsConnectionFactory` オブジェクトの `setReconnAttemptCount()`、`setReconnAttemptDelay()`、および `setReconnAttemptTimeout()` メソッドを使って、新しい再接続障害パラメータを設定します。

```
factory.setReconnAttemptCount(10);
factory.setReconnAttemptDelay(1000);
factory.setReconnAttemptTimeout(1000);
```

### C

`tibemsConnectionFactory_SetReconnectAttemptCount`、`tibemsConnectionFactory_SetReconnectAttemptDelay`、および `tibemsConnectionFactory_SetReconnectAttemptTimeout` 関数を使って、新しい再接続障害パラメータを設定します。

```
status = tibemsConnectionFactory_SetReconnectAttemptCount(
    factory, 10);

status = tibemsConnectionFactory_SetReconnectAttemptDelay(
    factory, 1000);

status = tibemsConnectionFactory_SetReconnectAttemptTimeout(
```

```
factory, 1000);
```

## C#

ConnectionFactory.SetReconnAttemptCount、  
ConnectionFactory.SetReconnAttemptDelay、および  
ConnectionFactory.SetReconnAttemptTimeout メソッドを使って、新しい再  
接続障害パラメータを設定します。

```
factory.setReconnAttemptCount(10);  
factory.setReconnAttemptDelay(1000);  
factory.setReconnAttemptTimeout(1000);
```

## 非共有ステート接続用のクライアントの設定



非共有ステートフェイルオーバーは、JMS 仕様の拡張機能です。ステートはサーバー間で共有されないため、フェイルオーバープロセスでメッセージの消失、重複、または不規則な配信が発生する可能性があります。

非共有ステート接続の作成には、共有ステート接続とは異なるいくつかの重要な点があります。

- Java アプリケーションの場合、JAR ファイルがクライアントの環境 CLASSPATH 内に存在する必要があります。
- C アプリケーションの場合、ヘッダファイルがインクルードされている必要があります、クライアントは非共有ステートライブラリを使用してリンクする必要があります。
- 接続は、非共有ステートの接続ファクトリを使用して作成する必要があります。
- サーバー URL は、非共有ステートの構文を使用して指定する必要があります。

### 非共有ステートライブラリのインクルード

Java アプリケーション	接続ファクトリを作成する前に、CLASSPATH に JAR ファイルがインクルードされていることを確認します。 <code>tibjmsufo.jar</code>
C アプリケーション	<code>tibemsufo.h</code> ヘッダファイルをインクルードします。

### 非共有ステート接続ファクトリの作成

非共有ステート接続を作成するには、関連メソッドを使用します。

Java アプリケーション	<code>java com.tibco.tibems.ufo</code> パッケージ
C アプリケーション	<code>tibemsufo</code> ライブラリおよび関数

#### 接続の回復

非共有ステート接続に失敗した場合、接続の `ExceptionListener` コールバックが呼び出されます。接続を回復（アクティブなサーバーに接続されるように修復）するには、クライアントで接続ファクトリの `recoverConnection` メソッドまたは `tibemsUFOConnectionFactory_RecoverConnection` 関数を呼び出します。

`ExceptionListener` コールバックでこれが実行される必要があります。接続の回復メソッドは、接続（およびセッション、プロデューサ、コンシューマを含む関連オブジェクト）が完全に再作成されるか、これらのオブジェクトの再作成がすべて失敗するまでブロックします。

非共有ステートのクライアントに有効な接続がある限り、API は標準の EMS クライアントと同様に動作します。ただし、非共有ステートクライアントの接続が失われた場合、API は以下のように動作します。

1. `MessageListener` コールバック内で呼び出されたメソッドは、Java の例外 `ConnectionFactoryException` または C の `TIBEMS_SERVER_NOT_CONNECTED` ステータスを即座に返します。
2. その他の場所で呼び出されたメソッドは、再度接続が有効になるまでブロックします。

基になる TCP/SSL 接続に失敗した時点から `recoverConnection` または `tibemsConnectionFactory.RecoverConnection` が正常に返されるまで、接続が失われているとみなされます。

## サーバーの URL の指定

非共有ステートのフェイルオーバー中にサーバーの接続が失われた場合、クライアントは 2 番目のサーバーへの再接続を試みます。クライアントが接続を変更できるようにするには、接続の作成時に両方のサーバーの URL を指定する必要があります。

- **非共有ステート** 複数のサーバーを指定する場合は、各 URL をプラス記号 (+) で区切って指定します。たとえば、最初のサーバー (`tcp://server0:7222`) と 2 番目のサーバー (`tcp://server1:7344`) を指定するには、次のように入力します。

```
serverUrl=tcp://server0:7222+tcp://server1:7344
```

- **デュアルステート** 共有ステートサーバーのペアを非共有ステートサーバーと組み合わせるには、ステートを共有するサーバーはコンマで区切り、ステートを共有しないサーバーはプラス記号 (+) で区切ります。たとえば、次の指定では、サーバー a1 と a2 はステートを共有するフォールトトレランスのペアで、サーバー b1 と b2 はステートを共有する 2 番目のペアです。

```
serverUrl=tcp://a1:8222,tcp://a2:8222+tcp://b1:8222,tcp://b2:8222
```

a1 と a2 は、b1 と b2 とはステートを共有しません。

クライアントは、入力した URL の順番でサーバーに接続を試みます。最初の URL への接続が失敗すると、次の URL への接続が試行されます。すべての URL への試行が終了するまで、クライアントは順番に URL への接続を試みます。最初に失敗した接続がリストの最初の URL でなかった場合、接続の試行はリストの先頭へ戻ります（これによって、すべての URL が試行されます）。どの URL への試行も成功しなかった場合、接続は失敗です。



サーバーの照合関数では、非共有ステートの構文は使用できません。つまり、サーバーの照合時にサーバーの URL をプラス記号 (+) で区切ることはできません。



## 第 20 章

# ルートの使用

この章では、TIBCO Enterprise Message Service サーバー間におけるメッセージのルーティングについて説明します。

### トピック

---

- [「ルーティングの概要」、498 ページ](#)
- [「ルート」、499 ページ](#)
- [「ゾーン」、502 ページ](#)
- [「アクティブルートとパッシブルート」、505 ページ](#)
- [「ルートとゾーンの設定」、506 ページ](#)
- [「トピックメッセージのルーティング」、511 ページ](#)
- [「ルーティング先キュー」、516 ページ](#)
- [「ルーティングと認可」、519 ページ](#)

## ルーティングの概要

---

TIBCO Enterprise Message Service サーバーは、別のサーバーにメッセージをルーティングすることができます。

- トピックメッセージは、最初のサーバーから 1 ホップまたは複数のホップを移動できます。
- キューメッセージは、ホームキューに向かって 1 ホップ、およびホームキューから 1 ホップだけ移動できます。

管理インタフェース（設定ファイル `tibemsadmin` または管理 API）を使用して、ルートを定義することができます。

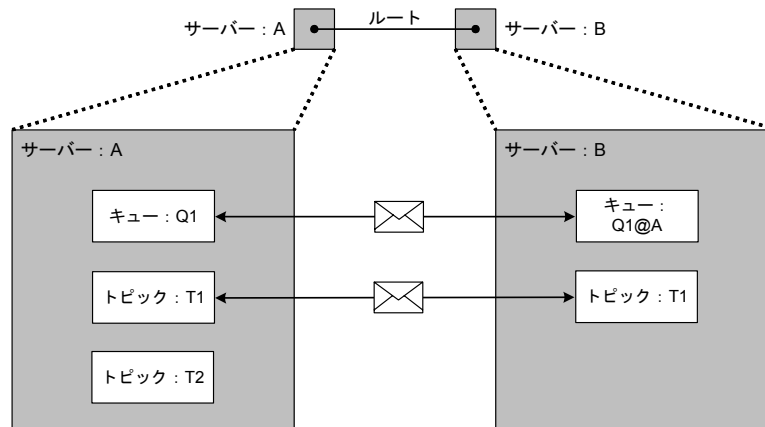
## ルート

### 基本操作

- 各ルートは、ふたつのTIBCO Enterprise Message Serviceサーバーを接続します。
- 各ルートは、ふたつのサーバー上の対応するデスティネーション（同じ名前を持つグローバルトピック、または明示的なルーティング先キュー）間で、メッセージを転送します。
- ルートは双方向であり、各ペアのサーバーは、互いに他方のサーバーへのルートに沿ってメッセージを転送します。

たとえば、図 31 の上部の簡略図は、ふたつのサーバー A と B をつなぐルートを示しています。その下の展開図は、ルートの動作を示しています。それぞれのサーバーには、グローバルトピック T1 と、ルーティング先キュー Q1 があります。これらのデスティネーションは対応しているため、ルートは両者間でメッセージを転送します。さらに、サーバー A には、サーバー B のいずれのトピックにも対応しないグローバルトピック T2 があります。ルートは、T2 からのメッセージを転送しません。

図 31 ルート：双方向に通信し合うデスティネーション

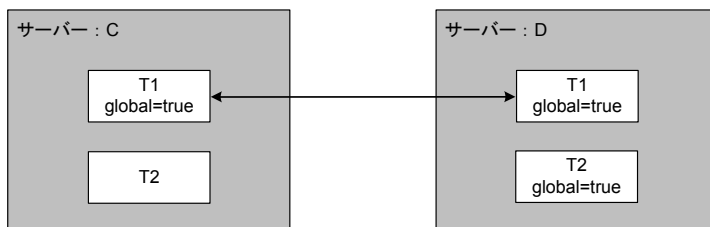


## グローバルデスティネーション

ルートは、グローバルデスティネーション間でのみメッセージを転送します。つまり、トピックの場合、**global** プロパティが両方のサーバーで設定されている必要があります（キューについては、516 ページの「[ルーティング先キュー](#)」を参照）。デスティネーションプロパティについての詳細は、55 ページの「[デスティネーションのプロパティ](#)」を参照してください。

図 32 は、対応するデスティネーション T1 と T2 を持つ、ふたつのサーバー C と D 間のルートを示しています。T1 はサーバー C と D のそれぞれにおいてグローバルであるため、両サーバーはルートを介して、対応するデスティネーションにメッセージを転送できます。一方、サーバー C にある T2 はグローバルではないため、サーバー C と D のどちらも T2 メッセージを相手に転送しません。

図 32 ルート：グローバルデスティネーション



## 一意のルーティングパス

複数のパスを経由してメッセージがサーバーに到達可能なルートのセットを定義することはできません。TIBCO Enterprise Message Service サーバーは、不正な重複ルートを検出すると、設定エラーとしてレポートします。

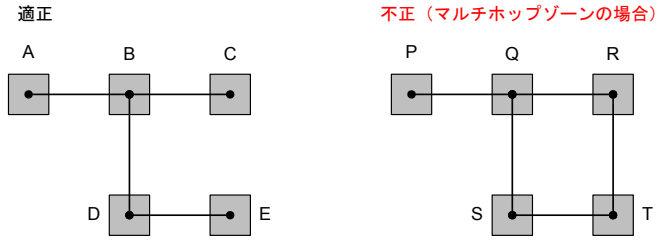
501 ページの「[図 33](#)」は、2 組のルートを示しています。左側の図では、サーバー A、B、C、D、および E を結ぶルートは閉路を形成していません。サーバーのペア間を接続しているのはひとつのルートだけです。この設定は、すべてのゾーンにおいて有効です。

一方、右側の図のようなルーティング設定は、マルチホップゾーンでは不正です。この図には、サーバー Q と S を結ぶ冗長なルーティングパスが含まれています（一方はダイレクトパス、もうひとつは R と T を経由するパス）。



ただし、右側の設定はマルチホップゾーンでのみ不正となりますが、1 ホップゾーンでは適正です。詳細は、502 ページの「[ゾーン](#)」を参照してください。

図 33 ルート：一意のパス



# ゾーン

ゾーンを使ってルートの動作を制限することにより、複雑なルーティングパスを設定できます。ゾーンが適用されるのはトピックメッセージだけで、キューメッセージには適用されません。

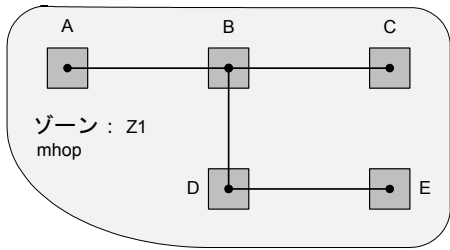
## 基本操作

「ゾーン」とは、ルートのセットのことです。すべてのルートは、ゾーンに属しています。ゾーンは、以下のようにルートの転送動作に影響を与えます。

- マルチホップ (mhop) ゾーンでは、トピックメッセージは、ルートによって接続されたゾーン内のすべてのサーバーを、すべての適正なルートに沿って移動できます。
- 1 ホップ (1hop) ゾーンでは、トピックメッセージは、最初のサーバーから 1 ホップのみ移動できます。
- キューメッセージが移動できるのは、マルチホップゾーンにおいても 1 ホップのみです。

たとえば、[図 34](#) は、マルチホップゾーン Z1 内の、ルートによって接続されたサーバーのセットを示しています。クライアントがメッセージをサーバー B のグローバルトピックに送信した場合、サーバー B がメッセージをサーバー A、C、D、E に転送します（各サーバーにサブスクライバが存在することを前提にしています）。一方、Z1 が 1 ホップゾーンである場合、サーバー B はメッセージを A、C、D に転送しますが、サーバー D はそのメッセージを E に転送しません。

図 34 ゾーン：マルチホップ



## 1 ホップゾーンの冗長パスの削除

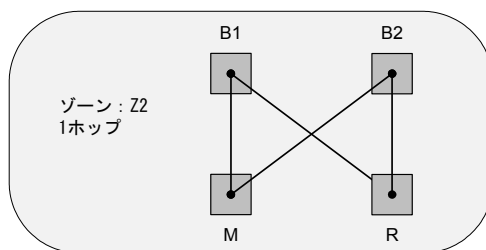
[図 35](#) は、次の四つのサーバーを持つ企業を示しています。

- 支社のサーバー B1 と B2 は、プロデューサとして機能します。
- 本社のサーバー M は、コンシューマとして機能し、支社からのメッセージを処理します。

- サーバー R は、コンシューマとして機能し、アーカイブ、監視、およびバックアップのためにメッセージを記録します。

ここでの目的は、サーバー B1 と B2 からのメッセージをサーバー M と R の両方に転送することです。ルーティング図には、ひとつの閉路が含まれるように見えます。たとえば、サーバー B1 から M と B2 を通って R に至るルートは、B1 から R へのルートと重複しています。ただし、これらのルートは 1 ホップゾーンの Z2 に属しているため、メッセージが長い方のルートを通ることはありません。この制限により、メッセージは、B1 から M と R に、また B2 から M と R に、というように最適なルートで転送されます。

図 35 ゾーン : 1 ホップ



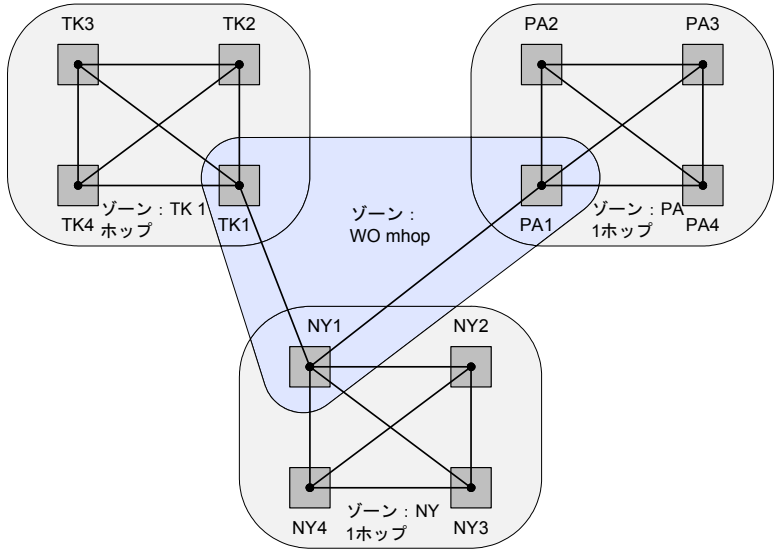
## 複数のゾーンのオーバーラップ

サーバーは、複数のゾーンに属するルートを持つことができます。ゾーンがあるサーバーでオーバーラップしている場合、各ゾーン内でのルーティング動作は、別のゾーンにおけるルーティングを制限しません。転送されたメッセージが、複数のゾーン内のルートを経由してサーバーに到達する場合、メッセージはゾーンの境界を越えることになり、そのたびにホップカウントがゼロにリセットされるからです。

504 ページの「図 36」の接続図は、ある都市内のすべてのサーバーを完全接続グラフの形で接続している 1 ホップゾーンを、複数の都市について持つ企業を示しています。ゾーン TK は東京にあるすべてのサーバーを接続し、ゾーン NY はニューヨークにあるすべてのサーバーを接続し、ゾーン PA はパリにあるすべてのサーバーを接続しています。さらに、マルチホップゾーン WO は、各都市からサーバーをひとつずつ接続しています。

サーバー TK3 のクライアントがメッセージをプロデュースすると、そのメッセージは東京の各サーバーを 1 ホップずつ移動します。メッセージが TK1 に到達すると、ゾーン WO に入ります。TK1 は、メッセージを NY1 に転送し、NY1 はそのメッセージを PA1 に転送します。メッセージは NY1 に到達すると、ゾーン NY に入ります (このとき、ホップ数がゼロにリセットされる)。NY1 は、このメッセージをニューヨークの他の各サーバーに 1 ホップずつ転送します。同様に、メッセージは PA1 に到達すると、ゾーン PA に入ります (このとき、ホップ数がゼロにリセットされる)。PA1 は、このメッセージをパリの他の各サーバーに 1 ホップずつ転送します。

図 36 ゾーン：オーバーラップ





## アクティブルートとパッシブルート

ルートは、ふたつのサーバーを接続します。ルートは、一方のサーバーだけで設定することも、両方のサーバーで設定することもできます。

### アクティブ - パッシブルート

一方のサーバーでみ、ルートを設定した場合、双方のサーバーから見たルートは、以下のように異なります。

- ルート設定を行ったサーバーから見た場合、ルートは「アクティブ」です。このサーバーは、別のサーバーへの接続を能動的に開始するため、ここではこのサーバーを「アクティブサーバー」または「開始サーバー」と呼びます。
- 他方のサーバーから見ると、ルートは「パッシブ」です。このサーバーは、アクティブサーバーからの接続要求を受動的に受け取るため、ここではこのサーバーを「パッシブサーバー」と呼びます。

ひとつのサーバーに、アクティブルートとパッシブルートの両方を設定することができます。つまり、サーバー S からルートを開始するように設定すると同時に、他方のサーバーから S へのルートを開始するように設定することができます。

アクティブルートのプロパティは指定および修正することができますが、パッシブルートのプロパティは指定も修正もできません。これは、ルートのプロパティは、ルートが設定された（接続を開始した）サーバーに関連付けられているからです。



ルートを定義することで、明示的または暗黙的にゾーンも指定されます。あるゾーンの最初のルートはルートのタイプを定義します。同じゾーンの以降のルートは、同じゾーンタイプにする必要があります（タイプが違うと、サーバーはエラーをレポートします）。

### アクティブ - アクティブルート

ふたつのサーバー間で、一方から他方へのルートを両方ともアクティブルートとして設定できます。この配置を、「アクティブ - アクティブ設定」と呼びます。たとえば、サーバー A からサーバー B へのルートを指定すると同時に、B から A へのルートを指定するとします。どちらのサーバーも接続の開始を試行することができます。この構成でされる接続は、ひとつだけなので、冗長ルートが作成されることはありません。

「アクティブ - パッシブ」ルートを、「アクティブ - アクティブ」ルートに昇格させることもできます。ルートを昇格させるには、パッシブサーバー上で、次のコマンドを使用します。

```
create route name url=url
```

パッシブサーバー（ルートの昇格元）でルートが切断された場合に他方のサーバーに接続できるように、引数 `url` は必ず指定してください。詳細は、[126 ページの「create route」](#)も参照してください。

昇格したルートは、静的に設定されたルートとして動作します。つまり、持続サブスクライバのメッセージを存続させます。また、管理者がそのルートのプロパティを変更できるように、そのルートの設定を `routes.conf` に保存します。

## ルートとゾーンの設定

管理ツール（117 ページの「[第 6 章](#)」を参照）、または管理 API（オンラインマニュアルの「`com.tibco.tibjms.admin.RouteInfo`」を参照）を使用して、ルートを作成することができます。

**構文** 管理ツールを使用してルートを作成するには、一方のサーバーに接続してから、次の構文で `create route` コマンドを実行します。

```
create route name url=URL zone_name=zone_name zone_type=1hop|mhop properties
```

- *name* には、サーバーの名前（ルートのもう一方の端）を指定します。これがルートの名前になります。
- *URL* には、プロトコルとポートを含む、その他のサーバーの URL を指定します。  
フォールトトレランス環境として設定されている場合、URL には、各フォールトトレランスサーバーの URL をコンマ区切りで列挙して指定することもできます。フォールトトレランスについての詳細は、473 ページの[第 19 章、「フォールトトレランス」](#)を参照してください。
- *zone\_name* には、ルートが属するルーティングゾーンの名前を指定します。指定しなかった場合、デフォルト値は `default_mhop_zone` となる（このデフォルト値は、4.0 より前のリリースの設定ファイルに下位互換性を提供する）。
- ゾーンのタイプ。1hop または mhop を指定します。指定しなかった場合、デフォルト値は mhop です。
- *properties* には、ルートのプロパティを空白で区切って指定します。各プロパティを指定する構文は次のとおりです。

```
prop_name=value
```

ルートに沿ってトピックフローを制御するゲーティングプロパティについては、513 ページの「[トピックメッセージのルーティングに使用するセクタ](#)」を参照してください。

ルートの SSL（Secure Sockets Layer）プロトコルを設定するプロパティについては、507 ページの「[ルーティングと SSL](#)」を参照してください。

**例** たとえば、サーバー A に対して次のコマンドを実行すると、サーバー B と C へのルートが作成されます。サーバー B へのルートは、1 ホップゾーン z1 に属しています。サーバー C へのルートは、マルチホップゾーン zM に属しています。

```
create route B url=tcp://B:7454 zone_name=z1 zone_type=1hop
create route C url=tcp://C:7455 zone_name=zM zone_type=mhop
```

**ルートの表示** 管理ツールの **show routes** コマンドを使用して、これらのルートを表示することができます。

```
show routes
Route      T      ConnID  URL              Zone      T
B          A      3      tcp://B:7454     Z1        1
C          A      -      tcp://C:7455     ZM        m
```

- **Route** 列は、パッシブサーバーの名前を示しています。
- **T** 列は、サーバー A から見た場合、ルートがアクティブ (A) とパッシブ (P) のいずれであるかを示しています。
- **ConnID** 列には、ルートが現在接続中の場合は整数による接続 ID が、ルートが接続されていない場合はダッシュ (-) が表示されます。

## フォールトトレランスサーバーへのルーティング

サーバーは、フォールトトレランスに設定することができます。クライアントアプリケーションは、プライマリサーバーとバックアップサーバーを指定することができます。プライマリサーバーへのクライアント接続に障害が発生した場合、クライアントはバックアップサーバーに接続して、処理を再開することができます。同様に、ルート指定では、プライマリパッシブサーバーと、セカンダリパッシブサーバーを指定できます。この設定により、プライマリサーバーへのルートに障害が発生した場合、アクティブサーバーはバックアップサーバーに接続してルーティングを再開することができます。

ルート接続におけるフェイルオーバー時の動作は、クライアント接続におけるフェイルオーバー時の動作とよく似ています。詳細は、492 ページの「[フォールトトレランスサーバー用のクライアントの設定](#)」を参照してください。

例

```
create route B url=tcp://B:7454,tcp://BBackup:7454 zone_name=Z1 zone_type=1hop
```

## ルーティングと SSL

ルートを設定する場合、接続用の SSL パラメータを指定することができます。SSL 接続を構成する両方のサーバーには、同様のパラメータセットを指定する必要がありますが、各サーバーで、この情報を指定する場所は異なります。

- パッシブサーバーは、SSL パラメータを主要設定ファイル **tibemsd.conf** に指定する必要があります。
- サーバーは、SSL 接続の開始時に、パッシブサーバーに対して自己の識別情報を提示し、認証を受けるためにルートの SSL パラメータを送信します。このパラメータは、ルートの作成時に指定することも、ルート設定ファイル **routes.conf** で指定することもできます。

表 80 は、`routes.conf` 設定ファイルで指定したり、ルートの作成時にコマンドラインから指定できるパラメータのリストです。ルーティング先サーバー間の SSL を設定するパラメータは、サーバーとクライアント間の SSL を設定するパラメータとほとんど同じです。詳細は、453 ページの第 18 章、「SSL プロトコルの使用」を参照してください。

表 80 ルートの SSL のパラメータ (1 / 3 シート)

パラメータ	説明
<code>ssl_identity</code>	<p>PEM、DER、または PKCS#12 形式のサーバーのデジタル証明書を指定します。デジタル証明書をこのパラメータの指定にコピーすることも、またはサポートされている形式の証明書が含まれているファイルへのパスを指定することもできます。</p> <p>詳細は、457 ページの「証明書と鍵のファイル名」を参照してください。</p>
<code>ssl_issuer</code>	<p>サーバーの証明書チェインメンバーを指定します。CA ルート証明書を含む、チェイン全体を指定します。サーバーはチェイン内の証明書を、このパラメータで指定されている順番で読み取ります。</p> <p>証明書は、PEM、DER、PKCS#7、または PKCS#12 形式のいずれかでなければなりません。</p> <p><b>例</b></p> <pre>ssl_issuer = certs¥CA_root.pem ssl_issuer = certs¥CA_child1.pem ssl_issuer = certs¥CA_child2.pem</pre> <p>詳細は、457 ページの「証明書と鍵のファイル名」を参照してください。</p>
<code>ssl_private_key</code>	<p>ローカルサーバーの秘密鍵を指定します。この情報が <code>ssl_identity</code> で指定されているデジタル証明書にすでに含まれている場合には、このパラメータを省略してもかまいません。</p> <p>このパラメータは、PEM、DER および PKCS#12 形式の秘密鍵を受け入れます。</p> <p>このパラメータに実際の鍵を指定することも、鍵が含まれているファイルへのパスを指定することもできます。</p> <p>詳細は、457 ページの「証明書と鍵のファイル名」を参照してください。</p>

表 80 ルートの SSL のパラメータ (2 / 3 シート)

パラメータ	説明
ssl_password	<p>秘密鍵または秘密鍵のパスワードを指定します。</p> <p>パスワードは、<code>tibemsadmin</code> ツールを使用して設定できます。このツールでパスワードを設定した場合、パスワードは設定ファイル内で暗号化されます。詳細は、117 ページの第 6 章、「<a href="#">EMS 管理ツールの使用</a>」を参照してください。</p>
ssl_trusted	<p>信頼する認証局を識別するために使用する証明書のリストを指定します。</p> <p>証明書は、PEM、DER、または PKCS#7 形式のいずれかである必要があります。実際の証明書を指定することも、証明書チェーンが含まれているファイルへのパスを指定することもできます。</p> <p>詳細は、457 ページの「<a href="#">証明書と鍵のファイル名</a>」を参照してください。</p>
ssl_verify_host	<p>サーバーに、別のサーバーの証明書を検証させるかどうかを指定します。このパラメータの値は、<code>enabled</code> および <code>disabled</code> です。</p> <p>省略した場合、サーバーが他方のサーバーの証明書を検証する必要があることを示す、デフォルトの <code>enabled</code> が設定されます。</p> <p>このパラメータが <code>disabled</code> に設定されている場合は、サーバーは他方のサーバーとのセキュアな接続を確立しますが、そのサーバーの身元は検証しません。</p>

表 80 ルートの SSL のパラメータ (3 / 3 シート)

パラメータ	説明
ssl_verify_hostname	<p>別のサーバーの証明書の CN フィールドに指定された名前を、サーバーで検証するかどうかを指定します。このパラメータの値は、<b>enabled</b> および <b>disabled</b> です。</p> <p>省略した場合、サーバーが接続先のホストの名前、または <b>ssl_expected_hostname</b> パラメータに指定されている名前を、サーバーの証明書の値に照合して検証する必要があることを示す、デフォルトの <b>enabled</b> が設定されます。名前が一致しない場合、接続は拒否されます。</p> <p>このパラメータが <b>disabled</b> に設定されている場合は、サーバーは他方のサーバーとのセキュアな接続を確立しますが、そのサーバーの名前は検証しません。</p>
ssl_expected_hostname	<p>他方のサーバーの証明書の CN フィールドで必要とされる名前を指定します。このパラメータが設定されていない場合は、デフォルトは、他方のサーバーのホスト名になります。</p> <p>このパラメータは、<b>ssl_verify_hostname</b> が有効になっている場合にのみ関係します。</p>
ssl_ciphers	<p>暗号スイートのリストをコロン (:) で区切って指定します。</p> <p>このパラメータには、暗号スイート用の OpenSSL 名を使用することも、より長くて分かりやすい名前を使用することもできます。</p> <p>利用可能な暗号スイートとその名前についての詳細は、<a href="#">464 ページの「暗号スイートの指定方法」</a>を参照してください。</p>
ssl_rand_egd	<p>EGD (Entropy Gathering Daemon) がインストールされている場合、その場所のパスを指定します。このデーモンは、乱数を生成します。</p>

## トピックメッセージのルーティング

サーバーは、トピックに `global` プロパティが定義されている場合のみ、トピックメッセージをルートに沿って転送します。詳細は、[123 ページの「addprop topic」](#) および [127 ページの「create topic」](#) を参照してください。

トピックメッセージは、複数のホップを移動できます。

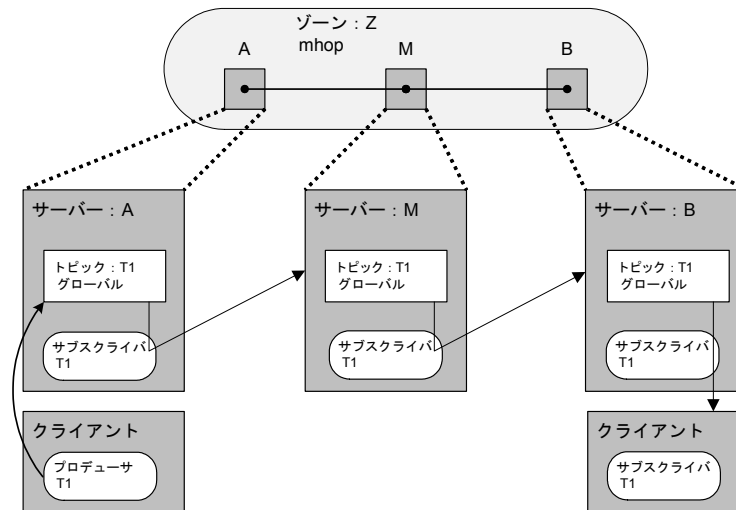
ルートの接続が解除された場合（ネットワーク障害が発生した場合など）、メッセージを転送するサーバーはトピックメッセージを保存します。ルートが再接続されたら、保存しておいたメッセージをこのサーバーが転送します。

ルートによって接続されたサーバーは、テンポラリトピックに送信されたメッセージを交換します。

### 登録されている対象サブスクリプションの伝播

メッセージをルートに沿って確実に転送するには、サーバーは自身のトピックサブスクリプションを別のサーバーに伝播する必要があります。たとえば、[図 37](#) の上部は、三つのサーバー、A、M、および B を、マルチホップゾーンでルートによって接続している企業の例を示しています。[図 37](#) の下部には、プロデューサクライアントのサーバー A から、サーバー M を経由し、サーバー B とそのサブスクライバクライアントに、メッセージがルーティングされる仕組みを示しています。各イベントの流れに注目してください。

図 37 ルーティング：サブスクライバの伝播



1. 三つのサーバーすべてに、グローバルトピック T1 が設定されています。

2. 図 37 の右側の下にあるサーバー B のクライアントが T1 に対するサブスクライバを作成します。
3. サーバー B は、内部サブスクライバのオブジェクトを作成することによって、クライアントに代って T1 にインタレストを登録します。
4. ルートがサーバー M とサーバー B を接続しているので、サーバー B は自身のインタレストを、サーバー M の T1 に伝播します。これに応答し、サーバー M はサーバー B に代わって、T1 に対する内部サブスクライバを作成します。このサブスクライバは、サーバー M がトピック T1 からのメッセージをサーバー B に確実に転送（配信）できるようにします。サーバー B は、サーバー M のクライアントとして動作します。
5. 同様に、ルートがサーバー A とサーバー M を接続しているので、サーバー M は自身のインタレストを、サーバー A の T1 に伝播します。これに応答し、サーバー A はサーバー M の代理で、T1 に対する内部サブスクライバを作成します。このサブスクライバは、サーバー A がトピック T1 からのメッセージを、サーバー M に確実に転送できるようにします。サーバー M は、サーバー A のクライアントとして動作します。
6. サーバー A のプロデューサクライアントがメッセージをトピック T1 に送信すると、サーバー A はそのメッセージをサーバー M に転送します。サーバー M は、自身のトピック T1 に対するメッセージを受け入れ、それをサーバー B に転送します。サーバー B は、自身のトピック T1 に対するメッセージを受け入れ、それをクライアントに転送します。

#### サブスクライバ クライアントの終了

サーバー B のクライアントが、T1 に対する「非持続」サブスクライバを作成し、クライアントプロセスを終了すると、サーバーは内部サブスクライバのシーケンス全体を削除します。クライアントが再起動すると、サブスクライバの新しいシーケンスが生成されます。クライアントは、その間に到着したメッセージを受け取れません。

サーバー B のクライアントが、T1 に対する「持続」サブスクライバを作成し、その後クライアントプロセスが終了した場合、内部サブスクライバのシーケンス全体はそのまま変更されません。メッセージは、保存後に転送される形で、サーバー内のフローに従って処理されます。クライアントが再起動すると、プロセスの停止中にサーバー B が保存していたすべてのメッセージをコンSUME することができます。

#### サーバーの障害

サーバー B とサーバー M 間のアクティブ - アクティブルートでは、サーバー B に障害が発生した場合、サーバー M が内部サブスクライバを保持し、サーバー B のクライアント宛てのメッセージを保存します。サーバー B が再接続したら、サーバー M は保存したメッセージを転送します。

サーバー B にアクティブ - パッシブルートが設定されていて、障害が発生した場合、サーバー M は自身の内部サブスクライバを削除し、サーバー B のクライアントのためにメッセージを保存することはありません。これにより、メッセージストリーム内にギャップが生じる可能性があります。サーバー B の再接続時に、サーバー M は新しい内部サブスクライバを作成し、メッセージの転送を再開します。

サーバー A にアクティブ - パッシブルートが設定されていて、いずれかのサーバーで障害が発生した場合、サーバー M はアクティブ - アクティブルートと同じ方法で内部サブスクライバを保持します。ただし、サーバー B はサーバー A からの重複メッセージを抑制するために使用する内部ステートを保持せず、メッセージのコンSUME 後にそのメッセージをコンSUMER に配信できます。したがって、メッセージの消失と重複を回避するには、ルートをアクティブ - アクティブにする必要があります。



- ネットワーク障害** サーバー B とサーバー M 間のアクティブ - パッシブ接続が中断された場合、サーバー M にはサーバーの障害発生時と同じ内容が表示されます。
- maxbytes** 持続サブスクライバをルートと組み合わせることにより、障害が発生した場合などに、ストレージに潜在的な需要が生じます。たとえば、サーバー B に障害が発生した場合、サーバー M はサーバー B が再開するまでメッセージを保存します。ストレージ使用率が無制限に増大したために処理が中断するのを防ぐには、各サーバーについて、トピック (T1) の **maxbytes** または **maxmsgs** プロパティを設定することをお勧めします。

## トピックメッセージのルーティングに使用するセレクトラ

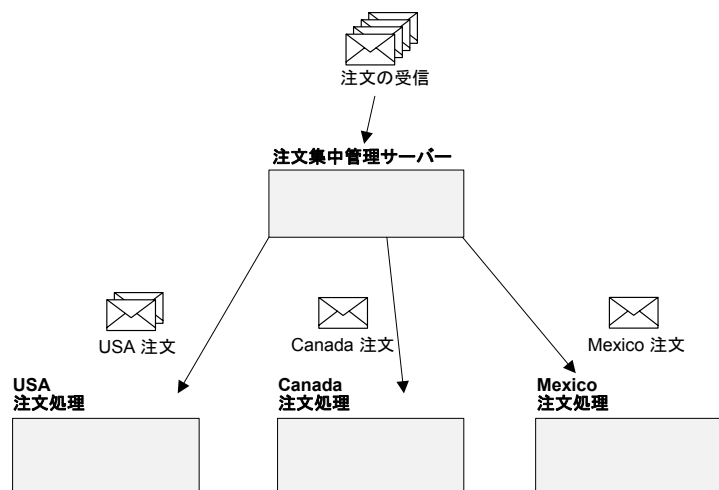
- 使用例** サーバーは、グローバルトピックメッセージを、そのトピックのサブスクライバを持つすべてのサーバーにルートに沿って転送します。転送先の各サーバーが、転送されるメッセージのごく一部のみを必要としている場合、この処理により、不要なネットワークトラフィックが大量に発生するおそれがあります。ルートに「メッセージセレクトラ」を設定することで、各ルートを移動するトピックメッセージのサブセットを制限することができます。



ルートのメッセージセレクトラは、個々のサブスクライバのメッセージセレクトラとは異なります。ルートのメッセージセレクトラは、サーバーがサブスクライバクライアントに配信するメッセージのサブセットを制限します。

- 例** 513 ページの「[図 38](#)」は、ある企業に、顧客の注文を処理する集中管理サーバーがひとつあり、注文の請求処理を行うサーバーが各地域に個別に設定されている様子を示しています。ネットワーク容量の使用を最適化するには、各地域のサーバーが担当地域内の顧客からの注文のみを受け取るように、トピックセレクトラを設定します。

図 38 ルーティング：トピックセレクトラの例



**セレクトラの指定** グローバルトピックのメッセージセレクトラを、ルートのプロパティとして指定します。プロパティを設定するには、次のふたつの方法があります。

- ルートの作成時にセレクトラを定義する (`routes.conf` 内に設定するか、または管理コマンドの `create route` を実行する)。
- 既存のルート of セレクトラを操作する (管理コマンドの `addprop`、`setprop`、または `removeprop` を使用する)。



ルート上のメッセージセレクトラを変更した場合、受信メッセージのみが新しいセレクトラで評価されます。サーバーで保留中のメッセージは、サーバーが再起動された場合にのみ再評価されます。

**構文** メッセージセレクトラのプロパティを指定するには、コマンドとして記述するか、設定ファイルに記述するかにかかわらず、同じ構文を使用します。

```
incoming_topic=topicName selector="msg-selector"
outgoing_topic=topicName selector="msg-selector"
```



*incoming* と *outgoing* は、ルートが定義されているアクティブサーバーの観点を示します。

*topicName* は、グローバルトピックの名前です。

*msg-selector* は、メッセージセレクトラの文字列です。メッセージセレクトラ構文についての詳細は、『TIBCO Enterprise Message Service Java API リファレンス』の `Message` クラスに関する記述を参照してください。

**構文例** 図 38 の例で、管理者が注文の集中管理サーバーのルートを設定する場合は、次のように入力します。

```
setprop route Canada outgoing_topic="orders" selector="country='Canada'"
setprop route Mexico outgoing_topic="orders" selector "country='Mexico'"
setprop route USA    outgoing_topic="orders" selector="country='USA'"
```

これらのコマンドにより、次の各エントリが `routes.conf` に作成されます。

```
[Canada]
url=ssl://canada:7222
outgoing_topic=orders selector="country='Canada'"
...
[Mexico]
url=ssl://mexico:7222
outgoing_topic=orders selector="country='Mexico'"
...
[USA]
url=ssl://usa:7222
outgoing_topic=orders selector="country='USA'"
...
```

- 対称性** `outgoing_topic` と `incoming_topic` には対称性があります。サーバー A が `incoming_topic` セレクタによりサーバー B へのルート指定する場合も、またはサーバー B が、`outgoing_topic` セレクタによりサーバー A へのルート指定する場合も、結果は同じです。サーバー B は、ルート上のセレクタに一致するメッセージのみを送信します。
- アクティ - アクティブ設定** アクティブ - アクティブ設定では、セレクタを一方のサーバーだけに指定することも、両方のサーバーに指定することもできます。サーバー A のトピック T に `outgoing_topic` セレクタ S1 を指定し、サーバー B のトピック T に `incoming_topic` セレクタ S2 を指定すると、サーバー A からサーバー B へのルート上のトピック T の有効なセレクタは、(S1 AND S2) となります。
- 505 ページの「[アクティブルートとパッシブルート](#)」も参照してください。
- ワイルドカード** トピック名にワイルドカードを指定することができます。個々の実際のトピックについて、サーバーは論理演算子 AND を使用して、トピックに一致するすべてのセレクタを組み合わせます。



ただし、トピックメッセージのルーティングは、サブスクリバのトピックが設定済みグローバルトピックのサブセット（または同一トピック）の場合にのみ確実にサポートされます。また、共有部分はサポートされていません。たとえば、`topics.conf` に `foo.*` と `foo.a*` が含まれる場合、以下のサブスクリプションは有効です。

```
foo.*
foo.1
bar.a.b
```

以下のサブスクリプションは無効です。

```
foo.>
bar.*.b
```

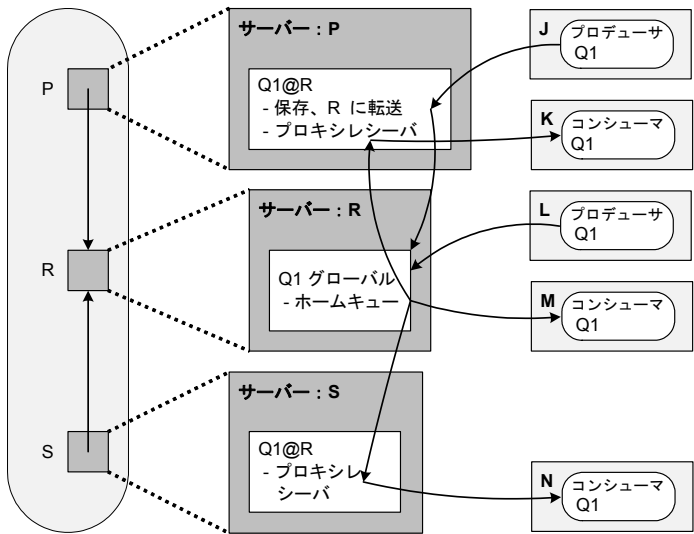
# ルーティング先キュー

ルーティングについては、キューは次のような点においてトピックとは異なります。

- サーバーは、キューの所有者と隣接するサーバー間のキューメッセージをルーティングします。
- ゾーンとホップの概念は、トピックメッセージにのみ該当し、キューメッセージには適用されません。

図 39 の左側は、いくつかのルートで接続された P、R、S の三つのサーバーを持つ企業を示しています。図 39 のその他の部分は、ルートが、サーバー（中央）およびクライアント（右側）間で、キューメッセージをルーティングする仕組みを示しています。

図 39 ルーティング：キュー



**所有者とホーム**      サーバー R は、Q1 という名前のグローバルキューを定義しています。R は Q1 の「所有者」です。

サーバー P と S は「ルーティング先キュー」Q1@R を定義しています。この指定は、これらのキューが、「ホームキュー」（サーバー R 上の Q1）に依存していて、これを反映していることを示しています。Q1@R は、設定用のためだけの指定です。P のクライアントは、ルーティング先キューを Q1 として参照します。

**例** クライアント J がメッセージを Q1 に送信するとき、サーバー P はメッセージをホームキュー（サーバー R 上の Q1）に転送します。

これで、この三つのサーバー P、R、S、のすべてのレシーバでメッセージが利用可能になりますが、メッセージをコンSUMするののはひとつのクライアントのみです。サーバー P の Q1 が、クライアント K の代理としてメッセージを受信するか、サーバー S の Q1 がクライアント N の代理として受信するか、またはクライアント M がホームキューから直接メッセージを受信します。

**プロデューサ** プロデューサクライアントから見ると、ルーティング先キューがメッセージを保存し、保存したメッセージをホームキューに転送します。たとえば、クライアント J がメッセージをサーバー P 上の Q1 に送信すると、P がそのメッセージをキューの所有者 R に転送し、R はそのメッセージを Q1（ホームキュー）に配信します。

メッセージは、ホームキューに到達するまでは、コンシューマから利用できません。つまり、クライアント K は、サーバー P から直接メッセージをコンSUMすることはできません。

サーバー R に障害が発生した場合、または P から R へのルート接続に障害が発生した場合、P は、クライアント K からのメッセージのキューへの保存を続行します。P と R が通信を再開したら、P は、保存したメッセージを R の Q1 に配信します。

また、ルーティング先サーバーで `maxbytes` と `maxmsgs` の制限を超えた場合、ルーティング先キューは例外を生成しません。クライアントは制限に達した後も引き続きキューにメッセージを送信可能で、メッセージはエラー条件がクリアされるまでルーティング先サーバーに保存されます。

**コンシューマ** コンシューマクライアントから見ると、ルーティング先キューは、プロキシレシーバとして動作します。たとえば、クライアント L が、サーバー R の Q1 にメッセージを送信した場合、サーバー P の Q1 は、クライアント K の代理として R からそのメッセージを受信し、即座にそのメッセージを K に送信します。

サーバー P に障害が発生した場合、または P から R へのルート接続に障害が発生した場合は、これらのサーバーが通信を再開するまで、クライアント K は Q1 からのメッセージを受信できません。その間、クライアント M と N は、Q1 からのメッセージを受信し続けます。サーバー P と R が通信を再開すると、クライアント K は P の Q1 を経由して再びメッセージを受信できるようになります。



短いタイムアウト（1 秒未満）または待ち時間なしでルーティング先キューからメッセージを受信すると、予期しない動作が発生する可能性があります。短いタイムアウト値は、複数のルーティング先サーバー間でプロトコルメッセージが正しく処理されない可能性が高くなります。たとえば、キューレシーバが適切に破棄されない可能性があります。

**設定** 各ルーティング先キューは、`queues.conf` で明示的に設定する必要があります。クライアントは、ルーティング先キューを動的に作成することはできません。



動的なルーティング先キューはサポートされていません。今後のリリースのサーバーでは、ワイルドカードを含むルーティング先キューは設定の誤りとみなされ、`startup_abort_list` に `CONFIG_ERRORS` が含まれる場合は起動に失敗します。

管理ツールまたは管理 API を使用して、ルーティング先キューを設定することができます。122 ページの「[addprop queue](#)」および 126 ページの「[create queue](#)」を参照してください。

ルーティング先キューを設定するには、キュー名と、キューの所有者のサーバー名を指定する必要があります。たとえば、サーバー P 上に設定する場合、以下のよう指定します。

Q1@R



この記法は、ホームキューを表現する場合でも有効です。キューの所有者は自身の名前を認識し、ローカル指定 (@R) を無視します。

ルーティング先キューを **exclusive** として設定することはできません。

#### ブラウザ

キューブラウザは、ルーティング先キューを調査できません。ブラウザは、ホームキューを所有するサーバー上にのみ作成できます。

#### トランザクション

TIBCO Enterprise Message Service では、(XA またはローカルトランザクションセッションを使用した) ルーティング先キューでのトランザクションコンシューマはサポートされていません。

## ルーティングと認可

### ユーザーおよびパスワード

サーバーの **authorization** パラメータが有効になっている場合、能動的にそのサーバーに接続する他のサーバーは、名前とパスワード、または X.509 証明書によって認証を受ける必要があります。

図 40 ルーティング：認可

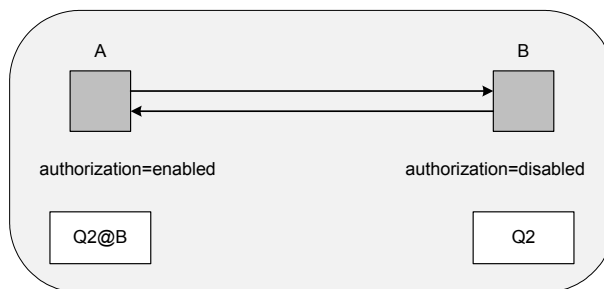


図 40 は、サーバー A と B が、お互いに他方へのアクティブルートを持つように設定されています。

- A では認可が有効になっているため、A は B という名前のユーザーを設定する必要があります。
- B では認可が無効になっているため、A は B に対して身元を証明する必要がありません。また、B に A という名前のユーザーを設定する必要はありません。

### ACL

セキュアトピックまたはセキュアキューをルーティングする場合、サーバーは各メッセージを転送する前に ACL の指定を確認します。サーバーは、送信、受信、パブリッシュ、サブスクライブを行うための適切なアクセス権限を互いに付与する必要があります。

たとえば、図 40 では、A（プロデューサの送信先）から B（コンシューマのコンシューム元）に転送されるメッセージに ACL は必要ありません。B では認可が無効になっていて、メッセージはキューに送信され、キューからコンシュームされるからです。ただし、メッセージを B から A に転送する場合（プロデューサから B に接続し、コンシューマから A に接続する場合）、サーバー A の ACL は、ユーザー B のキュー Q2 に **send** アクセス権限に付与する必要があります。

この例のトピックを使用する場合、A から B に転送されるメッセージについては、A が B にトピック（両サーバーで **global**）に対する **subscribe** および **durable** のアクセス権限を付与する必要があります。B から A に転送されるメッセージについては、トピック B に、トピックに対する **publish** アクセス権限を付与する必要があります。

関連トピック 255 ページの第 8 章、「認証とアクセス権限」





## 付録 A

# メッセージの監視

この付録では、システムイベントへのメッセージをサーバーがパブリッシュする際、そのパブリッシュ先となるすべてのトピックを記載しています。各トピックにパブリッシュするメッセージのメッセージプロパティについても説明します。トピックとメッセージの監視についての詳細は、442 ページの「[サーバーイベントの監視](#)」を参照してください。

## トピック

---

- 「[監視トピックの説明](#)」、522 ページ
- 「[トピックメッセージプロパティの説明](#)」、525 ページ

## 監視トピックの説明

表 81 では、各監視トピックを説明します。

表 81 監視トピック

Topic	メッセージがパブリッシュされる状況
<code>\$sys.monitor.admin.change</code>	管理者が設定に変更を加えた。
<code>\$sys.monitor.connection.connect</code>	ユーザーがサーバーに接続を試みた。
<code>\$sys.monitor.connection.disconnect</code>	ユーザー接続が切断された。
<code>\$sys.monitor.connection.error</code>	ユーザー接続でエラーが発生した。
<code>\$sys.monitor.consumer.create</code>	コンシューマが作成された。
<code>\$sys.monitor.consumer.destroy</code>	コンシューマが破棄された。
<code>\$sys.monitor.flow.engaged</code>	保存されたメッセージがデスティネーションの制限を上回るため、フロー制御機能を起動している。
<code>\$sys.monitor.flow.disengaged</code>	保存されたメッセージがデスティネーションの制限を下回るため、フロー制御機能を停止している。
<code>\$sys.monitor.limits.connection</code>	ホストまたは接続の最大数に到達した。
<code>\$sys.monitor.limits.queue</code>	キューストレージの最大バイト数に到達した。
<code>\$sys.monitor.limits.server</code>	サーバーのメモリ制限に達した。
<code>\$sys.monitor.limits.topic</code>	持続サブスクリプションの最大バイト数に到達した。
<code>\$sys.monitor.multicast.stats</code>	パブリッシュされたメッセージにサーバーとマルチキャストデーモンからの低レベルの PGM 統計情報が含まれている。
<code>\$sys.monitor.multicast.status</code>	メッセージコンシューマがマルチキャスト対応のトピックをサブスクライブしたかサブスクライブを試みた。
<code>\$sys.monitor.producer.create</code>	プロデューサが作成された。
<code>\$sys.monitor.producer.destroy</code>	プロデューサが破棄された。
<code>\$sys.monitor.queue.create</code>	動的キューが作成された。

表 81 監視トピック

Topic	メッセージがパブリッシュされる状況
<code>\$sys.monitor.route.connect</code>	ルート接続が試行された。
<code>\$sys.monitor.route.disconnect</code>	ルート接続が切断された。
<code>\$sys.monitor.route.error</code>	ルート接続でエラーが発生した。
<code>\$sys.monitor.route.interest</code>	ルートで、登録された対象が変更された。
<code>\$sys.monitor.server.info</code>	ログファイルがローテーションされたなど、イベントに関する情報をサーバーが送信した。
<code>\$sys.monitor.server.warning</code>	プライマリサーバーがバックアップサーバーからの切断を検出した。
<code>\$sys.monitor.topic.create</code>	動的トピックが作成された。
<code>\$sys.monitor.tx.action</code>	ローカルトランザクションがコミットまたはロールバックされた。
<code>\$sys.monitor.xa.action</code>	XA トランザクションがコミットまたはロールバックされた。

表 81 監視トピック

Topic	メッセージがパブリッシュされる状況
<code>\$sys.monitor.D.E.destination</code>	<p>メッセージはデスティネーションによって処理されます。この監視トピックの名前には、ふたつの修飾子 (<i>D</i> および <i>E</i>) と監視するデスティネーションの名前が含まれています。</p> <p><i>D</i> は、デスティネーションのタイプと、メッセージ全体を含めるかどうかを示します。</p> <ul style="list-style-type: none"><li>• <b>T</b> : トピック、各イベントへのメッセージ全体 (バイト配列) を含む</li><li>• <b>t</b> : トピック、各イベントへのメッセージ全体を含まない</li><li>• <b>Q</b> : キュー、各イベントへのメッセージ全体 (バイト配列) を含む</li><li>• <b>q</b> : キュー、各イベントへのメッセージ全体を含まない</li></ul> <p><i>E</i> は、イベントのタイプを示します。</p> <ul style="list-style-type: none"><li>• <b>r</b> : 受信</li><li>• <b>s</b> : 送信</li><li>• <b>a</b> : 確認応答</li><li>• <b>p</b> : メッセージの途中終了</li><li>• <b>*</b> : すべてのイベントタイプ</li></ul> <p>たとえば、<code>\$sys.monitor.T.r.corp.News</code> は、<code>corp.News</code> という名前のトピックに対するすべての受信メッセージを監視するトピックです。受信したすべてのメッセージの本文は、このトピックの監視メッセージに含まれます。トピック <code>\$sys.monitor.q.*.corp.*</code> は、名前 <code>corp.*</code> に一致するすべてのキューに対して、すべてのメッセージイベント (送信、受信、確認応答) を監視します。メッセージの本文は、このトピックのメッセージには含まれません。</p> <p>このタイプの監視トピックに送信されるメッセージには、イベントの説明、およびメッセージの送信元 (プロデューサ、ルート、外部システムなど) の情報が含まれます。さらに、<i>D</i> の値によっては、メッセージ本文が含まれる場合もあります。</p> <p>メッセージの監視についての詳細は、442 ページの「<a href="#">メッセージの監視</a>」を参照してください。</p>

## トピックメッセージプロパティの説明

表 82 では、監視トピックメッセージに含めることのできるプロパティを説明します。各監視メッセージには、これらのプロパティをさまざまに組み合わせて指定することができます。

表 82 メッセージのプロパティ

プロパティ	内容
conn_connid	イベントを生成した接続の接続 ID。
conn_ft	クライアント接続が、フォールトトレランスサーバーへの接続かどうか。
conn_hostname	イベントを生成した接続のホスト名。
conn_ssl	接続で SSL プロトコルが使用されているかどうか。
conn_type	<p>イベントを生成した接続のタイプ。このプロパティには、次の値を指定できます。</p> <ul style="list-style-type: none"> <li>• Admin</li> <li>• Topic</li> <li>• Queue</li> <li>• Generic</li> <li>• Route</li> <li>• FT (フォールトトレランスサーバーへの接続)</li> <li>• Unknown</li> </ul>
conn_username	イベントを生成した接続のユーザー名。
conn_xa	クライアント接続が XA 接続かどうか。
event_action	イベントを発生させたアクション。このプロパティには、530 ページの「表 83」に記載されている値を指定できます。
event_class	<p>監視イベントのタイプ（つまり、<code>\$sys.monitor</code> を除いたトピック名の最後の部分）。</p> <p>メッセージを監視するには、このプロパティの値を常に <code>message</code> に設定します。</p>

表 82 メッセージのプロパティ

プロパティ	内容
event_description	発生したイベントのテキストによる説明。
event_reason	イベントが発生した理由（通常はエラー）。このプロパティに設定可能な値については、532 ページの「表 84」を参照してください。
event_route	ルーティングする場合に、イベントが発生したルート。
message_bytes	メッセージ全体がメッセージの監視の対象となるときに、このフィールドにはメッセージがバイト配列として含まれます。各種クライアント API の <code>createFromBytes</code> メソッドを使用して、メッセージを回復することも可能です。
mode	メッセージの配信モード。このプロパティには、次の値を指定できます。 <ul style="list-style-type: none"><li><code>persistent</code></li><li><code>non_persistent</code></li><li><code>reliable</code></li></ul>
msg_id	メッセージ ID。
msg_seq	メッセージのシーケンス番号。
msg_size	メッセージサイズ（バイト単位）。
msg_timestamp	メッセージのタイムスタンプ。
msg_expiration	メッセージの有効期限。
replyTo	メッセージの <code>JMSReplyTo</code> 。
rv_reply	メッセージの RV 返信サブジェクト。
source_id	ソースオブジェクトの ID。

表 82 メッセージのプロパティ

プロパティ	内容
source_name	<p>イベントに関連するソースオブジェクトの名前。このプロパティには、次の値を指定できます。</p> <ul style="list-style-type: none"><li>• <code>XID</code> (グローバルトランザクション ID)</li><li>• <code>message_id</code></li><li>• <code>connections</code> (接続数)</li><li>• <code>unknown</code> (不明な名前)</li><li>• 任意のサーバープロパティの名前</li><li>• ユーザーの名前、または <code>anonymous</code></li></ul>
source_object	<p>イベントに関連したソースオブジェクト。このプロパティには、次の値を指定できます。</p> <ul style="list-style-type: none"><li>• <code>producer</code></li><li>• <code>consumer</code></li><li>• <code>topic</code></li><li>• <code>queue</code></li><li>• <code>permissions</code></li><li>• <code>durable</code></li><li>• <code>server</code></li><li>• <code>transaction</code></li><li>• <code>user</code></li><li>• <code>group</code></li><li>• <code>connection</code></li><li>• <code>message</code></li><li>• <code>jndiname</code></li><li>• <code>factory</code></li><li>• <code>file</code></li><li>• <code>limits</code> (メモリ制限など、制限)</li><li>• <code>route</code></li><li>• <code>transport</code></li></ul>

表 82 メッセージのプロパティ

プロパティ	内容
source_value	ソースオブジェクトの値。
stat_msgs	プロデューサまたはコンシューマのメッセージ数の統計情報。
stat_size	プロデューサまたはコンシューマのメッセージサイズの統計情報。
target_admin	ターゲットオブジェクトが <b>admin</b> 接続かどうか。
target_created	コンシューマが作成された時間（作成時点からのミリ秒数）。
target_dest_name	ターゲットデスティネーションの名前。
target_dest_type	ターゲットデスティネーションのタイプ。
target_durable	ターゲットが持続サブスクライバの場合、持続サブスクライバの名前。
target_group	イベントのターゲットであったグループの名前。
target_hostname	ターゲットオブジェクトのホスト名。
target_id	ターゲットオブジェクトの ID。
target_channel	マルチキャストチャネルの名前。
target_name	イベントのターゲットであったオブジェクトの名前。このプロパティには、次の値を指定できます。 <ul style="list-style-type: none"><li>• <b>XID</b>（グローバルトランザクション ID）</li><li>• <b>message_id</b></li><li>• <b>connections</b>（接続数）</li><li>• <b>unknown</b>（不明な名前）</li><li>• 任意のサーバープロパティの名前</li><li>• ユーザーの名前、または <b>anonymous</b></li><li>• <b>channel</b>（マルチキャストチャネル）</li></ul>
target_nolocal	ターゲットが持続サブスクライバの場合、 <b>NoLocal</b> フラグ。



表 82 メッセージのプロパティ

プロパティ	内容
target_object	<p>イベントのターゲットであった一般オブジェクトの名前。このプロパティには、次の値を指定できます。</p> <ul style="list-style-type: none"><li>• producer</li><li>• consumer</li><li>• topic</li><li>• queue</li><li>• durable</li><li>• server</li><li>• transaction</li><li>• user</li><li>• group</li><li>• connection</li><li>• message</li><li>• jndiname</li><li>• factory</li><li>• file</li><li>• limits (メモリ制限など、制限)</li><li>• route</li><li>• transport</li></ul>
target_selector	ターゲットがコンシューマの場合、セレクタ。
target_subscription	ターゲットが持続サブスクライバの場合、ターゲットオブジェクトのサブスクリプション。
target_url	ターゲットオブジェクトの URL。
target_username	ターゲットオブジェクトのユーザー名。
target_value	イベントのターゲットであったオブジェクトの値。たとえば、トピック、キューなどの名前。

表 83 イベントアクションのプロパティ値

イベントアクションの値	説明
accept	接続が受け入れられました。
acknowledge	メッセージが確認応答がされました。
add	ユーザーがグループに追加されました。
admin_commit	管理者が手動で XA トランザクションをコミットしました。
admin_rollback	管理者が手動で XA トランザクションをロールバックしました。
commit	トランザクションがコミットされました。
connect	接続が試行されました。
create	作成されたアイテムがあります。
delete	削除されたアイテムがあります。
disconnect	接続が切断されました。
flow_engaged	保存されたメッセージがデスティネーションの制限を上回るため、フロー制御機能を起動しています。
flow_disengaged	保存されたメッセージがデスティネーションの制限を下回るため、フロー制御機能を停止しています。
interest	登録された、ルートの対象。
modify	変更されたアイテムがあります。
grant	アクセス権限が付与されました。
premature_exit	メッセージが途中で終了しました。
purge	トピック、キュー、または持続サブスクライバがページされました。
receive	メッセージがデスティネーションにポストされました。
remove	ユーザーがグループから削除されました。
resume	管理者がルートを再開しました。
revoke	アクセス権限が取り消されました。
rollback	トランザクションがロールバックされました。
rotate_log	ログファイルがローテーションされました。

表 83 イベントアクションのプロパティ値

イベントアクションの値	説明
send	メッセージがサーバーによって別の送信先に送信されました。
subscribe	サブスクリプションが要求されました。
suspend	管理者がルートを一時停止しました。
txcommit	管理者が手動でローカルトランザクションをコミットしました。
txrollback	管理者が手動でローカルトランザクションをロールバックしました。
xaccommit	アプリケーションが XA トランザクション (2 フェーズ) をコミットしました。
xaccommit_1phase	アプリケーションが XA トランザクション (1 フェーズ) をコミットしました。
xastart	アプリケーションが新しい XA トランザクションを開始しました。
xastart_join	アプリケーションがリソースを既存のトランザクションに参加させました (追加しました)。
xastart_resume	アプリケーションが、一時停止された XA トランザクションを再開しました。
xaend_fail	アプリケーションが XA トランザクションを終了しました。障害を示しています。
xaend_success	アプリケーションが XA トランザクションを終了しました。成功を示しています。
xaend_suspend	アプリケーションが XA トランザクションを一時停止しました。
xaprepare	アプリケーションが XA トランザクションを準備しました。
xarecover	XA トランザクションのリストを取得するために、アプリケーションが回復を呼び出しました。
xarollback	アプリケーションが XA トランザクションをロールバックしました。

表 84 イベント発生理由のプロパティ値

イベント発生理由の値	説明
backup_connected	フォールトトレランスバックアップサーバーが接続されました。
backup_disconnected	フォールトトレランスバックアップサーバーの接続が切断されました。
bridge	ブリッジの結果として、メッセージがデスティネーションにポストされました。
closed	接続が閉じられました。
consumer	メッセージの監視の場合、コンシューマによってメッセージが送信されたか、または確認応答されたことを示します。それ以外のすべての状況では、コンシューマのために動的トピックまたはキューが作成されたことを示します。
cycle	ルートの閉路が作成されました。
disabled	機能が有効になっていません。
discarded	新しいメッセージ用の領域を確保するために、デスティネーションの最も古いメッセージが破棄されました。これは、デスティネーションに <code>overflowPolicy=discardOld</code> が設定され、さらにデスティネーションに設定されている <code>maxmsgs</code> や <code>maxbytes</code> 制限を超過したときに発生します。
duplicate	ルート、グローバルキュー、トピックなどが重複しています。
error	エラーが発生したため、接続が切断されました。
exceeded	制限を超過しました。
expired	サーバーによって、メッセージが期限切れになりました。
export	メッセージがトランスポートにエクスポートされました。

表 84 イベント発生理由のプロパティ値

イベント発生理由の値	説明
import	メッセージがトランスポートからインポートされました。
invalid_name	ルート名など、名前が無効です。
invalid_password	無効なパスワードが入力されました。
maxredelivery_exceeded	メッセージがキューの <b>maxRedelivery</b> の数を超過しました。
new_connection	サーバーに新しい接続が確立されました。
not_authorized	アクションを実行する権限がありません。
not_connected	接続を確立できませんでした。
not_found	要求されたアイテムが見つかりませんでした。
producer	メッセージの監視の場合、プロデューサによってポストされたメッセージを示します。それ以外のすべての状況では、プロデューサのために作成された動的トピックまたはキューを示します。
reconnect_active	アクティブな接続。
reconnected_connection	サーバーへの接続が再確立されました。
reconnect_unknown	不明な接続。
rotatelog	ログファイルがローテーションされました。
route	メッセージの監視の場合、ルートからメッセージが送信されたか、または受信されたことを示します。それ以外のすべての状況では、ルートのために作成された動的トピックまたはキューを示します。
shutdown	サーバーがシャットダウンされました。
standby	サーバーがスタンバイモードです。
subscribed	サブスクリプションの要求に成功しました。
terminated	接続が切断されました。



## 付録 B

## エラーメッセージとステータスメッセージ

この付録では、サーバーから出力される可能性があるすべてのエラーメッセージをカテゴリごとに記載しています。

---

## この付録の構成

カテゴリ	エラーの大分類を示します。 この付録は、カテゴリ別に構成されています。
説明	エラーのカテゴリを、より詳しく説明します。
解決方法	管理者が検討する必要がある、問題の解決に役立つアクションを示します。
エラー	EMS サーバーコードに表示される、すべてのエラーが記載されています。一部のカテゴリには複数のエラーが記載されていますが、それ以外のカテゴリではエラーはひとつだけです。エラーの文字列には、書式設定のための文字が含まれている場合があります。

---

## エラーメッセージとステータスメッセージ

---

カテゴリ	持続コンシューマが、存在しないルートストアファイルに見つかった
説明	サーバーの起動時に、 <code>routes.conf</code> ファイルにリストされていないルートストアファイル内に、持続コンシューマが見つかりました。この問題は、 <code>routes.conf</code> ファイルを手動で編集した場合に発生します。
解決方法	管理ツールを使用してルーティングを変更します。
エラー	Discarding durable '%s' for route '%s' because the route does not exist. ( ルート '%s' が存在しないため、このルートの持続 '%s' を破棄しています。 )

---

カテゴリ	管理コマンドの失敗
説明	管理ツールまたは管理 API を使用するプログラムが処理を試みましたが、指定された理由で失敗しました。
解決方法	管理ツールまたは管理 API に、失敗の理由が示されます。ツールまたは API のユーザーは、エラーを調査して、失敗の原因となった構文、パラメータ、または設定を修正してください。
エラー	<p>%s: create %s failed: conflicting zone: existing consumer has a different zone (%s : %s の作成に失敗しました : ゾーンが競合しています : 既存のコンシューマに、異なるゾーンがあります)</p> <p>%s: create %s failed: detected duplicate durable subscription [%s] for topic [%s]. (%s : %s の作成に失敗しました : トピック [%s] の持続サブスクリプション [%s] の重複が検出されました。)</p> <p>%s: create %s failed: illegal to use wildcard %s [%s]. (%s : %s の作成に失敗しました : ワイルドカード %s [%s] を使用することはできません。)</p> <p>%s: create %s failed: invalid %s [%s]. (%s : %s の作成に失敗しました : 無効な %s [%s] です。)</p> <p>%s: create %s failed: invalid session id=%d. (%s : %s の作成に失敗しました : 無効なセッション ID = %d です。)</p> <p>%s: create %s failed: invalid syntax of %s [%s]. (%s : %s の作成に失敗しました : %s [%s] の構文が無効です。)</p> <p>%s: create %s failed: invalid temporary %s [%s]. (%s : %s の作成に失敗しました : 無効なテンポラリー %s [%s] です。)</p>



%s: create %s failed: not allowed to create dynamic %s [%s]. (%s : %s の作成に失敗しました : 動的 %s [%s] の作成は許可されていません。)

Invalid consumer in recover one msg request. (ひとつのメッセージの回復要求内に無効なコンシューマがあります。)

Invalid sequence number in recover one msg request. (ひとつのメッセージの回復要求内に無効なシーケンス番号があります。)

カテゴリ 認証エラー

説明 EMS サーバーは、ユーザーまたはパスワードを認証できませんでした。

解決方法 EMS にユーザーを定義するのに、主要設定ファイル内の `user_auth` パラメータによって許可されているメソッドのいずれかが使用されていることを確認します。ユーザーは、アプリケーションによって、または SSL 証明書内で指定します。ユーザーが定義されている場合、パスワードをリセットしてからもう一度試してください。

エラー Unable to initialize connection, SSL username error. (接続を初期化できません。SSL ユーザー名エラーです。)

LDAP authentication failed for user '%s', status = %d (ユーザー '%s' の LDAP 認証に失敗しました。ステータス = %d)

LDAP authentication failed for user '%s', no password provided (ユーザー '%s' の LDAP 認証に失敗しました。パスワードが入力されていません)

カテゴリ バックアップサーバー '%s' の切断

説明 バックアップフォールトトレランスサーバーとの接続が失われました。

解決方法 バックアップサーバーが実行中かどうかを特定します。実行中の場合、ネットワークパーティションを確認します。

エラー Backup server '%s' disconnected. (バックアップサーバー '%s' が切断されました。)

カテゴリ コマンドラインパラメータの値が不正であるか、指定されていない

説明 コマンドラインパラメータに無効な値が指定されました。

解決方法 指定されたパラメータの値を、適切な値に変更します。tibemsd コマンドラインパラメータについての詳細は、EMS のマニュアルを参照してください。

エラー	<p>'%s' requires an integer argument. ('%s' の引数には、整数を指定する必要があります。)</p> <p>'%s' requires a positive integer argument. ('%s' の引数には、正の整数を指定する必要があります。)</p> <p>'%s' requires a string argument. ('%s' の引数には、文字列を指定する必要があります。)</p>
カテゴリ	バナーとデバッグトレース
説明	バナーとデバッグトレース
解決方法	該当なし
エラー	<p>%s: %s has been changed (%s : %s が変更されました )</p> <p>%s: %s has been changed to %s (%s : %s が %s に変更されました )</p> <p>%s: %s has been changed to %d (%s : %s が %d に変更されました )</p> <p>%s: %s has been changed to % PRINTF_LLFMT d (%s : %s が % PRINTF_LLFMT d に変更されました )</p> <p>Invalid session for route configuration. ( ルート設定のセッションが無効です。 )</p> <p>Invalid routed queue information message. ( ルーティング先キューの情報メッセージが無効です。 )</p> <p>Expired % PRINTF_LLFMT d message%s. (% PRINTF_LLFMT d メッセージ %s の有効期限が切れています。 )</p> <p>Discarded % PRINTF_LLFMT d message%s. (% PRINTF_LLFMT d メッセージ %s が破棄されました。 )</p> <p>[%s@%s]: rejected connect from route: invalid password ([%s@%s] : ルートからの接続が拒否されました : パスワードが無効です )</p> <p>%s: purged durable '%s' (%s : 持続 '%s' がパージされました )</p> <p>%s: %s %s '%s' permissions on %s '%s': %s (%s : %s '%s' に対する %s %s '%s' アクセス権限 : %s)</p> <p>%s: create %s failed: durable creation access denied for %s [%s]. (%s : %s の作成に失敗しました : %s [%s] に対する持続作成アクセス権限が拒否されました。 )</p> <p>Async Recs: max=%d avg=%.2f min=%d ( 非同期レコード : 最大 = %d 平均 = %.2f 最小 = %d)</p> <p>Process Id: %d ( プロセス ID : %d)</p>

Server activating on failure of '%s'. ('%s' で障害が発生したため、サーバーをアクティブ化しています。)

ldap\_search\_ext\_s(%x, %s, %s, %s) (ldap\_search\_ext\_s(%x, %s, %s, %s))

Flow Stall Recovery Timer: to recover stall of %s on route from %s, recovery count = %d (フローストール回復タイマー: %s からのルートに対する %s のストールの回復、回復回数 = %d)

Error, filter '%s' contains an illegal type substitution character, only %s is allowed (エラー。フィルタ '%s' に無効なタイプの置換文字が含まれています。使用できるのは %s のみです)

Rendezvous Certified Advisory: %s (Rendezvous 保証アドバイザー: %s)

LDAP response resulting from checking if an entry is a member of a dynamic group: (エントリが動的グループのメンバーである場合、確認を行うと LDAP 応答が返されます:)

ignoring route '%s' at '%s', route user does not exist. ('%s' のルート '%s' を無視します。ルートユーザーが存在しません。)

Created %s transport '%s' (%s トランSPORT '%s' を作成しました)

Send recover request for routed queue flow stall for queue %s (キュー %s の、ルーティング先キューフローストールの回復要求を送信します)

Removing routed topic consumer '%s' (ルーティング先トピックコンシューマ '%s' を削除しています)

License has been activated. (ライセンスがアクティブ化されました。)

Hostname: %s (ホスト名: %s)

Evaluation Software Notice: remaining uptime is %d hours %d minutes. (評価版ソフトウェアに関する通知: 試用期間は残り %d 時間 %d 分です。)

[%s@%s]: rejected connect from route: implicit route already exists ([%s@%s]: ルートからの接続が拒否されました: 暗黙的なルートがすでに存在しています)

LDAP response resulting from getting attributes for group '%s': (グループ '%s' の属性を取得した結果、LDAP 応答が返されます:)

ldap\_parse\_reference: %s (ldap\_parse\_reference: %s)

Storage Location: '%s'. (ストレージの場所: '%s'。)

Search reference: %s (検索参照: %s)

Route Recover Interval is %u seconds. (ルート回復間隔は %u 秒です。)

Route Recover Minimum Message Count is %u. (ルート回復最小メッセージ数は %u です。)

Route connect error: route has no zone setting ( ルート接続エラー : ルートにゾーン設定がありません )

SS: Deleting existing GMD file. (SS : 既存の GMD ファイルを削除しています。)

LDAP error: %s (LDAP エラー : %s)

Clean all flow stalls for route to server %s: %s ( サーバー %s へのルートのすべてのフローストールをクリーンアップします : %s)

%s: shutdown server (%s : サーバーをシャットダウンします)

Reading configuration from '%s'. ('%s' から設定を読み取っています。)

Configuration warning: file=%s, line=%d: illegal to specify both '%s' and '%s', ignoring '%s' ( 設定に関する警告 : ファイル = %s、行 = %d : '%s' と '%s' を同時に指定することはできません。'%s' を無視します。)

Recovered flow stalled consumer for destination: %s:%s ( デスティネーションの、フローストールされたコンシューマを回復しました : %s : %s)

%s: revoked all %s permissions on %s '%s' (%s : %s '%s' のすべての %s アクセス権限を取り消しました)

Error sending routing information to '%s'. ('%s' へのルーティング情報の送信時にエラーが発生しました。)

Send recover request ( 回復要求を送信します)

Skipping recover request, message count % PRINTF\_LLFMT d greater than recover count ( 回復要求をスキップしています。メッセージ数 % PRINTF\_LLFMT d が回復数よりも多くなっています)

Lazy Dels: max=%d avg=%.2f min=%d ( 遅延削除 : 最大 = %d 平均 = %.2f 最小 = %d)

Release Holds: max=%d avg=%.2f min=%d ( リリースの保持 : 最大 = %d 平均 = %.2f 最小 = %d)

%s: created rvcmlistener transport '%s' name '%s' dest '%s' (%s : rvcmlistener トランスポート '%s'、名前 '%s'、デスティネーション '%s' を作成しました)

ERROR: file=%s, line=%d: %s is too long. ( エラー : ファイル = %s、行 = %d : %s が長すぎます。)

Route '%s' connecting to url '%s'. ( ルート '%s' は URL '%s' に接続しています。)

Route '%s' connected to url '%s' with zone '%s:%s'. ( ルート '%s' は URL '%s' にゾーン '%s : %s' で接続されました。)

Detected tie during route creation '%s'. ('%s' のルート作成中に関連付けが検出されました。 ) Resolution, keep existing route (connID=% PRINTF\_LLFMT d). ( 解決方法 : 既存のルート (接続 ID = % PRINTF\_LLFMT d) を保持します。)

Detected tie during route creation '%s'. ('%s' のルート作成中に関連付けが検出されました。) Resolution, destroy existing route (connID=% PRINTF\_LLFMt d). (解決方法: 既存のルート (接続 ID = % PRINTF\_LLFMt d) を破棄します。)

Found connID=%PRINTF\_LLFMtd for existing route '%s', but no connection. (既存のルート '%s' で 接続 ID = %PRINTF\_LLFMtd が見つかりましたが、接続されていません。)

[%s@%s]: rejected connect from route: %s ([%s@%s]: ルートからの接続が拒否されました: %s)

Configuration warning: file=%s, line=%d: Use of Rendezvous Bridge via tibrv\_... parameters has been deprecated. (設定に関する警告: ファイル = %s、行 = %d: tibrv\_... パラメータによる Rendezvous ブリッジの使用は廃止されました。) This feature is subject to removal in the next release of this product. (この機能は、この製品の次のリリースで削除されます。) Please convert your configuration to utilize transports defined in transports.conf configuration file. (transports.conf 設定ファイルで定義されたトランスポートを利用するように設定を変更してください。)

Rendezvous %s %s enabled (RV %s). (Rendezvous %s %s が有効になりました (RV %s)。)

Error in ldap\_search\_ext\_s: %s (ldap\_search\_ext\_s でエラーが発生しました: %s)

Server is re-entering standby mode. (サーバーは再度スタンバイモードに入ります。)

Statistics database memory now below limit (統計情報データベースメモリが現在制限を下回っています)

SS: Destroying SmartSockets transport %s (SS: SmartSockets トランスポート %s を破棄しています)

Created file '%s' (ファイル '%s' を作成しました)

Restored routed topic consumer for '%s' ('%s' のルーティング先トピックコンシューマを復元しました)

Adding routed topic consumer for '%s' ('%s' のルーティング先トピックコンシューマを追加しています)

Subscriber %s for topic '%s' exceeded topic limit. (トピック '%s' のサブスクライバ %s がトピックの制限を超過しました。)

Refrained from removing configured durable '%s' (設定された持続 '%s' の削除は行われませんでした)

Sync Recs: max=%d avg=%.2f min=%d (非同期レコード: 最大 = %d 平均 = %.2f 最小 = %d)

SS: Unsubscribe from '%s' tport = %s (SS : '%s' トランSPORT = %s からサブスクリプションを解除します)

Recovered %d pending connection%s. (%d の保留中の接続 %s を回復しました。)

SS: Imported message on tport='%s', subject='%s', reply='%s'. (SS : トランSPORT = '%s'、サブジェクト = '%s'、応答 = '%s' でメッセージがインポートされました。)

Clean flow stall for consumers of destination %s:%s ( デスティネーション %s のコンシューマのフローストールをクリーンアップします : %s)

ldap\_search\_s(%x, %s, %s, %s, [NULL]) (ldap\_search\_s(%x, %s, %s, %s, [NULL]))

%s:%s queue browser failed: illegal to use wildcard queue [%s] (%s : %s キューブラウザが失敗しました : ワイルドカードキュー [%s] を使用することはできません)

There should be only one consumer reaching %s, but %d found (%s に到達するコンシューマはひとつのみである必要がありますが、%d 個のコンシューマが見つかりました)

%s:%s queue browser failed: cannot browse [%s] because it is a routed queue. (%s : %s キューブラウザが失敗しました : ルーティング先キューであるため [%s] は参照できません。)

Clear (Non-IO) flow stalled on dest %s:%s from route of %s ( デスティネーション %s で停滞している (非 IO) フローをクリアします : %s のルートからの %s)

Error sending routing information to %s, send failed ('%s' へのルーティング情報の送信時にエラーが発生しました。送信に失敗しました)

%s: %s updated: '%s' (%s : %s が更新されました : '%s')

%s: consumed\_msg\_hold\_time updated: '%d' (%s : consumed\_msg\_hold\_time が更新されました : '%d')

Authorization is disabled. ( 認可が無効になっています。)

SSL connect: using certificate username '%s'. (SSL 接続 : 証明書のユーザー名 '%s' を使用しています。)

SSL reset to TCP for connID=% PRINTF\_LLFFMT d, user='%s' (SSL は 接続 ID = % PRINTF\_LLFFMT d、ユーザー = '%s' の TCP にリセットされました)

Configuration warning: file=%s, line=%d: invalid trace option '%s' is ignored ( 設定に関する警告 : ファイル = %s、行 = %d : 無効なトレースオプション '%s' は無視されます)

Server is now active. ( サーバーは現在アクティブです。)

(NON-IO) Flow stalled on dest %s from route of %s ( (非 IO) フローが %s のルートからのデスティネーション %s で停滞しています)

Dump of user cache: ( ユーザーキャッシュのダンプ : )

Administrator group not found, created with default member. ( 管理者グループが見つかりません。デフォルトのメンバーで作成されました。 )

Received exception on route '%s': '%s' ( ルート '%s' で例外を受け取りました : '%s' )

ldap\_search\_s(%x, %s, %s, %s, [%s,%s,%s,%s,%s, NULL]) (ldap\_search\_s(%x, %s, %s, %s, [%s,%s,%s,%s,%s, NULL]))

EXPIRE: msg=%d ( 有効期限 : メッセージ = %d )

Clean flow stall for routed consumer of queue %s ( キュー %s のルーティング先コンシューマのフローストールをクリーンアップします )

EXPIRE: total=% PRINTF\_LLFFMT d expire=0 ( 有効期限 : 合計 = % PRINTF\_LLFFMT d、有効期限 = 0 )

EXITING ( 終了します )

Configuration error: file=%s, line=%d: ignoring invalid selector specifications in route parameters ( 設定エラー : ファイル = %s、行 = %d : ルートパラメータ内の無効なセクタ指定を無視します )

%s: created group '%s' (%s : グループ '%s' を作成しました )

set %s:% PRINTF\_LLFFMT d in flow stall recover request (%s を設定します : フローストール回復要求の % PRINTF\_LLFFMT d)

Error: unable to bind to LDAP server as: '%s', %s ( エラー : LDAP サーバーに '%s'、%s としてバインドできません )

DISK IO stats for %s: (%s のディスク IO 統計情報 : )

Authorization exception creating routed topic consumer for '%s' ('%s' のルーティング先トピックコンシューマの作成中に認可の例外が発生しました )

Fault tolerant reconnect timeout is set to a negative or 0 value of %d seconds, ( フォールトトレランス再接続タイムアウトが %d 秒の負の値またはゼロに設定されています。 )

Licensed server is waiting for license activation. ( ライセンスサーバーがライセンスのアクティブ化を待機しています。 )

LDAP message resulting from checking existence: ( 存在の確認による LDAP メッセージ : )

Memory limit of %d MB exceeded. (%dMB のメモリ制限を超過しました。 )

%s %s to %s: connID=% PRINTF\_LLFFMT d consID=% PRINTF\_LLFFMT d msgID='%s' %s='%s'%s%s%s%s%s%s (%s %s から %s : 接続 ID = % PRINTF\_LLFFMT d、コンシューマ ID = % PRINTF\_LLFFMT d、メッセージ ID = '%s'、%s = '%s'%s%s%s%s%s%s)

User '%s' is authenticated via LDAP ( ユーザー '%s' は LDAP 経由で認証されました )

User '%s' is authenticated via JAAS ( ユーザー '%s' は JAAS 経由で認証されました )

Route '%s' accepted from host '%s' with zone '%s:%s'. ( ルート '%s' はゾーン '%s : %s' でホスト '%s' に受け入れられました。 )

%s:%s queue browser failed: queue does not exist: [%s] (%s : %s キューブラウザが失敗しました : キューが存在しません : [%s])

%s acknowledged by %s: connID=% PRINTF\_LLFORMAT d consID=% PRINTF\_LLFORMAT d msgID='%s' %s='%s' (%s が %s で確認応答されました : 接続 ID = % PRINTF\_LLFORMAT d、コンシューマ ID = % PRINTF\_LLFORMAT d、メッセージ ID = '%s'、%s = '%s')

%s premature exit: %s : connID=% PRINTF\_LLFORMAT d prodID=% PRINTF\_LLFORMAT d msgID='%s' %s='%s' (%s の途中終了 : %s : 接続 ID = % PRINTF\_LLFORMAT d、プロデューサ ID = % PRINTF\_LLFORMAT d、メッセージ ID = '%s'、%s = '%s')

%s: removed user '%s' from group '%s' (%s : ユーザー '%s' をグループ '%s' から削除しました )

Reading SS configuration from '%s'. ('%s' から SS 設定を読み取っています。 )

Ignoring inbound routed topic '%s', illegal topic. ( インバウンドルーティング先トピック '%s' を無視します。不正なトピックです。 )

%s: Compacting %s with no time limit. (%s : 制限時間なしで %s を圧縮しています。 )

STARTING POP WAITING (POP の起動を待機しています )

Flow stall recover ack received Post IO ( フローストール回復確認応答を受信しました。IO をポストします )

RVCN name not specified for transport '%s', using RVCN name '%s' (RVCN 名がトランスポート '%s' に指定されていません。RVCN 名 '%s' を使用します )

Purged % PRINTF\_LLFORMAT d queue message%s. (% PRINTF\_LLFORMAT d キューメッセージ %s がパージされました。 )

Purged % PRINTF\_LLFORMAT d topic consumer message%s. (% PRINTF\_LLFORMAT d トピックコンシューマメッセージ %s がパージされました。 )

No memory to process incoming data from connection id=% PRINTF\_LLFORMAT d. Connection terminated. ( 接続 ID = % PRINTF\_LLFORMAT d からの受信データを処理するメモリがありません。接続が切断されました。 )

%s: Disconnected, connection id=% PRINTF\_LLFORMAT d, reason: %s%s (%s : 切断されました。接続 ID = % PRINTF\_LLFORMAT d、理由 : %s%s)



%s: connection id=% PRINTF\_LLFMT d purged after FT timeout (%s : 接続 ID = % PRINTF\_LLFMT d が FT タイムアウト後にページされました)

Error, missing %s parameter (エラー : %s パラメータが指定されていません)

Bytes: max=%d avg=%.2f min=%d (バイト : 最大 = %d 平均 = %.2f 最小 = %d)

Server is active. (サーバーはアクティブです。)

%s: %s bridge: source=[%s:%s] target=[%s:%s] (%s : %s ブリッジ : ソース = [%s : %s] ターゲット = [%s : %s])

Error, filter '%s' contains too many occurrences of %%%s, max allowed is: %d (エラー : フィルタ '%s' に使用されている %%%s の数が多過ぎます。最大許容数 : %d)

%s: created JNDI name '%s' for %s '%s' (%s : %s '%s' の JNDI 名 '%s' を作成しました)

Server is in standby mode. (サーバーがスタンバイモードです。)

%s: create %s failed: durable access denied for %s [%s]. (%s : %s の作成に失敗しました : %s [%s] に対する持続アクセス権限が拒否されました。)

%s: Destroyed producer (connid=% PRINTF\_LLFMT d, sessid=% PRINTF\_LLFMT d, prodid=% PRINTF\_LLFMT d) %s (%s : プロデューサ (接続 ID = % PRINTF\_LLFMT d、セッション ID = % PRINTF\_LLFMT d、プロデューサ ID = % PRINTF\_LLFMT d) %s を破棄しました)

ldap\_simple\_bind\_s(, \*\*\*\*\*) (ldap\_simple\_bind\_s(, \*\*\*\*\*))

[%s] %s ([%s] %s)

Active server '%s' not found. (アクティブサーバー '%s' が見つかりません。)

Backup server '%s' has connected. (バックアップサーバー '%s' が接続されました。)

Error in ldap\_set\_option: %s (ldap\_set\_option でエラーが発生しました : %s)

%s:%s queue browser failed: access denied for queue [%s] (%s : %s キューブラウザが失敗しました : [%s] に対するアクセス権限が拒否されました)

Ignoring inbound routed topic '%s', no corresponding topic. (インバウンドルーティング先トピック '%s' を無視します。対応するトピックがありません。)

%s: created user '%s' (%s : ユーザー '%s' を作成しました)

Unable to initialize route: expected route name '%s', received '%s'. (ルートを初期化できません : 必要なルート名は '%s' です。'%s' を受信しました。)

Evaluation Software Notice: remaining uptime is %d minutes. (評価版ソフトウェアに関する通知 : 試用期間は残り %d 分です。)

%s: created topic '%s'%s%s (%s : トピック '%s'%s%s を作成しました)

Connected to LDAP server %s (LDAP サーバー %s に接続しました)

Processed %d msgs (%d 件のメッセージを処理しました)

Error, LDAP is disabled (エラー: LDAP が無効になっています)

Configuration warning: file=%s, line=%d: illegal to use '.' in server name, replaced with '\_', (設定に関する警告: ファイル = %s、行 = %d: サーバー名には '.' を使用できません。 '.' と置き換えられました。)

%s: %s %s '%s' administrative permissions: %s (%s: %s %s '%s' 管理アクセス権限: %s)

Warning: statistics database memory exceeded limit (警告: 統計情報データベースメモリが制限を超過しました)

[%s@%s]: rejected connect from route: this shouldn't happen: route exists with no zone setting ([%s@%s]: ルートからの接続が拒否されました: これは予期しない動作です: ルートが存在しますがゾーン設定がありません)

Rejected connect from route '%s' at %s, routing disabled. (%s でルート '%s' からの接続が拒否されました。ルーティングは無効です。)

Missing heartbeats from primary server '%s'. (プライマリサーバー '%s' にハートビートが指定されていません。)

Flow stall recovery request received, send to IO (フローストール回復要求を受信しました。IO に送信します)

Unable to initialize route '%s': route server returned: '%s' (ルート '%s' を初期化できません: ルートサーバーが '%s' を返しました)

RUNNING SWAPPER %d %d needed = %u! (スワップ %d %d の実行に %u! が必要です)

Restoring consumer warning: zone of id %d does not exist in zone mapping entries (コンシューマの復元に関する警告: ID %d のゾーンがゾーンマッピングエントリ内に存在しません)

Ignoring inbound routed queue '%s', no corresponding queue. (インバウンドルーティング先キュー '%s' を無視します。対応するキューがありません。)

%s: %s queue browser failed: illegal to use reserved queue [%s] (%s: %s キューブラウザが失敗しました: 予約済みのキュー [%s] を使用することはできません)

%s: committed transaction %s (%s: トランザクション %s がコミットされました)

Trying to send flow stall recovery request for destination: %s (デスティネーションのフローストール回復要求を送信しようとしています: %s)

%s: destroyed connection % PRINTF\_LL\_FMT d (%s: 接続 % PRINTF\_LL\_FMT d が破棄されました)

ldap\_search\_s(%x, %s, %s, %s, [%s, %s, %s, NULL]) (ldap\_search\_s(%x, %s, %s, %s, [%s, %s, %s, NULL]))

Allocating storage to minimum %s for store '%s', please wait. ( ストレージにストア '%s' の最小容量 %s を割り当てています。しばらくお待ちください。 )

%s:%s queue browser failed: invalid name of queue [%s] (%s : %s キューブラウザが失敗しました : キュー [%s] の名前が無効です )

%s: updated group '%s' (%s : グループ '%s' を更新しました )

SS: Created subscriber to '%s' LB override=%d. tport='%s' (SS : '%s' LB 無効 = %d に対するサブスクライバを作成しました。トランスポート = '%s')

%s: destroyed message '%s' (%s : メッセージ '%s' を破棄しました )

Configuration warning: file=%s, line=%d: Use of Rendezvous import/export settings via tibrv\_... parameters has been deprecated. ( 設定に関する警告 : ファイル = %s、行 = %d : tibrv\_... パラメータによる Rendezvous のインポート / エクスポート設定の使用は廃止されました。 ) This feature is subject to removal in the next release of this product. ( この機能は、この製品の次のリリースで削除されます。 ) Please convert your configuration to utilize 'import' and 'export' properties and using transports defined in transports.conf configuration file. ('import' および 'export' プロパティを利用し、transports.conf 設定ファイルで定義されたトランスポートを使用するように設定を変更してください。 )

Route '%s' disconnected, connection id=% PRINTF\_LLFMT d ( ルート '%s' が切断されました。接続 ID = % PRINTF\_LLFMT d )

Routed Queue '%s' is not a home Queue ( ルーティング先キュー '%s' はホームキューではありません )

Logging into file '%s' ( ファイル '%s' にログインしています )

%s: create %s failed: access denied for %s [%s]. (%s : %s の作成に失敗しました : %s [%s] に対するアクセス権限が拒否されました。 )

Unable to obtain message type number for imported SS message ( インポートされた SS メッセージのメッセージタイプ番号を取得できません )

Route Warning: host of this name does not exist: %s ( ルートに関する警告 : この名前のホストは存在しません : %s )

%s: created queue '%s'%s%s (%s : キュー '%s'%s%s を作成しました )

now timer fired ( タイマーが起動しました )

Clock sync timer created with interval %d seconds ( クロック同期タイマーが %d 秒の間隔で作成されました )

Clock sync timer fired ( クロック同期タイマーが起動しました )

Clock sync timer error: %s ( クロック同期タイマーエラー : %s )

Breaking from remove thread for wantsLock! (wantsLock の削除スレッドから離脱しています )

%s: created JNDI name '%s' = '%s' (%s : JNDI 名 '%s' = '%s' を作成しました)

Metadata storage: '%s'. (メタデータストレージ: '%s')。)

Flow stall recover ack received for destination %s (デスティネーション %s でフローストール回復確認応答を受信しました)

Server rereading configuration. (サーバーが設定を再度読み取っています。)

Route '%s' sent resume request (ルート '%s' が再開要求を送信しました)

Refrained from deleting configured durable '%s' even though application's attributes differ from configuration (アプリケーションの属性は設定とは異なりますが、設定された持続 '%s' の削除は行われませんでした)

EXPIRE: msgs=% PRINTF\_LLFMT d exp=% PRINTF\_LLFMT d expd=% PRINTF\_LLFMT d int=% PRINTF\_LLFMT d tm=% PRINTF\_LLFMT d (有効期限: メッセージ = % PRINTF\_LLFMT d、有効期限 = % PRINTF\_LLFMT d、期限切れ = % PRINTF\_LLFMT d、間隔 = % PRINTF\_LLFMT d、tm=% PRINTF\_LLFMT d)

Server of version %d.%d does not support flow stall recovery, do nothing. (バージョン %d.%d のサーバーではフローストール回復はサポートされていません。何も行いません。)

%s: rotated log file. (%s : ログファイルがローテーションされました)

Asynchronous storage: '%s'. (非同期ストレージ: '%s')。)

Created Routed Dynamic Queue '%s' from '%s' ('%s' から動的なルーティング先キュー '%s' を作成しました)

Results of searching for dynamic groups: (動的グループの検索結果:)

Transaction for non-existent consumer: % PRINTF\_LLFMT d connID=% PRINTF\_LLFMT d sessID=% PRINTF\_LLFMT d %s (存在しないコンシューマに対するトランザクション: % PRINTF\_LLFMT d、接続 ID = % PRINTF\_LLFMT d、セッション ID = % PRINTF\_LLFMT d %s)

Error in ldap\_unbind: %s (ldap\_unbind でエラーが発生しました: %s)

Rendezvous %s %s enabled. (Rendezvous %s %s が有効になりました。)

[%s@%s]: route connect failed: invalid password ([%s@%s]: ルート接続に失敗しました: パスワードが無効です)

%s: updated topic '%s': %s (%s : トピック '%s' を更新しました: %s)

ldap\_search\_s(%x, %s, %s, %s, [%s, NULL]) (ldap\_search\_s(%x, %s, %s, %s, [%s, NULL]))

Configuration warning: file=%s, line=%d: invalid option '%s' is ignored (設定に関する警告: ファイル = %s、行 = %d: 無効なオプション '%s' は無視されます)

Server is in standby mode for '%s'. ('%s' でサーバーがスタンバイモードです。)

Error, references not supported (エラー：参照はサポートされていません)

Acknowledging the flow stall recover request for destination %s:%s and resume the flow ( デスティネーション %s のフローストール回復要求に確認応答しています：%s を行いフローを再開します )

Failed to rename file, original file %s saved as file %s. Please rename the file ( 名前を変更できませんでした。元のファイル %s はファイル %s として保存されました。ファイルの名前を変更してください )

Route connect error: can not connect to route '%s' at '%s', error=%s. ( ルート接続エラー：'%s' でルート '%s' に接続できません。エラー = %s。 )

Recovered %d message%s. (%d 件のメッセージ %s を回復しました。 )

%s: deleted group '%s' (%s：グループ '%s' を削除しました )

Start opening sync db ( 同期データベースを開きます )

Start opening async db ( 非同期データベースを開きます )

Removed Routed Dynamic Queue '%s' ( 動的なルーティング先キュー '%s' を削除しました )

%s: %s bridge: source=[%s:%s] target=[%s:%s] selector='%s' (%s：%s ブリッジ：ソース = [%s：%s] ターゲット = [%s：%s] セレクタ = '%s' )

Error, zero entries returned from getting attributes for group '%s': ( エラー：グループ '%s' の属性の取得によって返されたエントリはありません。 )

Warning: configuration file 'tibjmsd.conf' should be renamed to 'tibemsd.conf'. ( 警告：設定ファイル 'tibjmsd.conf' の名前を 'tibemsd.conf' に変更する必要があります。 )

Warning: [queue: %s]: dynamic routed queues are not supported. ( 警告：[キュー：%s]：動的なルーティング先キューはサポートされていません。 )

Transaction for non-existent message: % PRINTF\_LLFMT d connID=% PRINTF\_LLFMT d sessID=% PRINTF\_LLFMT d %s ( 存在しないメッセージに対するトランザクション：% PRINTF\_LLFMT d 接続 ID = % PRINTF\_LLFMT d、セッション ID = % PRINTF\_LLFMT d %s )

Route recovery of destination %s on route from %s failed, will try again: %s (%s からのルートにおけるデスティネーション %s のルート回復に失敗しました。再試行します：%s )

[%s@%s]: route connect failed: route server not authorized. ([%s@%s]：ルート接続に失敗しました：ルートサーバーは権限を持っていません。 )

Implicit route to [%s] already exists ([%s] への暗黙的なルートがすでに存在しています )

%s: %s route '%s', URL=[%s] (%s : %s ルート '%s'、URL = [%s])

Results of searching for static groups: ( 静的グループの検索結果 : )

%s: Queue limit exceeded for queue '%s'. (%s : キュー '%s' のキューの制限を超過しました。)

%s: Topic limit exceeded for topic '%s'. (%s : トピック '%s' のトピックの制限を超過しました。)

Implicit route to '%s' already exists. ('%s' への暗黙的なルートがすでに存在しています。)

%s: deleted rvcmlistener transport '%s' name '%s' dest '%s' (%s : rvcmlistener トランスポート '%s'、名前 '%s'、デスティネーション '%s' を削除しました)

Evaluation Software Notice: remaining uptime is %d hours. ( 評価版ソフトウェアに関する通知 : 試用期間は残り %d 時間です。)

Secure Socket Layer is enabled, using %s (Secure Socket Layer が有効になっています。 %s を使用しています)

Using cryptographic module %s ( 暗号化モジュール %s を使用しています)

Reserve memory reestablished, all client requests accepted. ( 予約メモリが再確立されました。すべてのクライアント要求を受け入れました。 ) Pending msg count = % PRINTF\_LLFMT d ( 保留中のメッセージ数 = % PRINTF\_LLFMT d)

Msg recs processed = %d ( 処理されたメッセージレコード数 = %d)

ldap\_search\_s(%x, %s, %s, %s, [%s,%s,NULL]) (ldap\_search\_s(%x, %s, %s, %s, [%s,%s,NULL]))

(IO) Flow stalled on dest %s:%s from route of %s ( (IO) フローがデスティネーション %s で停滞しています : %s のルートからの %s)

Invalid specifications for route '%s' topic '%s' ( ルート '%s'、トピック '%s' の指定は無効です)

%s: Created producer (connid=% PRINTF\_LLFMT d, sessid=% PRINTF\_LLFMT d, prodid=% PRINTF\_LLFMT d) %s (%s : プロデューサ (接続 ID = % PRINTF\_LLFMT d、セッション ID = % PRINTF\_LLFMT d、プロデューサ ID = % PRINTF\_LLFMT d) %s を作成しました)

SS: Consumer subscribe to '%s' LB override=%d. tport='%s' (SS : コンシューマは '%s' LB 無効 = %d をサブスクライブしています。トランスポート = '%s')

Routing is enabled. ( ルーティングが有効になっています。)

Recovering state, please wait. ( ステートを回復しています。しばらくお待ちください。)

Files opened. ( ファイルが開かれました。)

Starting msgPass. (msgPass を開始しています。)

Finished msgPass. (msgPass を終了しました。)

Administrator user not found, created with default password. ( 管理者ユーザーが見つかりません。デフォルトのパスワードで作成されました。)

Route '%s' sent suspend request ( ルート '%s' が一時停止要求を送信しました )

%d batches, %.2f batches/sec (%d バッチ、%.2f バッチ / 秒)

%s: purged queue '%s' (%s : キュー '%s' がパージされました)

Error in ldap\_search\_s: %s (ldap\_search\_s でエラーが発生しました : %s)

%s: created durable '%s' Selector: %s (%s : 持続 '%s' セレクタを作成しました : %s)

Accepted license with limits: conns=%d hosts=%d hours=%d ( 制限のあるライセンスに同意しました : 接続 = %d、ホスト = %d、時間 = %d)

USING %d memory (%d メモリを使用しています)

Continuing as active server. ( アクティブサーバーとして続行します。)

VALIDATING STORE %s ( ストア %s を検証しています)

[%s@%s]: rejected connect from route: route already connected ([%s@%s] : ルートからの接続が拒否されました : ルートがすでに存在しています)

%s: purged topic '%s' (%s : トピック '%s' がパージされました)

ldap\_search\_ext: %s (ldap\_search\_ext : %s)

Flow control %s on %s '%s' (%s '%s' でのフロー制御 %s)

Accepting connections on %s. (%s で接続を受け入れています)

Configuration warning: ignoring tibrvcm\_import property set on %s '%s' because it collides with tibrvcm\_import property on %s '%s' ( 設定に関する警告 : %s '%s' で設定された tibrvcm\_import プロパティは %s '%s' の tibrvcm\_import プロパティと競合するため無視します)

Warning: configuration file '%s' not found and has been created. ( 警告 : 設定ファイル '%s' が見つからないため作成されました。 ) All configuration settings have been reset to defaults. ( すべての設定はデフォルトにリセットされました。 )

EXPIRE ERROR: oldCount=% PRINTF\_LLFMT d found=% PRINTF\_LLFMT d walked=% PRINTF\_LLFMT d ( 有効期限エラー : 旧カウント = % PRINTF\_LLFMT d、検出済み = % PRINTF\_LLFMT d、ウォーク済み = % PRINTF\_LLFMT d)

Shutdown complete. ( シャットダウンが完了しました。)

Restarting now. ( 再起動しています。)



%s: Created %s consumer %s (connid=% PRINTF\_LLFMT d, sessid=% PRINTF\_LLFMT d, consid=% PRINTF\_LLFMT d) on %s '%s' %s %s %s (%s : %s '%s' %s %s の %s %s コンシューマ %s %s %s (接続 ID = % PRINTF\_LLFMT d、セッション ID = % PRINTF\_LLFMT d、コンシューマ ID = % PRINTF\_LLFMT d) が作成されました)

Search completed successfully. (検索は正常に完了しました。) Entries found: %d (見つかったエントリ数 : %d)

Created routed dynamic queue '%s' for '%s' ('%s' の動的なルーティング先キュー '%s' を作成しました)

%s: deleted durable '%s' (%s : 持続 '%s' を削除しました)

Part of the DN that matches an existing entry: %s (既存のエントリに一致する DN の一部 : %s)

Purged %d connection %s. (%d 件の接続 %s がページされました。)

Ignoring inbound routed topic '%s', local topic is not global. (インバウンドルーティング先トピック '%s' を無視します。ローカルトピックはグローバルではありません。)

%s: deleted user '%s' (%s : ユーザー '%s' を削除しました)

%s: create %s failed: access denied for monitoring %s [%s]. (%s : %s の作成に失敗しました : %s [%s] の監視に対するアクセス権限が拒否されました。)

Administrator group found with no members, added default member. (管理者グループのメンバーが見つかりません。デフォルトのメンバーを追加しました。)

%s: updated user '%s' (%s : ユーザー '%s' を更新しました)

SmartSockets transports are enabled. (SmartSockets トランスポートが有効になっています。)

Running in Temporary Destination Compliance mode. (テンポラリデスティネーション準拠モードで実行しています。)

ldap\_search\_s(%x, %s, %d, %s, [%s, NULL]) (ldap\_search\_s(%x, %s, %d, %s, [%s, NULL]))

%s: %s created dynamic %s '%s' (%s : %s は動的 %s '%s' を作成しました)

Using %d threads for LDAP processing. (LDAP 処理に %d スレッドを使用しています。)

Routing is disabled. (ルーティングが無効になっています。)

%s: %s factory '%s' (%s : %s ファクトリ '%s')

INIT-EXPIRE: exp=%d mexp=% PRINTF\_LLFMT d oldt=% PRINTF\_LLFMT d interval=%d (INIT-EXPIRE : 有効期限 = %d、mexp = % PRINTF\_LLFMT d、oldt = % PRINTF\_LLFMT d、間隔 = %d)



Error in ldap\_initialize(%s) (ldap\_initialize(%s) でエラーが発生しました )

Created dynamic %s '%s' ( 動的 %s '%s' を作成しました )

%s: Compacting %s with time limit %PRINTE\_LLFMtd seconds (%s :  
%PRINTE\_LLFMtd 秒の制限時間で %s を圧縮しています )

Error sending route query to '%s'. ('%s' へのルートクエリの送信時にエラーが発生しました。 )

%s: updated queue '%s': %s (%s : キュー '%s' を更新しました : %s)

Server name: '%s'. ( サーバー名 : '%s'。 )

Route configuration error: global queue '%s' from route '%s' collides, global queues must be unique. ( ルート設定エラー : ルート '%s' からのグローバルキュー '%s' が競合しています。グローバルキューは固有である必要があります。 )

Route configuration error: routed queue '%s' from route '%s' is not global in '%s'. ( ルート設定エラー : ルート '%s' からのルーティング先キュー '%s' は '%s' でグローバルではありません。 )

Server shutting down. ( サーバーをシャットダウンしています。 )

Server preparing to restart. ( サーバーの再起動を準備しています。 )

Route configuration warning: global queue '%s' from route '%s' not configured on local server ( ルート設定に関する警告 : ルート '%s' からのグローバルキュー '%s' はローカルサーバーで設定されていません。 )

Flow stall recovery request received, recover consumer of id % PRINTE\_LLFMtd ( フローストール回復要求を受信しました。ID % PRINTE\_LLFMtd のコンシューマを回復します )

Flow Control is enabled. ( フロー制御が有効になっています。 )

%s: deleted JNDI name '%s' (%s : JNDI 名 '%s' を検出しました )

Clear (IO) flow stalled on dest %s:%s from route of %s ( デスティネーション %s で停滞している (IO) フローをクリアします : %s のルートからの %s )

Removing route '%s', URL='%s': this route is duplicate, creates a loop or has configuration errors ( ルート '%s'、URL = '%s' を削除しています : このルートは重複しています。ループを作成しない場合は設定エラーになります )

EXPIRE: giving up amid lock ( 有効期限 : ロック状態を停止します )

Done opening async db ( 非同期データベースが開かれました )

Error, invalid search scope: %s ( エラー。無効な検索範囲 : %s )

Error in ldap\_url\_parse, returned: %d (ldap\_url\_parse でエラーが発生しました。戻り値 : %d)

%s: create %s failed: durable recreation access denied for %s [%s]. (%s : %s の作成に失敗しました : %s [%s] に対する持続再作成アクセス権限が拒否されました。)

Ignoring inbound routed queue '%s', illegal queue. (インバウンドルーティング先キュー '%s' を無視します。キューが無効です。)

key: '%s' value: '%s' (キー : '%s' 値 : '%s')

Connection to primary server '%s' has been lost. (プライマリサーバー '%s' への接続が失われました。)

Route connect error: failed connect to server '%s' at '%s' (ルート接続エラー : '%s' でサーバー '%s' に接続できませんでした)

%s: rolled back transaction %s (%s : トランザクション %s がロールバックされました)

LDAP response resulting from checking existence: (存在の確認による LDAP 応答 : )

ldap\_parse\_result: %s (ldap\_parse\_result : %s)

Hostname IP address: %s (ホスト名 IP アドレス : %s)

Flow stall recovery request received, after IO (フローストール回復要求を IO の後に受信しました)

%s: Connected, connection id=% PRINTF\_LL\_FMT d, type: %s%s (%s : 接続されました。接続 ID = % PRINTF\_LL\_FMT d、タイプ : %s%s)

%s: Reconnected, connection id=% PRINTF\_LL\_FMT d, type: %s%s (%s : 再接続されました。接続 ID = % PRINTF\_LL\_FMT d、タイプ : %s%s)

Unable to initialize one hop route: One-hop routing is not supported for server version %d.%d (1 ホップルートを初期化できません : 1 ホップルーティングはサーバーバージョン %d.%d ではサポートされていません)

Error, must provide static and/or dynamic group name attribute (エラー。静的か動的またはその両方のグループ名属性を指定する必要があります)

Unable to initialize route, expected route name not received. (ルートを初期化できません。必要なルート名を受け取りませんでした。)

Stat: rate=%d cleanup=%d memory=%d (統計情報 : 転送率 = %d、クリーンアップ = %d、メモリ = %d)

%s: Destroyed %sconsumer%s%s%s (connid=% PRINTF\_LL\_FMT d, sessid=% PRINTF\_LL\_FMT d, consid=% PRINTF\_LL\_FMT d) on %s '%s' (%s : %s '%s' の %s コンシューマ %s%s%s (接続 ID = % PRINTF\_LL\_FMT d、セッション ID = % PRINTF\_LL\_FMT d、コンシューマ ID = % PRINTF\_LL\_FMT d) が破棄されました)

%s: Unsubscribed durable consumer '%s' due to administrator deleting durable (consid=% PRINTF\_LL\_FMT d) on topic '%s' (%s : 管理者がトピック '%s' の持続 (コンシューマ ID = % PRINTF\_LL\_FMT d) を削除したため、持続コンシューマ '%s' のサブスクライブが解除されました)

%s: Unsubscribed durable consumer '%s' due to user calling unsubscribe (connid=% PRINTF\_LL\_FMT d, consid=% PRINTF\_LL\_FMT d) on topic '%s' (%s : ユーザーがトピック '%s' のサブスクライブの解除 (接続 ID = % PRINTF\_LL\_FMT d、コンシューマ ID = % PRINTF\_LL\_FMT d) を呼び出したため、持続コンシューマ '%s' のサブスクライブが解除されました)

%s %s from %s: connID=% PRINTF\_LL\_FMT d prodID=% PRINTF\_LL\_FMT d msgID='%s' %s mode=%s size=%d %s='%s'%s%s (%s から %s %s : 接続 ID = % PRINTF\_LL\_FMT d、プロデューサ ID = % PRINTF\_LL\_FMT d、メッセージ ID = '%s'、%s モード = %s、サイズ = %d、%s = '%s'%s%s)

ldap\_search\_s(%x, %s, %s, %s, [%s,%s,%s,%s,NULL]) (ldap\_search\_s(%x, %s, %s, %s, [%s,%s,%s,%s,NULL]))

ldap\_search\_s: %s (ldap\_search\_s : %s)

Error, filter '%s' too long, max length is %d characters (エラー: フィルタ '%s' が長過ぎます。最大長は %d 文字です)

%s: deleted %s '%s' (%s : %s '%s' を削除しました)

Disallowing Rendezvous Certified Message listener '%s' (Rendezvous 保証メッセージリスナー '%s' の許可を取り消しています)

SS: Exporting EMS message: tport='%s', dest='%s', reply='%s' SSDelivery=%d, LB=%d (SS : EMS メッセージをエクスポートしています: トランSPORT = '%s'、デスティネーション = '%s'、応答 = '%s' SS 配信 = %d、LB=%d)

Refrained from removing configured route durable '%s' (設定されたルートの持続 '%s' の削除は行われませんでした)

Authorization is enabled. (認可が有効になっています。)

Rendezvous Advisory: %s (Rendezvous アドバイザリ : %s)

Refrained from removing durable for route '%s' due to configured durable (設定された持続が原因でルート '%s' の持続の削除は行われませんでした)

Configuration error: file=%s, line=%d: invalid value of option '%s'. (設定エラー: ファイル = %s、行 = %d: オプション '%s' の値が無効です。) Unable to start. (起動できません。)

%s: %sconnect failed: %s%s (%s : %s 接続に失敗しました : %s%s)

unable to create connection with existing ID % PRINTF\_LL\_FMT d (既存の ID % PRINTF\_LL\_FMT d との接続を作成できません)

Synchronous storage: '%s'. (同期ストレージ : '%s'。)

Clean all flow stalls for destination %s ( デスティネーション %s のすべてのフローストールをクリーンアップします )

Done opening sync db ( 同期データベースが開かれました )

%s: added user '%s' to group '%s' (%s : ユーザー '%s' をグループ '%s' に追加しました )

ldap\_search\_s(%x, %s, %s, %s, [%s,%s,%s,%s,%s,%s,NULL]) (ldap\_search\_s(%x, %s, %s, %s, [%s,%s,%s,%s,%s,%s,NULL]))

Configuration warning: file=%s, line=%d: can not specify 'NONE' with other options ( 設定に関する警告 : ファイル = %s、行 = %d : その他のオプションで 'NONE' を指定することはできません )

%s: %s failed: access denied for %s [%s]. (%s : %s に失敗しました : %s [%s] に対するアクセス権限が拒否されました。)

Process started from '%s'. ( プロセスが '%s' から開始されました。)

Clean flow stall for routed consumer of queue %s: no other remote consumers, remove the stall ( キュー %s のルーティング先コンシューマのフローストールをクリーンアップします : 他のリモートコンシューマはありません。ストールを削除します。)

[%s@%s]: connect failed: reached maximum number of %s %d ([%s@%s] : 接続に失敗しました : %s %d の最大数に達しました )

Refrained from removing durable for route '%s' due to configured durable ( 設定された持続が原因でルート '%s' の持続の削除は行われませんでした )

LDAP Cache: User '%s' is a member of group(s): (LDAP キャッシュ : ユーザー '%s' は次のグループのメンバーです : )

LDAP Cache: Deleting cached record of user: '%s' (LDAP キャッシュ : 次のユーザーのキャッシュ済みレコードを削除しています : '%s')

Client ID is too long. ( クライアント ID が長すぎます。)

Durable name is too long. ( 持続名が長すぎます。)

Durable name must be specified (connID=% PRINTF\_LL\_FMT d) ( 持続名を指定する必要があります (接続 ID = % PRINTF\_LL\_FMT d) )

Consumed Msg Hold Time is %d seconds. ( コンシュームされたメッセージの保持時間は %d 秒です。)

A duplicate durable instance (conn=%s durable=%s dest=%s) was detected, discarding old one ( 重複持続インスタンス (接続 = %s 持続 = %s デスティネーション = %s) が検出されたため、古いものを破棄しています)

A duplicate connection with same client id (clientid=%s) detected, destroying old conn (conn-id=% PRINTF\_LLFMT d) ( 同じクライアント ID (クライアント ID = %s) を持つ重複接続が検出されたため、古い接続 (接続 ID = % PRINTF\_LLFMT d) を破棄しています )

Warning, deleting and recreating durable '%s' due to change in client attributes: %s%s%s ( 警告。次のクライアント属性が変更されたため、持続 '%s' を削除して、再作成しています : %s%s%s )

Unable to build monitor message. ( 監視メッセージを作成できません。 ) Error code: %d - %s ( エラーコード : %d - %s )

Multicast is enabled. ( マルチキャストが有効です。 )

%s: Multicast consumers require NO\_ACKNOWLEDGE sessions: %s (%s : マルチキャストコンシューマが NO\_ACKNOWLEDGE セッションを要求しています : %s )

%s: Multicast consumers require non-transacted sessions: %s (%s : マルチキャストコンシューマが非トランザクションセッションを要求しています : %s )

%s: Multicast consumer status: %s (consid=% PRINTF\_LLFMT d) (%s : マルチキャストコンシューマのステータス : %s (接続 ID = % PRINTF\_LLFMT d) )

%s: Multicast consumer status: %s (consid=% PRINTF\_LLFMT d channel='%s') (%s : マルチキャストコンシューマのステータス : %s (コンシューマ ID = % PRINTF\_LLFMT d チャンネル = '%s') )

[%s]: tx=% PRINTF\_LLFMT d bytes, rtx=% PRINTF\_LLFMT d bytes, buf=% PRINTF\_LLFMT d bytes ([%s] : tx=% PRINTF\_LLFMT d バイト、rtx=% PRINTF\_LLFMT d バイト、buf=% PRINTF\_LLFMT d バイト )

[%s@%s %s]: rcv=% PRINTF\_LLFMT d bytes, lost=% PRINTF\_LLFMT d, naks=% PRINTF\_LLFMT d, failed=% PRINTF\_LLFMT d ([%s@%s %s] : rcv=% PRINTF\_LLFMT d バイト、lost=% PRINTF\_LLFMT d、naks=% PRINTF\_LLFMT d、failed=% PRINTF\_LLFMT d)

%s (%s)

%s: JMX stats for store '%s' updated: %s (%s : ストア '%s' の JMX ステートが更新されました : %s)

Route configuration: Adding topic '%s' for server %s ( ルート設定 : サーバー '%s' のトピック %s を追加しています )

Route configuration: Sending %d topics to server %s at %s - %s ( ルート設定 : %d 個のトピックをサーバー %s に %s - %s で送信しています )

Route configuration: Sending topics to server %s at %s - %s ( ルート設定 : トピックをサーバー %s に %s - %s で送信しています )

Route configuration: Sending queue '%s' to server %s at %s - %s ( ルート設定 : キュー '%s' をサーバー %s に %s - %s で送信しています )

Route configuration: Sending queues to server %s at %s - %s ( ルート設定 : キューをサーバー %s に %s - %s で送信しています )

Route configuration: Processing topics from server %s at %s. ( ルート設定 : サーバー %s からのトピックを %s で処理しています。 )

Route configuration: Processing queue '%s' from server %s at %s. ( ルート設定 : サーバー %s からのキュー '%s' を %s で処理しています。 )

Route '%s' acknowledgment timer destroyed. ( ルート '%s' 確認応答タイマーが破棄されました。 )

Route '%s' acknowledgment timer unknown event type. ( ルート '%s' 確認応答タイマーが不明なイベントタイプです。 )

Route '%s' acknowledgment timer send failed: %s ( ルート '%s' 確認応答タイマーの送信が失敗しました : %s )

Route '%s' acknowledgment timer connection % PRINTF\_LLFMT d not found. ( ルート '%s' 確認応答タイマー接続 % PRINTF\_LLFMT d が見つかりません。 )

Route configuration: Processing queues from server %s at %s. ( ルート設定 : サーバー %s からのキューを %s で処理しています。 )

Discarded incoming client message exceeding size limit: connID=% PRINTF\_LLFMT d, %s='%s', size=%d. ( 破棄された着信クライアントメッセージがサイズ制限を超えています : 接続 ID = % PRINTF\_LLFMT d、%s='%s'、サイズ = %d。 )

%s%s selector exceeded selector\_logical\_operator\_limit of %d. (%s%s セレクタが %d の selector\_logical\_operator\_limit を超えました。 )

Configuration update failure: %s ( 設定の更新が失敗しました : %s )

Rolling back to configuration on disk. ( ディスク上の設定にロールバックしています。 )

Rollback failed: %s ( ロールバックが失敗しました : %s )

Rollback succeeded. ( ロールバックが成功しました。 )

%s property [%s]. (%s プロパティ [%s]。 )

%s configuration item: (%s 設定項目 : )

%s status: %s (%s ステータス : %s )

Illegal property get in \_emsdConfigurationObjectProperty\_GetStringValue. ( 不正なプロパティが \_emsdConfigurationObjectProperty\_GetStringValue にあります。 )

Attempting to generate a search key for an unsearchable object (%s). ( 検索不可能なオブジェクトの検索キーを生成しようとしています (%s) )

More than one result found searching route selectors route=%s topic=%s. ( ルートセレクトタの検索中に、複数の結果が検出されました (ルート = %s トピック = %s) )

More than one result found searching alias %s=%s jndiName=%s. ( 別名 %s=%s jndi 名 = %s の検索中に、複数の結果が検出されました )

Fault tolerant configurations should have a primary listens marked as ft\_active - FT will be disabled. ( フォールトトレランス設定では、プライマリリスンを ft\_active としてマークする必要があります - FT は無効になります )

Fault tolerant configurations should have a secondary listens marked as ft\_active - FT will be disabled. ( フォールトトレランス設定では、セカンダリリスンを ft\_active としてマークする必要があります - FT は無効になります )

Configured as fault tolerant secondary. ( フォールトトレランスのセカンダリとして設定しました )

Configured as fault tolerant primary. ( フォールトトレランスのプライマリとして設定しました )

Unable to find group %s. ( グループ %s を検出できません )

Detected Mixed mode configuration: Ignoring property %s file=%s, line=%d ( 検出された混合モード設定 : プロパティ %s を無視します。ファイル = %s、行 = %d )

Missing field %s from server object %s near line %d. ( フィールド %s がサーバーオブジェクト %s から %d 行の近辺で欠落しています )

Invalid field detected in server object (%s), field (%s), at line %d. ( 無効なフィールドがサーバーオブジェクト (%s)、フィールド (%s)、行 %d で検出されました )

%s will not complete until the server is restarted. ( サーバーが再起動されるまで、%s は完了しません )

Detected unsupported password hash for user %q%s¥q. The user's password may need to be reset. ( ユーザー %q%s¥q にサポートされていないパスワードハッシュが検出されました。ユーザーのパスワードをリセットする必要がある可能性があります )

Ignoring request to remove an administrative user. ( 管理者ユーザーを削除する要求を無視します )

Ignoring request to remove the administrative group. ( 管理者グループを削除する要求を無視します )

%s will not occur until fault tolerant failover or restart. ( フォールトトレランスフェイルオーバーまたは再起動が実行されるまで、%s は発生しません )

Ignoring unknown property %s. ( 不明なプロパティ %s を無視します )



Configuration error: file=%s, line=%d: ignoring rvcmlistener ( 設定エラー : ファイル = %s、行 = %d : rvcmlistener を無視します )

Illegal configuration namespace set, %s=%s Expecting %s. ( 不正な設定名前空間のセット、%s = %s は %s である必要があります。 )

Failed to write file %s: (%s) ( ファイル %s に書き込めませんでした : (%s) )

Error loading configuration: %s ( 設定の読み込み中にエラーが発生しました : %s )

Applying configuration changes. ( 設定の変更を適用しています。 )

Configuration update status: %s ( 設定の更新ステータス : %s )

Rollback unable to load file %s. ( ロールバックによりファイル %s を読み込めません。 )

Initialized queue browser on '%s'%s%s%s ( キューブラウザを '%s'%s%s%s で初期化しました。 )

Creating store '%s' file '%s' ... ( ストア '%s' ファイル '%s' を作成しています ... )

Converting %s format from %s to %s (%s 形式を %s から %s に変換しています )

First scan on '%s' is finished ('%s' の最初のスキャンが完了しました )

Destination cursor id % PRINTF\_LL\_FMT d %s slot %d ( デスティネーションカーソル ID % PRINTF\_LL\_FMT d %s スロット %d )

Client browsing mstore-based queue %s needs to be upgraded for optimum performance. ( 最適なパフォーマンスを得るために、mstore ベースのキュー %s を参照しているクライアントをアップグレードする必要があります。 )

Warning: file=%s, line=%d: multicast\_udp\_encapsulation is no longer a supported parameter. ( 警告 : ファイル = %s、行 = %d : multicast\_udp\_encapsulation はサポートされているパラメータではありません。 )

Network IO thread: %d bound to Processor id: %d ( ネットワーク IO スレッド %d はプロセッサ ID %d にバインドされました )

Storage thread for store '%s' bound to Processor id: %d ( ストレージ '%s' のストレージスレッドはプロセッサ ID にバインドされました : %d )

Disconnecting connection connID=% PRINTF\_LL\_FMT d, because %s (connID=% PRINTF\_LL\_FMT d) has reached the limit of non-acknowledged messages (%s (接続 ID = % PRINTF\_LL\_FMT d) が非確認応答メッセージの制限に到達したため、接続 (接続 ID = % PRINTF\_LL\_FMT d) を切断しています )

カテゴリ 基本的な初期化の失敗

説明 tibemsd を起動できませんでした。



**解決方法** 設定または起動パラメータを修正し、再起動します。

- エラー** Unable to add admin user into admin group: error=(%d) %s (admin ユーザーを admin グループに追加できません: エラー = (%d) %s)
- Fault tolerant activation has to be greater than 2x heartbeat (フォールトトレランスのアクティブ化には、ハートビートの 2 倍よりも大きい値を指定する必要があります)
- Server heartbeat client should be non-zero and no more than a third of the client timeout server connection (サーバーのハートビートクライアントはゼロ以外で、クライアントタイムアウトサーバー接続の 3 分の 1 より多い必要があります)
- Server heartbeat server should be non-zero and no more than a third of the server timeout server connection (サーバーのハートビートサーバーはゼロ以外で、サーバータイムアウトサーバー接続の 3 分の 1 以下である必要があります)
- Client heartbeat server should be non-zero and no more than a third of the server timeout client connection (クライアントのハートビートサーバーはゼロ以外で、サーバータイムアウトクライアント接続の 3 分の 1 以下である必要があります)
- Fault Tolerant configuration error, can't create loop. (フォールトトレランス設定のエラーです。ループを作成できません。)
- Fault tolerant connection failed, fault tolerant mode not supported on '%s'. (フォールトトレランス接続に失敗しました。'%s' ではフォールトトレランスモードはサポートされていません。)
- Fault tolerant heartbeat has to be greater than 0 (フォールトトレランスハートビートには、0 より大きい値を指定する必要があります)
- Initialization failed due to errors in configuration. (設定にエラーがあるため、初期化に失敗しました。)
- Initialization failed due to errors in SSL. (SSL にエラーがあるため、初期化に失敗しました。)
- Initialization failed due to errors with transports. (トランスポートにエラーがあるため、初期化に失敗しました。)
- Initialization failed. (初期化に失敗しました。) Exiting. (終了します。)
- Initialization has failed. (初期化に失敗しました。) Exiting. (終了します。)
- Initialization of thread pool failed (%s). (スレッドプールの初期化に失敗しました (%s)。) Exiting. (終了します。)
- Startup aborted. (起動を中断しました。)
- Server failed to read configuration. (サーバーが設定の読み取りに失敗しました。)
- Initialization failed: storage for '%s' not found. (初期化に失敗しました: '%s' のストレージが見つかりません。)

Failure initializing storage thread: %s. ( ストレージスレッドの初期化に失敗しました : %s。 )

Initialization failed due to errors with multicast. ( マルチキャストにエラーがあるため、初期化に失敗しました。 )

Configuration error: dbstore\_driver\_name for store [%s] cannot be empty ( 設定エラー : ストア [%s] の dbstore\_driver\_name を空にすることはできません )

Configuration error: dbstore\_driver\_url for store [%s] cannot be empty ( 設定エラー : ストア [%s] の dbstore\_driver\_url を空にすることはできません )

Configuration error: dbstore\_driver\_dialect for store [%s] cannot be empty ( 設定エラー : ストア [%s] の dbstore\_driver\_dialect を空にすることはできません )

Configuration error: dbstore\_driver\_username for store [%s] must be specified ( 設定エラー : ストア [%s] の dbstore\_driver\_username を指定する必要があります )

Configuration error: dbstore\_driver\_password for store [%s] must be specified ( 設定エラー : ストア [%s] の dbstore\_driver\_password を指定する必要があります )

Error Loading JVM: %s (JVM の読み込み中にエラーが発生しました : %s)

Unknown Error Loading JVM (JVM の読み込み中に不明なエラーが発生しました )

Trying JVM location: %s (JVM を検出しています : %s)

Error Loading JVM: %s (JVM の読み込み中にエラーが発生しました : %s)

Unknown Error Loading JVM (JVM の読み込み中に不明なエラーが発生しました )

\$sys.meta store's type must be 'file' or 'dbstore'. (\$sys.meta ストアのタイプは 'file' または 'dbstore' である必要があります。 )

Configuration error: file=%s, line=%d: The parameter '%s' is not supported on this platform ( 設定エラー : ファイル = %s、行 = %d : パラメータ '%s' はこのプラットフォームでサポートされていません )

Configuration error: file=%s, line=%d: The processor ID '%d' is greater than '%d', which is the maximum processor ID for this machine ( 設定エラー : ファイル = %s、行 = %d : プロセッサ ID '%d' が、このマシンの最大プロセッサ ID である '%d' より大きくなっています )

Unable to bind network IO thread: %d to Processor Id: %d. Exiting! ( ネットワーク IO スレッド %d をプロセッサ ID %d にバインドできません。終了します。 )

Unable to bind storage thread for store '%s' to Processor Id: %d. Exiting! ( ストア '%s' のストレージスレッドをプロセッサ ID %d にバインドできません。終了します。 )

カテゴリ	以前に障害が発生していたため、またはフォールトトレランス スイッチオーバー後であるため、コミットに失敗した
説明	クライアントアプリケーションのトランザクションのコミットに失敗したことを示す警告メッセージです。このトランザクションを処理したときに初期段階でエラーがあったか、またはフォールトトレランスフェイルオーバーの前にプライマリサーバー上でトランザクションが開始されたことが原因です。
解決方法	クライアントアプリケーションで、トランザクションを再試行します。
エラー	Commit failed due to prior failure or after fault-tolerant switch. ( 以前に障害が発生していたため、またはフォールトトレランス スイッチオーバー後であるため、コミットに失敗しました。 )
<hr/>	
カテゴリ	圧縮の失敗
説明	ストアファイルの圧縮に失敗しました。
解決方法	多くの場合、このエラーの原因はメモリ不足です。tibemspd をシャットダウンし、メモリ不足時の対処方法を確認してください。
エラー	<p>Compaction failed on file '%s': %d (%s). ( ファイル '%s' の圧縮に失敗しました : %d (%s)。 ) Please shutdown and restart tibemspd. (tibemspd をシャットダウンしてから再起動してください。 )</p> <p>Compaction failed on file '%s': %d (%s). ( ファイル '%s' の圧縮に失敗しました : %d (%s)。 )</p> <p>Initialization of file_destination_defrag feature failed for queue '%s' (store '%s') due to an out of memory condition. ( メモリが不足しているため、キュー '%s' (ストア '%s') に対する file_destination_defrag 機能の初期化が失敗しました。 ) Feature is disabled. ( 機能が無効になっています。 )</p> <p>file_destination_defrag of queue '%s' (store '%s') failed: %d (%s). ( キュー '%s' (ストア '%s') の file_destination_defrag が失敗しました : %d (%s)。 )</p>
<hr/>	
カテゴリ	設定された持続が、保存されている持続と異なる
説明	持続設定ファイルで、特定の名前とクライアント ID を持つ持続を指定しましたが、その持続の属性が、meta.db ファイル内で見つかった、同じ名前の持続の属性と異なります。
解決方法	meta.db ファイルに定義されている持続と一致するように持続設定ファイルを修正するか、または定義されている持続を管理者権限で削除し、再定義します。

エラー	Configured durable '%s' differs from durable in storage, storage version used. ( 設定された持続 '%s' は、ストレージ内の持続と異なります。ストレージ内の持続が使用されます。 )
カテゴリ	グローバルルーティング先トピックの作成に失敗しました。動的トピックの作成は許可されていません。
説明	サーバーが、設定で許可されているトピックと一致しない通知対象を、別のサーバーから受信しました。
解決方法	これは、トレースに ROUTE_DEBUG が含まれている場合にのみ出力されます。サーバーのトピック定義が指定どおりである場合、このステートメントを無視するか、または ROUTE_DEBUG トレース指定を削除してエラーが出力されないようにします。
エラー	Create of global routed topic failed: not allowed to create dynamic topic [%s]. ( グローバルルーティング先トピックの作成に失敗しました。動的トピックの作成は許可されていません [%s]。 )
カテゴリ	ルーティング先キューの作成の失敗：動的キューの作成は許可されない
説明	このデーモンへのルートを持つ tibemsd にはグローバルに設定されたキューがありますが、このデーモンがそのキューの動的な作成を許可しないことを示す警告です。
解決方法	指定されたキューにこのデーモンからアクセスできるようにするには、指定されたキューまたはそのキューを含むパターンをこのデーモンに追加します。または、この警告を無視しても構いません。
エラー	Create of routed queue failed: not allowed to create dynamic queue [%s]. ( ルーティング先キューの作成に失敗しました：動的キューの作成は許可されていません [%s]。 )
カテゴリ	データベースレコードの破損
説明	tibemsd ストアファイルの読み取り中にエラーが発生しました。
解決方法	エラーおよびエラーが発生した状況の詳細を TIBCO サポートに送信します。
エラー	Server failed to recover state. ( サーバーが状態の回復に失敗しました。 )

カテゴリ	重複メッセージの検出
説明	tibemspd が受信したメッセージのメッセージ ID が別のメッセージのメッセージ ID と一致するときに生成される警告です。
解決方法	この警告は、メッセージ ID の追跡が有効になっている場合にのみ表示されます。
エラー	Detected duplicate %s message, messageID='%s' ( 重複するメッセージ %s が検出されました。メッセージ ID = '%s' )
<hr/>	
カテゴリ	動的モジュールの読み込みエラー
説明	共有ライブラリモジュールの読み込みまたは使用中にエラーが発生しました。
解決方法	モジュールの読み込みは、モジュールパスに共有ライブラリが存在することによって影響を受けます。+load トレースフラグを使用して、サーバーがモジュールを読み込む方法についての詳細情報を取得します。詳細は、「EMS サーバーの起動」のセクションを参照してください。
エラー	<p>Problem loading %s: %s (%s の読み込み中に問題が発生しました : %s)</p> <p>Unknown problem loading %s. (%s の読み込み中に不明な問題が発生しました。)</p> <p>Loaded %s (%s を読み込みました )</p> <p>Problem binding %s: %s (%s のバインド中に問題が発生しました : %s)</p> <p>Unknown problem binding %s. (%s のバインド中に不明な問題が発生しました。)</p> <p>Unable to locate %s (%s を見つけることができません )</p> <p>Fatal error: Returned from exec(), errno = %d ( 致命的なエラー : exec() から返されました。エラー番号 = %d )</p> <p>OpenSSL library version mismatch (OpenSSL ライブラリのバージョンが一致しません )</p>
<hr/>	
カテゴリ	設定ファイルのエラー
説明	サーバーは、指定された設定ファイルの指定された行で、無効な設定ステートメントを検出しました。
解決方法	該当する設定ファイルを調査して、構文エラーを修正します。

- エラー Configuration warning: file=%s, line=%d: route '%s' does not have a user configured for authorization. ( 設定に関する警告 : ファイル = %s、行 = %d : ルート '%s' に、認可が設定されているユーザーが存在しません。 )
- SSL Configuration error: file=%s, line=%d: invalid certificate file name, unknown extension or invalid encoding specification (SSL 設定エラー : ファイル = %s、行 = %d : 無効な証明書ファイル名、不明な拡張子、または無効なエンコーディング指定があります )
- Configuration error: file=%s, line=%d: illegal to specify %s for routed queue ( 設定エラー : ファイル = %s、行 = %d : ルーティング先キューに %s を指定することはできません )
- Configuration error: file=%s, line=%d: bad destination specification: %s ( 設定エラー : ファイル = %s、行 = %d : 間違ったデスティネーション指定 : %s )
- Configuration warning: file=%s, line=%d: illegal to specify prefetch=none for global or routed queue. ( 設定に関する警告 : ファイル = %s、行 = %d : グローバルキューまたはルーティング先キューに、"prefetch=none" を指定することはできません。 ) Prefetch reset to default. ( プリフェッチはデフォルトにリセットされます。 )
- Configuration warning: file=%s, line=%d: illegal to specify prefetch=none for topic. ( 設定に関する警告 : ファイル = %s、行 = %d : トピックに prefetch=none を指定することはできません。 ) Prefetch reset to default. ( プリフェッチはデフォルトにリセットされます。 )
- Configuration error: file=%s, line=%d: ignored alias '%s' for %s '%s' because such alias already exists ( 設定エラー : ファイル = %s、行 = %d : 同じエイリアスがすでに存在するため、%s '%s' のエイリアス '%s' は無視されました )
- Configuration error: The specified file '%s' is empty or does not exist ( 設定エラー : 指定されたファイル '%s' は空であるか、存在しません )
- Configuration error: file=%s, line=%d: both tibrv\_export and tibrvcm\_export are specified, ignoring tibrv\_export ( 設定エラー : ファイル = %s、行 = %d : tibrv\_export と tibrvcm\_export が両方指定されています。 tibrv\_export を無視します。 )
- Configuration error: file=%s, line=%d: ignoring transport '%s' in %s list, transport not found ( 設定エラー : ファイル = %s、行 = %d : リスト %s のトランスポート '%s' を無視します。トランスポートが見つかりません。 )
- Configuration error: file=%s, line=%d: multiple bridge entries for the same destination '%s' are not allowed. ( 設定エラー : ファイル = %s、行 = %d : 同じデスティネーション '%s' への複数のブリッジエントリは、許可されていません。 )

Configuration error: file=%s, line=%d: Ignoring durable, name cannot start with \$sys.route, use route property instead. ( 設定エラー : ファイル = %s、行 = %d : 持続を無視します。\$sys.route で始まる名前は許可されていません。代わりにルートプロパティを使用します。)

Configuration error: file=%s, line=%d: Rendezvous transport not specified for Rendezvous CM transport '%s' ( 設定エラー : ファイル = %s、行 = %d : Rendezvous CM トランスポート '%s' に Rendezvous トランスポートが指定されていません)

Configuration error: file=%s, line=%d: ignoring invalid max connections in the line, reset to unlimited ( 設定エラー : ファイル = %s、行 = %d : この行に指定されている無効な接続の最大数を無視します。制限なしにリセットします。)

Configuration error: file=%s, line=%d: ignoring invalid max\_client\_msg\_size in the line, reset to unlimited ( 設定エラー : ファイル = %s、行 = %d : この行に指定されている無効な max\_client\_msg\_size を無視します。制限なしにリセットします)

Configuration error: file=%s, line=%d: value of %s out of range, reset to default ( 設定エラー : ファイル = %s、行 = %d : %s の値が範囲外です。デフォルトにリセットします。)

Configuration error: max\_msg\_field\_print\_size >= max\_msg\_print\_size, resetting both to default ( 設定エラー : max\_msg\_field\_print\_size >= max\_msg\_print\_size、両方をデフォルトにリセットしています)

Configuration error: file=%s, line=%d: unable to create %s '%s': invalid destination name, invalid parameters or out of memory ( 設定エラー : ファイル = %s、行 = %d : %s '%s' を作成できません : デスティネーション名またはパラメータが無効であるか、メモリ不足です。)

Configuration error: file=%s, line=%d: value of db\_pool\_size too big or less than allowed minimum, reset to default value of %d bytes ( 設定エラー : ファイル = %s、行 = %d : db\_pool\_size の値が大き過ぎるか、最小許容値未満です。デフォルト値の %d バイトにリセットします。)

Configuration error: file=%s, line=%d: Ignoring durable, route does not allow clientid, selector or nlocal. ( 設定エラー : ファイル = %s、行 = %d : 持続を無視します。ルートには clientid、セレクトタ、または nlocal を指定できません。)

Configuration error: file=%s, line=%d: Route '%s' does not exist for configured durable. ( 設定エラー : ファイル = %s、行 = %d : 設定された持続にルート '%s' が存在しません。)

Configuration error: file=%s, line=%d: unable to process selector in route parameters, error=%s ( 設定エラー : ファイル = %s、行 = %d : ルートパラメータ内のセレクトタを処理できません。エラー = %s)



Configuration error: file=%s, line=%d: both tibrv\_import and tibrvcn\_import are specified, ignoring tibrv\_import ( 設定エラー : ファイル = %s、行 = %d : tibrv\_import と tibrvcn\_import が両方指定されています。tibrv\_import を無視します。)

Configuration error: file=%s, line=%d: ignored route '%s' because route represents route to this server. ( 設定エラー : ファイル = %s、行 = %d : ルートがこのサーバーへのルートを表しているため、ルート '%s' は無視されました。)

Configuration error: file=%s, line=%d: ignoring invalid topic selector specifications in route parameters ( 設定エラー : ファイル = %s、行 = %d : ルートパラメータ内の無効なトピックセレクタ指定を無視します)

Configuration error: file=%s, line=%d: value of max\_msg\_memory less than allowed, reset to %dMB ( 設定エラー : ファイル = %s、行 = %d : max\_msg\_memory の値が許容値未満です。%dMB にリセットします。)

Configuration error: file=%s, line=%d: ignored alias '%s' for factory because such alias already exists ( 設定エラー : ファイル = %s、行 = %d : 同じエイリアスがすでに存在するため、ファクトリのエイリアス '%s' は無視されました)

Configuration error: file=%s, line=%d: invalid certificate file name, unknown extension or invalid encoding specification ( 設定エラー : ファイル = %s、行 = %d : 無効な証明書ファイル名、不明な拡張子、または無効なエンコーディング仕様があります。)

Configuration error: file=%s, line=%d: ignored route '%s' because route has invalid zone information. ( 設定エラー : ファイル = %s、行 = %d : ルート '%s' に無効なゾーン情報があるため、このルートは無視されました。)

Configuration error: file=%s, line=%d: ignored route '%s' because route with such name or URL already exists. ( 設定エラー : ファイル = %s、行 = %d ; 同じ名前のルートまたは URL がすでに存在するため、ルート '%s' は無視されました。)

Configuration error: file=%s, line=%d: value of msg\_pool\_size invalid or too big or less than allowed minimum of %d, reset to default value of %d ( 設定エラー : ファイル = %s、行 = %d : msg\_pool\_size の値が無効であるか、大き過ぎるか、または最小許容値未満です。デフォルト値 %d にリセットします。)

SSL Configuration error: file=%s, line=%d: invalid private key file name, unknown extension or invalid encoding specification (SSL 設定エラー : ファイル = %s、行 = %d : 無効な秘密鍵ファイル名、不明な拡張子、または無効なエンコーディング指定があります。)

Configuration conflict: file=%s, line=%d: value of msg\_pool\_block\_size already set at line=%d. Ignoring msg\_pool\_size. ( 設定の競合 : ファイル = %s、行 = %d : msg\_pool\_block\_size の値は、行 = %d にすでに設定されています。msg\_pool\_size を無視します。)



Configuration error: file=%s, line=%d: bridge has no targets, unable to process ( 設定エラー : ファイル = %s、行 = %d : ブリッジにターゲットがないため、処理できません )

Configuration error: file=%s, line=%d: Illegal to specify routed queue as a bridge source ( 設定エラー : ファイル = %s、行 = %d : ルーティング先キューをブリッジソースとして指定することはできません )

Configuration error: file=%s, line=%d: client\_trace error: %s ( 設定エラー : ファイル = %s、行 = %d : client\_trace エラー : %s )

Configuration error: file=%s, line=%d: %s ( 設定エラー : ファイル = %s、行 = %d : %s )

Configuration error: file=%s, line=%d: duplicate specification of transport type ( 設定エラー : ファイル = %s、行 = %d : トランスポートタイプの指定が重複しています )

Configuration error: file=%s, line=%d: duplicate value ( 設定エラー : ファイル = %s、行 = %d : 値が重複しています )

Configuration error: file=%s, line=%d: Ignoring durable, duplicate of earlier entry. ( 設定エラー : ファイル = %s、行 = %d : 持続を無視します。以前のエン트리と重複しています。 )

Configuration error: file=%s, line=%d: Ignoring durable, name is invalid. ( 設定エラー : ファイル = %s、行 = %d : 持続を無視します。名前が無効です。 )

Configuration error: file=%s, line=%d: Ignoring durable, name is missing or invalid. ( 設定エラー : ファイル = %s、行 = %d : 持続を無視します。名前が指定されていないか、または無効です。 )

Configuration error: file=%s, line=%d: Ignoring durable, topic is invalid. ( 設定エラー : ファイル = %s、行 = %d : 持続を無視します。トピックが無効です。 )

Configuration error: file=%s, line=%d: Ignoring durable, topic is missing or invalid. ( 設定エラー : ファイル = %s、行 = %d : 持続を無視します。トピックが指定されていないか、または無効です。 )

Configuration error: file=%s, line=%d: error in the bridge description, unable to proceed. ( 設定エラー : ファイル = %s、行 = %d : ブリッジの説明にエラーがあります。続行できません。 )

Configuration error: file=%s, line=%d: error in permissions ( 設定エラー : ファイル = %s、行 = %d : アクセス権限にエラーがあります )

Configuration error: file=%s, line=%d: error in the transport description, unable to proceed. ( 設定エラー : ファイル = %s、行 = %d : トランスポートの説明にエラーがあります。続行できません。 )

Configuration error: file=%s, line=%d: errors in line, some options may have been ignored ( 設定エラー : ファイル = %s、行 = %d : 行にエラーがあります。一部のオプションが無視された可能性があります。)

Error: unable to add bridge specified in file=%s, line=%d. Error=%s ( エラー : ファイル = %s、行 = %d に指定されているブリッジを、追加することができません。エラー = %s)

Configuration error: file=%s, line=%d: Unable to create destination defined by the bridge source ( 設定エラー : ファイル = %s、行 = %d : ブリッジソースによって定義されたデスティネーションを、作成できません)

Unable to create Rendezvous Certified transport '%s' because it references undefined Rendezvous transport '%s' (Rendezvous 保証トランスポート '%s' を作成できません。この保証トランスポートは未定義の Rendezvous トランスポート '%s' を参照しています。)

Configuration error: file=%s, line=%d: failed to create ACL entry, reason=%s ( 設定エラー : ファイル = %s、行 = %d : ACL エントリの作成に失敗しました。原因 = %s)

Unable to export message to SmartSockets. error=%s. (SmartSockets にメッセージをエクスポートできません。エラー = %s。)

Use fsync error: file=%s, line=%d: invalid property value (fsync の使用エラー : ファイル = %s、行 = %d : プロパティ値が無効です)

Use fsync (min disk) error: file=%s, line=%d: invalid property value (fsync (最小ディスク) の使用エラー : ファイル = %s、行 = %d : プロパティ値が無効です)

exit\_on\_nonretryable\_disk\_error: file=%s, line=%d: invalid boolean property value (exit\_on\_nonretryable\_disk\_error : ファイル = %s、行 = %d : ブール値プロパティの値が無効です)

consumed\_msg\_hold\_time: file=%s, line=%d: invalid property value (consumed\_msg\_hold\_time : ファイル = %s、行 = %d : プロパティ値が無効です)

active\_route\_connect\_time: file=%s, line=%d: invalid property value (active\_route\_connect\_time : ファイル = %s、行 = %d : プロパティ値が無効です)

Fault tolerant reread error: file=%s, line=%d: invalid property value ( フォールトレランス再読み込みエラー : ファイル = %s、行 = %d : プロパティ値が無効です)

Fault standby lock check error: file=%s, line=%d: invalid property value ( フォールトスタンバイロックチェックエラー : ファイル = %s、行 = %d : プロパティ値が無効です)

Configuration error: file=%s, line=%d: ignored unknown permission '%s' ( 設定エラー : ファイル = %s、行 = %d : 不明なアクセス権限 '%s' は無視されました)

Configuration error: file=%s, line=%d: ignoring duplicate %s '%s' specified earlier ( 設定エラー : ファイル = %s、行 = %d : 以前に指定された、重複する %s '%s' を無視します )

Configuration error: file=%s, line=%d: ignoring duplicate transport name '%s' in %s list ( 設定エラー : ファイル = %s、行 = %d : %s リスト内の重複するトランスポート名 '%s' を無視します )

Configuration error: file=%s, line=%d: ignoring duplicate user ( 設定エラー : ファイル = %s、行 = %d : 重複するユーザーを無視します )

Configuration error: file=%s, line=%d: ignoring errors in permission line ( 設定エラー : ファイル = %s、行 = %d : アクセス権限行のエラーを無視します )

Configuration error: file=%s, line=%d: ignoring invalid connect attempt count ( 設定エラー : ファイル = %s、行 = %d : 無効な接続試行回数を無視します )

Configuration error: file=%s, line=%d: ignoring invalid connect attempt delay ( 設定エラー : ファイル = %s、行 = %d : 無効な接続試行遅延を無視します )

Configuration error: file=%s, line=%d: ignoring invalid connect attempt timeout ( 設定エラー : ファイル = %s、行 = %d : 無効な接続試行タイムアウトを無視します )

Configuration error: file=%s, line=%d: ignoring invalid disk statistic period ( 設定エラー : ファイル = %s、行 = %d : 無効なディスク統計情報期間を無視します )

Configuration error: file=%s, line=%d: ignoring invalid entry syntax ( 設定エラー : ファイル = %s、行 = %d : 無効なエントリ構文を無視します )

Configuration error: file=%s, line=%d: ignoring invalid factory load balancing metric ( 設定エラー : ファイル = %s、行 = %d : 無効なファクトリ負荷分散の測定基準を無視します )

Configuration error: file=%s, line=%d: ignoring invalid ft activation in the line ( 設定エラー : ファイル = %s、行 = %d : この行の無効な ft アクティブ化を無視します )

Configuration error: file=%s, line=%d: ignoring invalid ft heartbeat in the line ( 設定エラー : ファイル = %s、行 = %d : この行の無効な ft ハートビートを無視します )

Configuration error: file=%s, line=%d: ignoring invalid ft reconnect timeout in the line ( 設定エラー : ファイル = %s、行 = %d : この行の無効な ft 再接続タイムアウトを無視します )

Configuration error: file=%s, line=%d: ignoring invalid line ( 設定エラー : ファイル = %s、行 = %d : 無効な行を無視します )

Configuration error: file=%s, line=%d: ignoring invalid line in factory parameters ( 設定エラー : ファイル = %s、行 = %d : ファクトリパラメータ内の無効な行を無視します )

Configuration error: file=%s, line=%d: ignoring invalid line in route parameters (設定エラー: ファイル = %s、行 = %d: ルートパラメータ内の無効な行を無視します)

Configuration error: file=%s, line=%d: ignoring invalid line: invalid syntax in the line (設定エラー: ファイル = %s、行 = %d: 無効な行を無視します: この行に無効な構文があります)

Configuration error: file=%s, line=%d: ignoring invalid reconnect attempt count (設定エラー: ファイル = %s、行 = %d: 無効な再接続試行回数を無視します)

Configuration error: file=%s, line=%d: ignoring invalid reconnect attempt delay (設定エラー: ファイル = %s、行 = %d: 無効な再接続試行遅延を無視します)

Configuration error: file=%s, line=%d: ignoring invalid reconnect attempt timeout (設定エラー: ファイル = %s、行 = %d: 無効な再接続試行タイムアウトを無視します)

Configuration error: file=%s, line=%d: ignoring invalid value of %s (設定エラー: ファイル = %s、行 = %d: 無効な値 %s を無視します)

Configuration error: file=%s, line=%d: ignoring invalid value '%s' for property '%s' (設定エラー: ファイル = %s、行 = %d: プロパティ '%s' の無効な値 '%s' を無視します。)

Configuration error: file=%s, line=%d: ignoring unknown property '%s' (設定エラー: ファイル = %s、行 = %d: 不明なプロパティ '%s' を無視します)

Configuration error: file=%s, line=%d: ignoring unrecognized property '%s' (設定エラー: ファイル = %s、行 = %d: 認識できないプロパティ '%s' を無視します)

Configuration error: file=%s, line=%d: ignoring user out of group context (設定エラー: ファイル = %s、行 = %d: グループコンテキスト外のユーザーを無視します)

Configuration error: file=%s, line=%d: illegal to use predefined name %s (設定エラー: ファイル = %s、行 = %d: 定義済みの名前 '%s' を使用することはできません)

Configuration error: file=%s, line=%d: Invalid clientid value (設定エラー: ファイル = %s、行 = %d: clientid の値が無効です)

Configuration error: file=%s, line=%d: invalid value of db\_pool\_size, reset to default of %d bytes (設定エラー: ファイル = %s、行 = %d: db\_pool\_size の値が無効です。デフォルトの %d バイトにリセットします)

Configuration error: file=%s, line=%d: invalid line syntax or line out of order (設定エラー: ファイル = %s、行 = %d: 行の構文が無効であるか、または行の順序が違います)

Configuration error: file=%s, line=%d: invalid value of max memory, reset to unlimited ( 設定エラー : ファイル = %s、行 = %d : メモリ最大値の値が無効です。制限なしにリセットします )

Configuration error: file=%s, line=%d: invalid value of max\_msg\_memory, reset to unlimited ( 設定エラー : ファイル = %s、行 = %d : max\_msg\_memory の値が無効です。制限なしにリセットします )

Configuration error: file=%s, line=%d: invalid property value ( 設定エラー : ファイル = %s、行 = %d : プロパティ値が無効です )

Configuration error: file=%s, line=%d: invalid property value, reset to default. ( 設定エラー : ファイル = %s、行 = %d : プロパティ値が無効です。デフォルトにリセットします。 )

Configuration error: file=%s, line=%d: invalid password ( 設定エラー : ファイル = %s、行 = %d : パスワードが無効です )

Configuration error: file=%s, line=%d: invalid value of reserve\_memory, reset to zero ( 設定エラー : ファイル = %s、行 = %d : reserve\_memory の値が無効です。ゼロにリセットします )

Configuration error: file=%s, line=%d: invalid value of route\_recover\_interval, reset to default %d ( 設定エラー : ファイル = %s、行 = %d : route\_recover\_interval の値が無効です。デフォルトの %d にリセットします )

Configuration error: file=%s, line=%d: invalid value of route\_recover\_count, line ignored ( 設定エラー : ファイル = %s、行 = %d : route\_recover\_count の値が無効です。この行は無視されました )

Configuration error: file=%s, line=%d: Invalid selector value ( 設定エラー : ファイル = %s、行 = %d : セレクタ値が無効です )

Configuration error: file=%s, line=%d: invalid syntax of %s, unable to continue. ( 設定エラー : ファイル = %s、行 = %d : 構文 %s が無効です。続行できません。 )

Configuration error: file=%s, line=%d: invalid transport parameter '%s' ( 設定エラー : ファイル = %s、行 = %d : トランスポートパラメータ '%s' が無効です )

Configuration error: file=%s, line=%d: invalid transport type '%s' ( 設定エラー : ファイル = %s、行 = %d : トランスポートタイプ '%s' が無効です )

Configuration error: file=%s, line=%d: invalid trace\_client\_host value ( 設定エラー : ファイル = %s、行 = %d : trace\_client\_host の値が無効です )

Configuration error: file=%s, line=%d: invalid trace\_millisecond value ( 設定エラー : ファイル = %s、行 = %d : trace\_millisecond の値が無効です )

Configuration error: file=%s, line=%d: invalid value of %s, reset to unlimited ( 設定エラー : ファイル = %s、行 = %d : %s の値が無効です。制限なしにリセットします )

Configuration error: file=%s, line=%d: invalid value '%s' ( 設定エラー : ファイル = %s、行 = %d : 値 '%s' が無効です )

Configuration error: file=%s, line=%d: invalid value '%s' for parameter '%s' ( 設定エラー : ファイル = %s、行 = %d : パラメータ '%s' の値 '%s' が無効です )

Configuration error: file=%s, line=%d: invalid value of '%s' ( 設定エラー : ファイル = %s、行 = %d : '%s' の値が無効です )

Configuration error: file=%s, line=%d: invalid value of %s ( 設定エラー : ファイル = %s、行 = %d : %s の値が無効です )

Configuration error: file=%s, line=%d: invalid value of %s, reset to 256MB ( 設定エラー : ファイル = %s、行 = %d : %s の値が無効です。256MB にリセットします )

Configuration error: file=%s, line=%d: invalid value of %s, reset to default ( 設定エラー : ファイル = %s、行 = %d : %s の値が無効です。デフォルトにリセットします )

Configuration error: file=%s, line=%d: line too long, ignoring it ( 設定エラー : ファイル = %s、行 = %d : 行が長過ぎます。この行を無視します )

Configuration error: file=%s, line=%d: maximum number of listen interfaces reached. ( 設定エラー : ファイル = %s、行 = %d : リスニンタフェースの最大数に達しました。 )

Configuration error: file=%s, line=%d: multiple principals specified, line ignored ( 設定エラー : ファイル = %s、行 = %d : 複数のプリンシパルが指定されています。この行は無視されました )

Configuration error: file=%s, line=%d: multiple targets specified, line ignored ( 設定エラー : ファイル = %s、行 = %d : 複数のターゲットが指定されています。この行は無視されました。 )

Configuration error: file=%s, line=%d: out of memory, unable to create Rendezvous transport ( 設定エラー : ファイル = %s、行 = %d : メモリ不足です。Rendezvous トランスポートを作成できません )

Configuration error: file=%s, line=%d: no permissions found in acl entry ( 設定エラー : ファイル = %s、行 = %d : acl エントリにアクセス権限が見つかりません )

Configuration error: file=%s, line=%d: no target found in acl entry ( 設定エラー : ファイル = %s、行 = %d : acl エントリにターゲットが見つかりません )

Configuration error: file=%s, line=%d: %s '%s' not found ( 設定エラー : ファイル = %s、行 = %d : %s '%s' が見つかりません )

Configuration error: No topic exists for configured durable '%s%s%s'. ( 設定エラー : 設定された持続 '%s%s%s' のトピックは存在しません。 )

failed to create durable '%s', exception: %s. ( 持続 '%s' を作成できませんでした。例外 : %s。 )

Configuration error: file=%s, line=%d: no valid user or group found in acl entry ( 設定エラー : ファイル = %s、行 = %d : acl エントリに有効なユーザーまたはグループが見つかりません )

Configuration conflict: file=%s, line=%d: Overriding value of msg\_pool\_size already set at line=%d. ( 設定の競合 : ファイル = %s、行 = %d : 行 = %d に設定済みの msg\_pool\_size の値をオーバーライドしています。 )

Configuration warning: file=%s, line=%d: parameter '%s' is deprecated ( 設定に関する警告 : ファイル = %s、行 = %d : パラメータ '%s' は廃止されました )

Configuration error: file=%s, line=%d: value of reserve\_memory too small, reset to 16MB ( 設定エラー : ファイル = %s、行 = %d : reserve\_memory の値が小さ過ぎます。16MB にリセットします )

Configuration error: file=%s, line=%d: ignoring invalid line in route parameters: invalid zone type, too long ( 設定エラー : ファイル = %s、行 = %d : ルートパラメータ内の無効な行を無視します : 長過ぎる、無効なゾーンタイプがあります )

Configuration error: file=%s, line=%d: ignoring invalid line in route parameters: zone name exceeding %d bytes ( 設定エラー : ファイル = %s、行 = %d : ルートパラメータ内の無効な行を無視します : ゾーン名が %d バイトを超えています )

Routing Configuration error: file=%s, line=%d: invalid property value ( ルーティング設定エラー : ファイル = %s、行 = %d : プロパティ値が無効です )

Configuration warning: file=%s, line=%d: ignoring rvcmlistener, duplicate ( 設定に関する警告 : ファイル = %s、行 = %d : 重複する rvcmlistener を無視します )

Configuration error: file=%s, line=%d: ignoring rvcmlistener, first token is invalid ( 設定エラー : ファイル = %s、行 = %d : rvcmlistener を無視します。最初のトークンが無効です )

Configuration error: file=%s, line=%d: ignoring rvcmlistener, invalid destination ( 設定エラー : ファイル = %s、行 = %d : rvcmlistener を無視します。無効なデスティネーションです )

Configuration error: file=%s, line=%d: ignoring rvcmlistener, second token is invalid ( 設定エラー : ファイル = %s、行 = %d : rvcmlistener を無視します。2 番目のトークンが無効です )

Configuration error: file=%s, line=%d: ignoring rvcmlistener, third token is invalid ( 設定エラー : ファイル = %s、行 = %d : rvcmlistener を無視します。3 番目のトークンが無効です )

Configuration error: file=%s, line=%d: ignoring rvcmlistener, wildcards are not permitted ( 設定エラー : ファイル = %s、行 = %d : rvcmlistener を無視します。ワイルドカードは許可されていません )

SmartSockets configuration directory name is too long. error=%s. (SmartSockets 設定ディレクトリ名が長過ぎます。エラー = %s。 )



SmartSockets file '%s' not found. (SmartSockets ファイル '%s' が見つかりません。)

SSL Configuration error: file=%s, line=%d: duplicate value (SSL 設定エラー : ファイル = %s、行 = %d : 値が重複しています)

SSL Configuration error: file=%s, line=%d: invalid value of DH key size. (SSL 設定エラー : ファイル = %s、行 = %d : DH キーサイズの値が無効です。)

SSL Configuration error: file=%s, line=%d: invalid property value (SSL 設定エラー : ファイル = %s、行 = %d : プロパティ値が無効です)

Configuration error: file=%s, line=%d: syntax error in the line, ignoring (設定エラー : ファイル = %s、行 = %d : この行に構文エラーがあります。無視します)

Configuration error: file=%s, line=%d: syntax errors in line, line ignored (設定エラー : ファイル = %s、行 = %d : 行に構文エラーがあります。この行は無視されました)

Topic '%s' not valid in configured durable '%s'. (設定された持続 '%s' にあるトピック '%s' が無効です。)

Configuration error: file=%s, line=%d: Unrecognized attribute (設定エラー : ファイル = %s、行 = %d : 認識できない属性です)

Configuration error: file=%s, line=%d: user '%s' not found, ignoring (設定エラー : ファイル = %s、行 = %d : ユーザー '%s' が見つかりません。無視します)

Configuration error: file=%s, line=%d: value is invalid or less than minimum %d, reset to 0 (設定エラー : ファイル = %s、行 = %d : 値が無効であるか、最小値の %d 未満です。0 にリセットします)

Configuration error: file=%s, line=%d: value less than allowed minimum, reset to 0 (設定エラー : ファイル = %s、行 = %d : 値が最小許容値未満です。0 にリセットします)

Configuration error: file=%s, line=%d: value of %s less than allowed minimum of %dKB, reset to unlimited (設定エラー : ファイル = %s、行 = %d : %s の値が最小許容値の %dKB 未満です。制限なしにリセットします)

Configuration error: file=%s, line=%d: Invalid value or value does not fall between %d and %d (設定エラー : ファイル = %s、行 = %d : 値が無効であるか、値が %d から %d の間ではありません)

Configuration error: Invalid line: file=%s, line=%d (設定エラー : 無効な行 : ファイル = %s、行 = %d)

Configuration error: Missing store header: file=%s, line=%d (設定エラー : ストアヘッダがありません : ファイル = %s、行 = %d)

Configuration error: Mixed mode configuration: file=%s, line=%d (設定エラー : 混合モード設定 : ファイル = %s、行 = %d)



Configuration error: Invalid store parameter: file=%s, line=%d ( 設定エラー : 無効なストアパラメータ : ファイル = %s、行 = %d)

Configuration error: Store definition failed ( 設定エラー : ストア定義に失敗しました )

Configuration error: Unrecognized store type requested. ( 設定エラー : 認識できないストアタイプが要求されました。 )

Configuration error: Filename for store '%s' cannot be empty. ( 設定エラー : ストア '%s' のファイル名を空にすることはできません。 )

Error occurred writing store definition into file. ( ファイルへのストア定義の書き込み中にエラーが発生しました。 )

Configuration error: file=%s, line=%d: ignoring channel '%s' on topic '%s', channel does not exist ( 設定エラー : ファイル = %s、行 = %d : トピック '%s' でチャンネル '%s' を無視します。チャンネルが存在しません )

Configuration error: file=%s, line=%d: ignoring channel '%s' on topic '%s', overlaps with channel '%s' on topic '%s' ( 設定エラー : ファイル = %s、行 = %d : トピック '%s' でチャンネル '%s' を無視します。トピック '%s' のチャンネル '%s' と重複しています )

Configuration error: file=%s, line=%d: ignoring channel '%s', duplicate name ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s' を無視します。名前が重複しています )

Configuration error: file=%s, line=%d: ignoring channel '%s', address of '%s:%d' already defined ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s' を無視します。'%s:%d' のアドレスはすでに定義されています )

Configuration error: file=%s, line=%d: channel '%s', %s ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、%s )

Configuration error: file=%s, line=%d: channel '%s', no address specified. ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、アドレスが指定されていません。 )

Configuration error: file=%s, line=%d: channel '%s', invalid address syntax: port not specified. ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、無効なアドレス構文 : ポートが指定されていません。 )

Configuration error: file=%s, line=%d: channel '%s', invalid address: group must be in the range 224.0.0.0 to 239.255.255.255 ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、無効なアドレス : グループの範囲は 224.0.0.0 から 239.255.255.255 である必要があります )

Configuration error: file=%s, line=%d: channel '%s', interface must address a valid multicast-capable network interface. ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、インタフェースは有効なマルチキャスト可能ネットワークインタフェースをアドレス指定する必要があります。 )

Configuration error: file=%s, line=%d: channel '%s', invalid address: port must be in the range 1 to 65535 ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、無効なアドレス : ポートは 1 から 65535 である必要があります )

Configuration error: file=%s, line=%d: channel '%s', ttl must be in the range 1 to 255 ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、ttl は 1 から 255 の範囲である必要があります )

Configuration error: file=%s, line=%d: channel '%s', priority must be in the range -5 to 5 ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、プライオリティは - 5 から 5 の範囲である必要があります )

Configuration error: file=%s, line=%d: channel '%s', maxrate must be less than 512MB ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、maxrate は 512MB 未満である必要があります )

Configuration error: file=%s, line=%d: channel '%s', maxtime must be greater than 0 ( 設定エラー : ファイル = %s、行 = %d : チャンネル '%s'、maxtime は 0 より大きい必要があります )

Configuration error: file=%s, line=%d: cannot store messages in: %s ( 設定エラー : ファイル = %s、行 = %d : メッセージを %s に保存できません )

Configuration error: file=%s, line=%d: cannot find store: %s ( 設定エラー : ファイル = %s、行 = %d : ストア %s が見つかりません )

Required store param 'type' not specified for store '%s' ( 必須のストアパラメータ 'type' がストア '%s' に指定されていません )

Invalid params at line %d in file '%s' for store '%s' when store type is 'file' %s ( ストアタイプが 'file' %s である場合、ストア '%s' のファイル '%s' にある行 %d は無効です )

Invalid params at line %d in file '%s' for store '%s' when store type is 'dbstore' %s ( ストアタイプが 'dbstore' %s である場合、ストア '%s' のファイル '%s' にある行 %d は無効です )

Invalid params at line %d in file '%s' for store '%s' when store type is 'mstore' %s ( ストアタイプが 'mstore' %s である場合、ストア '%s' のファイル '%s' にある行 %d は無効です )

Store '%s' already defined ( ストア '%s' はすでに定義されています )

Configuration error: Store with similar dbstore\_driver\_url exists, file=%s, line=%d ( 設定エラー : 類似する dbstore\_driver\_url を持つストアが存在します。ファイル = %s、行 = %d )

Configuration error: duplicate file name %s for stores %s and %s ( 設定エラー : ストア %s および %s のファイル名 %s が重複しています )

Configuration warning: file=%s, line=%d: the discardAmount is too small for the selected RV Queue Limit Policy. ( 設定に関する警告 : ファイル = %s、行 = %d : discardAmount は選択した RV キュー制限ポリシーに対して小さすぎます。 ) It is recommended to have at least 10%% of the maxEvents (maxEvents の少なくとも 10%% にすることをお勧めします )

Configuration error: file=%s, line=%d: the discardAmount is too big compared to the maxEvents value. ( 設定エラー : ファイル = %s、行 = %d : maxEvents の値と比較すると、discardAmount は大きすぎます。 ) Defaulting to TIBRVQUEUE\_DISCARD\_NONE policy ( デフォルトの TIBRVQUEUE\_DISCARD\_NONE ポリシーに設定します )

Configuration error: file=%s, line=%d: maxEvents and discardAmount values must be strictly positive for an RV Queue Limit Policy other than TIBRVQUEUE\_DISCARD\_NONE. ( 設定エラー : ファイル = %s、行 = %d : TIBRVQUEUE\_DISCARD\_NONE を除き、RV キュー制限ポリシーに対して maxEvents と discardAmount の値は厳密に正である必要があります。 ) Defaulting to TIBRVQUEUE\_DISCARD\_NONE policy ( デフォルトの TIBRVQUEUE\_DISCARD\_NONE ポリシーに設定します )

Configuration error: file=%s, line=%d: RV Queue Limit Policy '%s' unknown or not supported. ( 設定エラー : ファイル = %s、行 = %d : RV キュー制限ポリシー '%s' が不明であるか、サポートされていません。 ) Defaulting to TIBRVQUEUE\_DISCARD\_NONE policy ( デフォルトの TIBRVQUEUE\_DISCARD\_NONE ポリシーに設定します )

Configuration error: file=%s, line=%d: Error parsing the RV Queue Limit Policy value '%s'. ( 設定エラー : ファイル = %s、行 = %d : RV キュー制限ポリシー値 '%s' の解析中にエラーが発生しました。 ) Defaulting to TIBRVQUEUE\_DISCARD\_NONE policy ( デフォルトの TIBRVQUEUE\_DISCARD\_NONE ポリシーに設定します )

Configuration warning: file=%s, line=%d: The bridge's source destination '%s' is dynamic but has no parent. ( 設定に関する警告 : ファイル = %s、行 = %d : ブリッジのソースデスティネーション '%s' は動的ですが、親を持ちません。 ) The bridge should either be removed or a static parent destination added ( このブリッジを削除するか、静的な親のデスティネーションを追加する必要があります )

---

カテゴリ	コミット要求の書き込み中のエラー。このトランザクションでは、すでに別のエラーが発生している
説明	クライアントアプリケーションがトランザクションをコミットしようとして失敗しました。トランザクションに関連した処理の実行中に、サーバーでエラーが発生したためです。

解決方法	以前のエラーステートメントを調査して、処理に失敗した原因を特定し、問題を修正してからトランザクションを再試行します。
エラー	Error writing commit request, errors already occurred in this transaction. ( コミット要求の書き込み中にエラーが発生しました。このトランザクションでは、別のエラーがすでに発生しています。 )
カテゴリ	設定ファイルへの書き込みエラー
説明	tibemspd は、以下の変更を適用するために、設定ファイルのひとつを更新することができませんでした。
解決方法	tibemspd を起動したユーザーに設定ファイルを変更するアクセス権限があること、およびデバイスに十分なディスク領域があることを確認してください。
エラー	<p>Error occurred saving acl information ( acl 情報の保存中にエラーが発生しました )</p> <p>Error occurred saving bridges information ( ブリッジ情報の保存中にエラーが発生しました )</p> <p>Error occurred saving durables information ( 持続情報の保存中にエラーが発生しました )</p> <p>Error occurred saving factories information ( ファクトリ情報の保存中にエラーが発生しました )</p> <p>Error occurred saving file '%s' ( ファイル '%s' の保存中にエラーが発生しました )</p> <p>Error occurred saving group information ( グループ情報の保存中にエラーが発生しました )</p> <p>Error occurred saving %s information ( %s 情報の保存中にエラーが発生しました )</p> <p>Error occurred saving main configuration file '%s' ( 主要設定ファイル '%s' の保存中にエラーが発生しました )</p> <p>Error occurred saving routes information ( ルート情報の保存中にエラーが発生しました )</p> <p>Error occurred saving tibrvcm information ( tibrvcm 情報の保存中にエラーが発生しました )</p> <p>Error occurred while updating main configuration file '%s'. ( 主要設定ファイル '%s' の更新中にエラーが発生しました。 ) Configuration has not been saved. ( 設定は保存されていません。 )</p> <p>Error occurred writing bridges into file. ( ファイルへのブリッジの書き込み中にエラーが発生しました。 )</p>

Error occurred writing destination '%s' into file ( ファイルへのデスティネーション '%s' の書き込み中にエラーが発生しました )

Error occurred writing factory into file. ( ファイルへのファクトリの書き込み中にエラーが発生しました。 )

Error occurred writing group '%s' into file ( ファイルへのグループ '%s' の書き込み中にエラーが発生しました )

Error occurred writing into the file '%s'. ( ファイル '%s' への書き込み中にエラーが発生しました。 )

Error occurred writing route into file. ( ファイルへのルートの書き込み中にエラーが発生しました。 )

I/O error occurred saving bridge information ( ブリッジ情報の保存中に I/O エラーが発生しました )

I/O error occurred saving group information ( グループ情報の保存中に I/O エラーが発生しました )

I/O error occurred saving route information ( ルート情報の保存中に I/O エラーが発生しました )

I/O error occurred writing into file '%s' ( ファイル '%s' への書き込み中に I/O エラーが発生しました。 )

---

カテゴリ	ストアファイルへの書き込みエラー
説明	tibemspd は、ストアファイルのひとつにデータを書き込むことができませんでした。
解決方法	ストアファイルを含むディレクトリがマウントされていて tibemspd にアクセス可能なこと、およびデバイスに十分な空き領域があることを確認してください。
エラー	<p>Failed writing block data to '%s': %s ('%s' へのブロックデータの書き込みに失敗しました : %s)</p> <p>Failed writing message to '%s': I/O error or out of disk space. ('%s' へのメッセージの書き込みに失敗しました : I/O エラーまたはディスク領域不足です。)</p> <p>Failed writing purge state for queue '%s': I/O error or out of disk space. ( キュー '%s' へのページ状態の書き込みに失敗しました : I/O エラーまたはディスク領域不足です。 )</p> <p>Failed writing purge state for topic consumer: I/O error or out of disk space. ( トピックコンシューマのページ状態の書き込みに失敗しました : I/O エラーまたはディスク領域不足です。 )</p>

Exception trying to create confirm record, %s. ( 確認レコードを作成しようとして、例外が発生しました。 %s。 )

Exception trying to create message from store: %s ( ストアからメッセージを作成しようとして、例外が発生しました : %s。 )

Exception trying to create transaction record. ( トランザクションレコードを作成しようとして、例外が発生しました。 )

Exception trying to create valid messages record, %s. ( 有効なメッセージレコードを作成しようとして、例外が発生しました。 %s。 )

Exception trying to export message to RV. (RV にメッセージをエクスポートしようとして、例外が発生しました。 )

Failed writing message to '%s': %s. ( メッセージの '%s' への書き込みに失敗しました : %s)

Exception writing transaction commit record: %s. ( トランザクションコミットレコードの書き込み中に例外が発生しました : %s。 )

Exception writing transaction rollback record: %s. ( トランザクションロールバックレコードの書き込み中に例外が発生しました : %s。 )

Exception writing transaction prepare record: %s. ( トランザクション準備レコードの書き込み中に例外が発生しました : %s。 )

Failure deleting old version of transaction record: %s. ( トランザクションレコードの古いバージョンの削除に失敗しました : %s。 )

Failed deleting '%s' record from %s: %s ('%s' レコードを %s から削除できませんでした : %s)

カテゴリ	データベースストアのセットアップでエラーが発生しました
説明	データベースストアのセットアップで、実行時にエラーが発生します
解決方法	各種レコードの書き込み、削除、読み取りの間に障害が発生する場合は、データベースサーバーのベンダーおよびデータベース管理者に確認してください。データベースストアを開いているときに障害が発生する場合は、データベース管理者にアクセス権限の内容とデータベースが存在することを確認してください。すべてのストアがデータベースストアである場合に FT のセットアップ中に障害が発生する場合は、データベースサーバーのベンダーかデータベースの管理者に確認してください。両方がアクティブである場合は、両方のサーバーをシャットダウンして、問題を調査することをお勧めします。
エラー	Unable to open store [%s]: [ ESTATUS = %d, ERRSTR = %s ] ( ストア [%s] を開くことができませんでした : [ ESTATUS = %d、ERRSTR = %s ])

Failed to store message record in store [%s]: [ ESTATUS = %d, ERRSTR = %s ] ( ス  
トア [%s] にメッセージレコードを保存できませんでした : [ ESTATUS = %d、  
ERRSTR = %s ] )

Failed to write ack record in store [%s]: [ ESTATUS = %d, ERRSTR = %s ] ( ストア  
[%s] に確認応答レコードを書き込めませんでした : [ ESTATUS = %d、ERRSTR  
= %s ] )

Failed to write txn record in store [%s]: [ ESTATUS = %d, ERRSTR = %s ] ( ストア  
[%s] に txn レコードを書き込めませんでした : [ ESTATUS = %d、ERRSTR = %s ] )

Failed to update txn record in store [%s]: [ ESTATUS = %d, ERRSTR = %s ] ( スト  
ア [%s] で txn レコードを更新できませんでした : [ ESTATUS = %d、ERRSTR =  
%s ] )

No memory to create no hold list for valid msgs record ( メモリ不足のため、有効  
なメッセージレコードの保持されていないリストを作成できません )

No memory to create hold list for valid msgs record ( メモリ不足のため、有効な  
メッセージレコードの保持リストを作成できません )

No memory to create held list for valid msgs record ( メモリ不足のため、有効な  
メッセージレコードが保持されているリストを作成できません )

Failed to write valid msg record in store [%s]: [ ESTATUS = %d, ERRSTR = %s ]  
( ストア [%s] に有効なメッセージレコードを書き込めませんでした : [ ESTATUS  
= %d、ERRSTR = %s ] )

Failed to update msg record with record id [% PRINTF\_LL\_FMT d] in store [%s]:  
[ ESTATUS = %d, ERRSTR = %s ] ( ストア [%s] で msg レコードを更新できませ  
んでした : [ ESTATUS = %d、ERRSTR = %s ] )

Failed to delete %s record id = % PRINTF\_LL\_FMT d : [ ESTATUS = %d, ERRSTR =  
%s ] (%s レコード ID = % PRINTF\_LL\_FMT d を削除できませんでした :  
[ ESTATUS = %d,ERRSTR = %s ] )

Failed to read message with store id = % PRINTF\_LL\_FMT d: [ ESTATUS = %d,  
ERRSTR = %s ] ( ストア ID = % PRINTF\_LL\_FMT d のメッセージを読み取れませ  
んでした : [ ESTATUS = %d、ERRSTR = %s ] )

Failed to initialize dbstore [%s]: [ ERRSTR = %s ] ( データベースストア [%s] を初  
期化できませんでした : [ ERRSTR = %s ] )

Failed to open store [%s], error = %s ( ストア [%s] を開けませんでした。エラー =  
%s )

Unable to restore %s records from store [%s]: [ ESTATUS = %d, ERRSTR = %s ]  
( %s 個のレコードをストア [%s] から復元できません : [ ESTATUS = %d、  
ERRSTR = %s ] )

Failed to delete meta record: [ ESTATUS = %d, ERRSTR = %s ] ( メタレコードを削  
除できませんでした : [ ESTATUS = %d、ERRSTR = %s ] )



Failed to beginTransaction: [ ESTATUS = %d, ERRSTR = %s ] ( トランザクションを開始できませんでした : [ ESTATUS = %d、ERRSTR = %s ])

Failed to read message with store id = % PRINTF\_LL\_FMT d: [ ESTATUS = %d, ERRSTR = %s ] ( ストア ID = % PRINTF\_LL\_FMT d のメッセージを読み取れませんでした : [ ESTATUS = %d、ERRSTR = %s ])

Store [%s] locked by server %s ( ストア [%s] はサーバー %s によってロックされています )

Store [%s] cannot be locked by server %s ( ストア [%s] をサーバー %s によってロックすることはできません )

Failed to store txn record: [ txn id = % PRINTF\_LL\_FMT d, ESTATUS = %d ] (txn レコードを保存できませんでした : [ txn id = % PRINTF\_LL\_FMT d、ESTATUS = %d ])

Failed to update txn record: [ txn record id = % PRINTF\_LL\_FMT d, ESTATUS = %d ] (txn レコードを更新できませんでした : [ txn record id = % PRINTF\_LL\_FMT d、ESTATUS = %d ])

Exception while processing msg from database store [%s], error = %d ( データベースストア [%s] からのメッセージを処理しているときに、例外が発生しました。エラー = %d)

Failed to write meta record: [ ESTATUS = %d, ERRSTR = %s ] ( メタレコードを書き込めませんでした : [ ESTATUS = %d、ERRSTR = %s ])

Failed to update meta record: [ ESTATUS = %d, ERRSTR = %s ] ( メタレコードを更新できませんでした : [ ESTATUS = %d、ERRSTR = %s ])

Failed to write connection record: error = %d ( 接続レコードを書き込めませんでした : エラー = %d)

Failed to write session record: error = %d ( セッションレコードを書き込めませんでした : エラー = %d)

Failed to write consumer record: error = %d ( コンシューマレコードを書き込めませんでした : エラー = %d)

Failed to write producer record: error = %d ( プロデューサレコードを書き込めませんでした : エラー = %d)

Failed to write zone record: error = %d ( ゾーンレコードを書き込めませんでした : エラー = %d)

Failed to update connection record: error = %d ( 接続レコードを更新できませんでした : エラー = %d)

Failed to update consumer record: error = %d ( コンシューマレコードを更新できませんでした : エラー = %d)



Failed to write purge record: [ ESTATUS = %d, ERRSTR = %s ] ( パージレコードを書き込めませんでした : [ ESTATUS = %d、ERRSTR = %s ] )

Commit Transaction Failed [ ESTATUS = %d, ERRSTR = %s ] ( トランザクションのコミットに失敗しました [ ESTATUS = %d、ERRSTR = %s ] )

No Memory to create lock manager: Store [%s] cannot be locked by server %s ( メモリ不足のため、ロックマネージャを作成できません : ストア [%s] をサーバー %s によってロックすることはできません )

Could not find system record for store [%s] ( ストア [%s] のシステムレコードを見つけることができませんでした )

---

カテゴリ	システムリソースを超過しました。
説明	システムリソースは、サーバーアクティビティのタイムリーな処理に適していません。
解決方法	指定されたリソースを増やすか、サーバー上の作業負荷を減らします。
エラー	Slow clock tick %d, delayed messaging and timeouts may occur. ( クロックの進み方 %d が遅くなっているため、メッセージングとタイムアウトに遅延が生じる可能性があります。 )

---

カテゴリ	TCP ポートを開けない
説明	tibemspd は、TCP ポートを開くことができませんでした。
解決方法	ポートを使用しているプロセスをシャットダウンするか、サーバーの tibemspd.conf ファイルの 'listen' パラメータの値を、使用中でないポートに変更します。
エラー	Binding connection to TCP port %d failed:%d (%s). (TCP ポート %d に対する接続のバインドに失敗しました : %d (%s)。 )

---

カテゴリ	ファイルアクセスエラー
説明	tibemspd は指定されたファイルに適切にアクセスできませんでした。
解決方法	パス名が正確でディレクトリが存在すること、tibemspd を起動したユーザーが指定のディレクトリとパスの読み取りアクセス権限を持っていること、ファイルが存在するかまたは tibemspd によって作成可能であること、ファイルが別の tibemspd または別のプロセスによって使用されていないことを確認してください。

エラー	<p>Configuration file '%s' not found. ( 設定ファイル '%s' が見つかりません。 )</p> <p>Failed to create file '%s' ( ファイル '%s' の作成に失敗しました )</p> <p>failed to open file '%s'. ( ファイル '%s' を開くことができませんでした。 )</p> <p>failed to open log file '%s'. ( ログファイル '%s' を開くことができませんでした。 )</p> <p>Failed to read message from store. ( ストアからのメッセージの読み取りに失敗しました。 )</p> <p>Failed to rename file %s into %s: %s ( ファイル %s の名前を %s に変更できませんでした : '%s' )</p> <p>Unable to open metadata file '%s', error '%s'. ( メタデータファイル '%s' を開くことができませんでした。エラー '%s'。 )</p> <p>Unable to open metadata file '%s', file may be locked. ( メタデータファイル '%s' を開くことができませんでした。ファイルがロックされている可能性があります。 )</p> <p>Unable to open store file '%s', error '%s'. ( ストアファイル '%s' を開くことができませんでした。エラー '%s'。 )</p> <p>Unable to open store file '%s', file may be locked. ( ストアファイル '%s' を開くことができませんでした。ファイルがロックされている可能性があります。 )</p> <p>Unable to preallocate storage file '%s'. ( ストレージファイル '%s' を事前に割り当てることはできません。 )</p> <p>I/O error occurred reading from the file '%s'. ( ファイル '%s' から読み取り中に I/O エラーが発生しました。 )</p> <p>Exiting on non-retryable disk error: %d ( 再試行不能なディスクエラーが発生したため、終了します。エラー : %d )</p> <p>Exception trying to read message from store. ( ストアからメッセージを読み取ろうとして、例外が発生しました。 )</p> <p>Error during file close of '%s' - %d. ('%s' のファイルを閉じているときにエラーが発生しました - %d。 )</p>
カテゴリ	FIPS 140-2 モードのエラー
説明	FIPS 140-2 準拠モードでサーバーを起動または実行しているときにエラーが発生しました。
解決方法	SSL 関連パラメータの設定を確認して、互換性のない暗号または処理が要求されていないかを確認します。
エラー	Cannot specify ldap_tls_cipher_suite in FIPS 140-2 mode. (FIPS 140-2 モードで ldap_tls_cipher_suite を指定することはできません。 )

Cannot specify ldap\_tls\_rand\_file in FIPS 140-2 mode. (FIPS 140-2 モードで ldap\_tls\_rand\_file を指定することはできません。)

Cannot specify SSL cipher list in FIPS 140-2 mode. (FIPS 140-2 モードで SSL 暗号リストを指定することはできません。)

Cannot specify random data source file in FIPS 140-2 mode. (FIPS 140-2 モードで ランダムデータソースファイルを指定することはできません。)

Cannot specify ssl\_dh\_size in FIPS 140-2 mode. (FIPS 140-2 モードで ssl\_dh\_size を指定することはできません。)

Cannot specify ssl\_server\_ciphers in FIPS 140-2 mode. (FIPS 140-2 モードで ssl\_server\_ciphers を指定することはできません。)

Cannot specify ssl\_rand\_file in FIPS 140-2 mode. (FIPS 140-2 モードで ssl\_rand\_file を指定することはできません。)

---

カテゴリ	一般的なマルチキャストのステータスコードとエラー
説明	サーバーおよびマルチキャストデーモンで発生する可能性のある一般的なマルチキャストとエラー。
解決方法	マルチキャストデーモンとサーバーの設定に加え、ネットワークの動作状態も確認します。
エラー	<p>PGM ERROR: %s - %s (%d) (PGM エラー : %s - %s (%d) )</p> <p>PGM ERROR: channel=%q%s¥q - %s (%d) (PGM エラー : チャネル = ¥q%s¥q - %s (%d) )</p> <p>Error setting PGM parameter %s=%u: %s (%d) (PGM パラメータ %s=%u の設定中にエラーが発生しました : %s (%d) )</p> <p>Error setting PGM parameter %s=¥q%s¥q: %s (%d) (PGM パラメータ %s=¥q%s¥q の設定中にエラーが発生しました : %s (%d) )</p> <p>Error getting PGM parameter ¥q%s¥q: %s (%d) (PGM パラメータ ¥q%s¥q の取得中にエラーが発生しました : %s (%d) )</p> <p>Error getting PGM statistic ¥q%s¥q: %s (%d) (PGM 統計情報 ¥q%s¥q の取得中にエラーが発生しました : %s (%d) )</p> <p>Received an invalid EMS Message. ( 無効な EMS メッセージを受け取りました。)</p> <p>Received a message spanning mulitple fragments. ( 複数のフラグメントにわたるメッセージを受け取りました。)</p>

PGM Session was reset for channel %q%s%q, PGM seqno=%PRINTF\_LLFMt d, code=%c ( チャンネル %q%s%q の PGM セッションがリセットされました。PGM seqno=%PRINTF\_LLFMt d、コード=%c)

Stopped receiving on channel %q%s%q ( チャンネル %q%s%q での受信を停止しました )

Started receiving on channel %q%s%q ( チャンネル %q%s%q での受信を開始しました )

Error receiving on channel %q%s%q ( チャンネル %q%s%q での受信中にエラーが発生しました )

Stopped sending on channel %q%s%q ( チャンネル %q%s%q での送信を停止しました )

Started sending on channel %q%s%q ( チャンネル %q%s%q での送信を開始しました )

Error creating sender on channel %q%s%q: %s ( チャンネル %q%s%q でセNDERを作成中にエラーが発生しました : %s)

カテゴリ ステータス駆動型の内部エラー

説明 サーバーが内部的な不整合を検出しました。

解決方法 エラーステートメントと環境の説明を TIBCO サポートに送信します。

エラー **\*\*Error\*\* unable to process message, error = %s** (\*\* エラー \*\* メッセージを処理できません。エラー = %s)

Admin user not found during initialization ( 初期化中に admin ユーザーが見つかりませんでした )

Error bridging transacted data message, '%s'. ( トランザクションデータメッセージのブリッジ中にエラーが発生しました。'%s'。 )

Error processing xa commit request, %s. connID=% PRINTF\_LLFMt d %s (xa コミット要求の処理中にエラーが発生しました。%s。connID=% PRINTF\_LLFMt d %s)

Error processing xa end - transaction marked ROLLBACKONLY, %s. connID=% PRINTF\_LLFMt d sessID=% PRINTF\_LLFMt d %s (xa 終了の処理中にエラーが発生しました。トランザクションは、ROLLBACKONLY とマークされています。%s。connID=% PRINTF\_LLFMt d sessID=% PRINTF\_LLFMt d %s)

Error processing xa prepare request, %s. connID=% PRINTF\_LLFMt d %s (xa 準備要求の処理中にエラーが発生しました。%s。connID=% PRINTF\_LLFMt d %s)

Error processing xa rollback request, %s. connID=% PRINTF\_LLFMt d %s (xa ロールバック要求の処理中にエラーが発生しました。%s。connID=% PRINTF\_LLFMt d %s)

Error decoding sequence data in xa rollback request. connID=% PRINTF\_LL\_FMT d %s (xa ロールバック要求内でシーケンスデータのデコード中にエラーが発生しました。connID=% PRINTF\_LL\_FMT d %s)

Error decoding sequence data in route ack response. ( ルート確認応答内でシーケンスデータのデコード中にエラーが発生しました。 )

Unable to create internal session ( 内部セッションを作成できません )

Problem setting flow stall recover message on route queue:%s: %s ( ルートキューに対して、フローストール回復メッセージを設定中に問題が発生しました : %s : %s )

Failed to handle connection initialization: %s. ( 接続の初期化の処理に失敗しました : %s。 )

Problem trying to recover routed consumer for queue %s: setting recover message. ( キュー %s のルーティング先コンシューマを回復しようとして、問題が発生しました : 回復メッセージを設定します。 ) Error: %s ( エラー : %s )

Failed to send the flow stall recover request: %s. ( フローストール回復要求の送信に失敗しました : %s。 )

Unable to handle transacted data message, '%s'. ( トランザクションデータメッセージの処理に失敗しました。 '%s'。 )

Unable to handoff connection init message: %s. ( 接続の初期化メッセージをハンドオフすることができませんでした : %s。 )

Unable to initialize fault tolerant connection, remote server returned '%s' ( フォールトトレランス接続を初期化できません。リモートサーバーから '%s' が返されました )

Unable to process producer message, failed to add sender name, error=%s. ( プロデューサメッセージを処理できません。送信者名の追加に失敗しました。エラー = %s。 )

Unable to process sequence for message. ( メッセージのシーケンスを処理できません。 )

Unable to send recover ack on flow stall: %s ( フローストールの回復確認応答を送信できません : %s )

Handling of route flow stall recovery request from %s failed: unable to get message property %s: %s (%s からのルートフローストール回復要求の処理に失敗しました : メッセージプロパティ %s を取得できません : %s )

Handling of route flow stall recovery request failed: Unable to get message properties:%s ( ルートフローストール回復要求の処理に失敗しました : メッセージプロパティを取得できません : %s )

Failed to send acknowledge to the stall recover request of server %s, will try later.  
(サーバー %s のストール回復要求に対して確認応答の送信に失敗しました。)  
Error: %s (エラー : %s)

failed to send recover ack on stalled flow: invalid consumer (ストールフローに対する回復の確認応答の送信に失敗しました : 無効なコンシューマです)

unable to create recovered connection, status: %s (回復した接続を作成できません。ステータス : %s)

Exception creating purge record. (パージレコードの作成中に例外が発生しました。)

Exception creating zone. (ゾーンの作成中に例外が発生しました。)

Exception creating zone: adding zone to state. (ゾーンの作成中に例外が発生しました : 状態にゾーンを追加します。)

Exception in startup, exiting. (起動時に例外が発生しました。終了します。)

Exception preparing message for client send. (クライアントに送信するためのメッセージの準備中に例外が発生しました。)

Exception sending flow recover acknowledge (フロー回復確認応答の送信中に例外が発生しました)

Exception sending routing information to %s - %s (%s へのルーティング情報の送信中に例外が発生しました - %s)

Exception sending session init response (セッション初期化応答の送信中に例外が発生しました)

Exception sending queue acknowledge response to %s: %s (%s へのキュー確認応答の送信中に例外が発生しました : %s)

Exception trying to initialize connection. (接続を初期化しようとして、例外が発生しました。)

Exception trying to initialize connection, can't send response: %s (接続を初期化しようとして、例外が発生しました。応答を送信できません : %s)

Exception trying to initialize route. (ルートを初期化しようとして、例外が発生しました。)

Exception trying to initialize route '%s' configured durables: %s (ルート '%s' を初期化しようとして、例外が発生しました : %s)

Exception trying to process message, '%s'. (メッセージを処理しようとして、例外が発生しました。'%s'。)

Exception trying to process message from store. (ストアからのメッセージを処理しようとして、例外が発生しました。)

Failure queuing incoming message for processing: %s. ( 処理用に受信メッセージをキューに入れようとして失敗しました : %s。 )

Failure queuing message for removal from system: %s. ( システムから削除するためにメッセージをキューに入れようとして失敗しました : %s。 )

Failure queuing message to add to dead queue: %s. ( 停止中のキューに追加するためにメッセージをキューに入れようとして失敗しました : %s。 )

Failure discarding topic overflow: %s. ( トピックオーバーフローの破棄に失敗しました : %s。 )

Failure processing system request. ( システム要求の処理に失敗しました。 )

Failure processing transaction message. ( トランザクションメッセージの処理に失敗しました。 )

Failure bridging incoming message: %s. ( 受信メッセージのブリッジに失敗しました : %s。 )

Failure verifying uniqueness of routed message: %s. ( ルーティング先メッセージの一意性の検証に失敗しました : %s。 )

Failure scheduling message hold release: %s. ( 保持したメッセージのリリースのスケジュール作成に失敗しました : %s。 )

Exception adding message write context: %s. ( メッセージ書き込みコンテキストを追加しようとして例外が発生しました : %s。 )

%s: Failure processing multicast request: %s (%s : マルチキャスト要求の処理に失敗しました : %s)

%s: Failure sending multicast request response: %s (%s : マルチキャスト要求の応答の送信に失敗しました : %s)

%s: Failure processing multicast status: %s (%s : マルチキャストステータスの処理に失敗しました : %s)

%s: Failure sending multicast status response: %s (%s : マルチキャストステータスの応答の送信に失敗しました : %s)

%s: Failure sending multicast configuration: %s (%s : マルチキャスト設定の送信に失敗しました : %s)

Failure sending multicast message on channel '%s': %s ( チャンネル '%s' でマルチキャストメッセージの送信に失敗しました : %s)

Failure enqueueing multicast message on channel '%s': %s ( チャンネル '%s' でマルチキャストメッセージのエンキューに失敗しました : %s)

Failure starting multicast engine: %s ( マルチキャストエンジンの開始に失敗しました %s)

Failure starting multicast channel '%s': %s ( マルチキャストチャネル '%s' の開始に失敗しました : %s)

Failure posting multicast channel '%s' wake event: %s ( マルチキャストチャネル '%s' のウェイクイベントのポストに失敗しました : %s)

Failed preparing message for writing: %s ( 書き込み用のメッセージの準備に失敗しました : %s)

Failed discarding local transaction: %s ( ローカルトランザクションの破棄に失敗しました : %s)

Abandoning transaction record due to IO failure. (IO 障害のため、トランザクションレコードを中止しています。)

Error sending acknowledgment to route '%s': %s ( ルート '%s' に確認応答を送信しているときにエラーが発生しました : %s)

Error processing acknowledgments from route '%s': %s ( ルート '%s' から確認応答を処理しているときにエラーが発生しました : %s)

---

カテゴリ	無効な接続
説明	tibemspd がルートを経由して別の tibemspd にメッセージ配信の再確立を試みましたが、そのルートへの接続を検出できなかったことを示す警告です。
解決方法	メッセージをコンシュームするか、キューからメッセージを削除することによって、tibemspd のメモリ要件を低くします。または、コンピュータ上の他のプロセスをシャットダウンするか、コンピュータのメモリを増設することによって、tibemspd が利用可能なメモリ容量を増やします。
エラー	Recovery flow stall for destination %s failed: invalid route connection ( デスティネーション %s のフローストールの回復に失敗しました : 無効なルート接続です )

---

カテゴリ	無効なデスティネーション
説明	アプリケーションが、無効なデスティネーション名を使用しようとしています。
解決方法	適切なデスティネーション名を使用するようアプリケーションコードを変更します。
エラー	<p>%s: create %s failed: Not permitted to use reserved queue [%s]. (%s : %s の作成に失敗しました : 予約済みのキュー [%s] を使用することはできません。)</p> <p>%s: %s failed: illegal to use wildcard %s [%s]. (%s : %s が失敗しました : ワイルドカード %s [%s] を使用することはできません。)</p>



%s: %s failed: %s [%s] not configured. (%s : %s が失敗しました : %s [%s] は設定されていません。)

At least one bridge is referencing %s [%s] as a target. (少なくともひとつのブリッジが %s [%s] をターゲットとして参照しています。) This destination does not exist and there is no parent that would allow its dynamic creation. (このデスティネーションは存在せず、その動的作成を行える親が存在しません。) The destination has been forcefully created. (デスティネーションは強制的に作成されました。) To avoid this, the bridge(s) referencing this target should be destroyed. (これを回避するには、このターゲットを参照しているブリッジを破棄する必要があります。)

Use of '\$' destination prefix is not supported [%s %s]. ('\$ デスティネーションプリフィックスの使用はサポートされていません [%s %s]。)

---

カテゴリ	無効なリスン指定
説明	サーバーは、tibemsd.conf ファイル内のリスンパラメータを解析できませんでした。
解決方法	マニュアルで指定されている [protocol]://[url] の形式に一致するように、リスンパラメータを修正します。
エラー	Invalid listen specification: '%s'. (無効なリスン指定 : '%s'。) Invalid request to create temporary destination. (テンポラリデスティネーションの作成要求は無効です。)

---

カテゴリ	無効なセッション
説明	tibemsd が、現在は存在しないセッションを参照する要求を受信しました。
解決方法	エラーおよびエラーが発生した状況の詳細を TIBCO サポートに送信します。
エラー	Cannot find session for ack (確認応答のためのセッションが見つかりません) Cannot find session for ack range (確認応答範囲のためのセッションが見つかりません) %s: destroy %s failed: invalid session id=%d. (%s : %s の破棄に失敗しました : 無効なセッション ID = %d です。) Unable to destroy session, invalid session. (セッションを破棄できません。無効なセッションです。)

Invalid session in commit request. (コミット要求内に無効なセッションがあります。)

Invalid commit request. (コミット要求が無効です。)

Invalid session trying to update(%d) tx record. (無効なセッションが (%d) tx レコードを更新しようとしています。)

Invalid session in commit transaction record. (コミットトランザクションレコード内に無効なセッションがあります。)

Invalid session in recover request. (回復要求内に無効なセッションがあります。)

Invalid session in rollback request. (ロールバック要求内に無効なセッションがあります。)

Invalid session in xa end request. connID=% PRINTF\_LL\_FMT d (xa 終了要求内に無効なセッションがあります。接続 ID =% PRINTF\_LL\_FMT d)

Invalid session in xa start request. connID=% PRINTF\_LL\_FMT d (xa 開始要求内に無効なセッションがあります。接続 ID =% PRINTF\_LL\_FMT d)

カテゴリ	LDAP エラー - 常に LDAP エラーを表示する必要がある
説明	外部 LDAP サーバーを使用してクライアントの userid とパスワードの認証を試みましたが、失敗しました。
解決方法	メッセージングサーバーから出力されたエラーコードを調査し、外部 LDAP サーバーのマニュアルを参照してください。
エラー	<p>Filter '%s' contains an illegal type substitution character, only %%s is allowed (フィルタ '%s' に不正なタイプの置換文字が含まれています。使用できるのは %%s のみです)</p> <p>Filter '%s' contains too many occurrences of %%s, max allowed is: %d (フィルタ '%s' に使用されている %%s の数が多過ぎます。最大許容数: %d)</p> <p>Filter '%s' too long, max length is %d characters (フィルタ '%s' が長過ぎます。最大長は %d 文字です)</p> <p>Invalid search scope: %s (無効な検索範囲: %s)</p> <p>LDAP Configuration error: file=%s, line=%d: invalid property value (LDAP 設定エラー: ファイル=%s、行=%d: プロパティ値が無効です)</p> <p>LDAP is not present (LDAP が存在しません)</p> <p>LDAP search resulted %d hits. (LDAP の検索結果は %d 件でした。)</p> <p>ldap_url_parse failed, returned: %d (ldap_url_parse が失敗しました。戻り値: %d)</p>

Lookup of group '%s' produced incorrect or no results (グループ '%s' の検索結果が間違っているか、検索結果がありません)

Missing LDAP URL (LDAP URL が指定されていません)

Missing %s parameter (%s パラメータが指定されていません)

Zero entries returned from getting attributes for group '%s': (グループ '%s' の属性の取得によって返されたエントリはありません。)

Failed adding user '%s' into LDAP user cache (LDAP ユーザーキャッシュへのユーザー '%s' の追加に失敗しました)

---

カテゴリ	ライセンスに関する警告
説明	サーバーが、ライセンス違反を検出しました。
解決方法	このエラーは、評価版のサーバーまたは組み込み型でのみ発生します。このエラーを修正するには、評価版のサーバーを実稼動版に置き換えるか、または組み込み型の提供元のベンダーに連絡してください。
エラー	License violation: %s. (ライセンス違反 : %s。)

---

カテゴリ	設定が指定されていない
説明	必須な属性が設定されていません。
解決方法	属性に値を指定できるように <code>ibemsd.conf</code> ファイルを変更します。
エラー	Configuration error with metadata database. (メタデータデータベースに設定エラーがあります。) Configuration error with storage databases. (ストレージデータベースに設定エラーがあります。)

---

カテゴリ	トランザクションが指定されていない
説明	クライアントアプリケーションが、 <code>tibemsd</code> の現在のトランザクションのリストにないトランザクションの状態の変更を試みました。

**解決方法**    tibemspd トレースログを調べて、管理者によってトランザクションがコミットされたか、またはロールバックされたかを確認してください。どちらにも該当しない場合、クライアントコードを調べて、クライアントコードまたはそのトランザクションマネージャがトランザクション処理を正しい順序で呼び出しているかどうかを確認してください。

**エラー**    Cannot find transaction referred to transaction record update(%d) request. connID=% PRINTF\_LL\_FMT d %s ( トランザクションレコード更新 (%d) 要求の参照先のトランザクションが見つかりません。connID=% PRINTF\_LL\_FMT d %s)

Cannot find transaction referred to in xa commit request. connID=% PRINTF\_LL\_FMT d %s (xa コミット要求に、参照先のトランザクションが見つかりません。connID=% PRINTF\_LL\_FMT d %s)

Cannot find transaction referred to in xa prepare request. connID=% PRINTF\_LL\_FMT d %s (xa 準備要求に、参照先のトランザクションが見つかりません。connID=% PRINTF\_LL\_FMT d %s)

Cannot find transaction referred to in xa rollback request. connID=% PRINTF\_LL\_FMT d %s (xa ロールバック要求に、参照先のトランザクションが見つかりません。connID=% PRINTF\_LL\_FMT d %s)

Received prepare request for transaction already prepared. connID=% PRINTF\_LL\_FMT d %s (すでに準備済みのトランザクションに対する準備要求を受信しました。connID=% PRINTF\_LL\_FMT d %s)

Cannot find transaction referred to in xa start (resume) request. connID=% PRINTF\_LL\_FMT d sessID=% PRINTF\_LL\_FMT d %s (xa 開始 (再開) 要求に、参照先のトランザクションが見つかりません。connID=% PRINTF\_LL\_FMT d sessID=% PRINTF\_LL\_FMT d %s)

Cannot find transaction referred to in xa start (join) request. connID=% PRINTF\_LL\_FMT d sessID=% PRINTF\_LL\_FMT d %s (xa 開始 (参加) 要求に、参照先のトランザクションが見つかりません。connID=% PRINTF\_LL\_FMT d sessID=% PRINTF\_LL\_FMT d %s)

Cannot find transaction referred to in xa end request. connID=% PRINTF\_LL\_FMT d sessID=% PRINTF\_LL\_FMT d %s (xa エンド要求に、参照先のトランザクションが見つかりません。connID=% PRINTF\_LL\_FMT d sessID=% PRINTF\_LL\_FMT d %s)

---

**カテゴリ**    Mstore のエラー

**説明**    Mstore データベースファイルの使用中にエラーが発生しました。

**エラー**    Unable to open store %s: %s: %d (%s). ( ストア %s を開くことができません : %s:%d(%s)。 )

Wrong schema version. (スキーマバージョンが不正です。) Found %d, expected %d. (%d が見つかりましたが、必要なものは %d でした。)

Schema creation failed: '%s' error: %d %s (スキーマの作成に失敗しました: '%s' エラー: %d %s)

Unable to reset a statement (%s): %s. (ステートメントをリセットできません (%s) : %s。)

Unable to step a statement (%s): %s: %d. (ステートメントをステップ実行できません (%s) : %s : %d。)

Store %s: %s: bind fail: %d. (ストア %s : %s : バインドに失敗しました: %d。)

Store %s: Fail retrieving consumer interest: %d. (ストア %s : 対象コンシューマの取得に失敗しました: %d。)

Store %s: Fail retrieving msg interest info: %s. (ストア %s : 対象メッセージ情報の取得に失敗しました: %s。)

Store %s: Fail writing transaction record: %s. (ストア %s : トランザクションレコードの書き込みに失敗しました: %s。)

Store %s: Fail reading data: %s. (ストア %s : データの読み取りに失敗しました: %s。)

Store %s: Fail reading topic message: %s. (ストア %s : トピックメッセージの読み取りに失敗しました: %s。)

Store %s: Fail marking topic message non-pending: %s. (ストア %s : トピックメッセージへの非保留中のマーキングに失敗しました: %s。)

Store %s: Fail reading next topic message: %s. (ストア %s : 次のトピックメッセージの読み取りに失敗しました: %s。)

Store %s: Fail reading queue message: %s. (ストア %s : キューメッセージの読み取りに失敗しました: %s。)

Store %s: Fail getting next queue message: %s. (ストア %s : 次のキューメッセージの取得に失敗しました: %s。)

Store %s: Fail marking queue message non-pending: %s. (ストア %s : キューメッセージへの非保留中のマーキングに失敗しました: %s。)

Store %s: Fail writing transaction info: %s. (ストア %s : トランザクション情報の書き込みに失敗しました: %s。)

Store %s: Fail deleting transaction acks: %s. (ストア %s : トランザクション確認応答の削除に失敗しました: %s。)

Store %s: Fail recording transaction msg: %s. (ストア %s : トランザクションメッセージの記録に失敗しました: %s。)

Store %s: Fail recording transaction acks: %s. (ストア %s : トランザクション確認  
応答の記録に失敗しました : %s。)

Store %s: Fail deleting ack: %s. (ストア %s : 確認応答の削除に失敗しました :  
%s。)

Store %s: Fail completing transaction: %s. (ストア %s : トランザクションの完了  
に失敗しました : %s。)

Store %s: Too many entries in memory message interest. (ストア %s : 対象のメモ  
リーメッセージ内のエントリが多すぎます。)

Store %s: Invalid message interest for destination % PRINTF\_LLFMT d. (ストア  
%s : デスティネーション % PRINTF\_LLFMT d の対象のメッセージが無効です。)

Store %s: Invalid destination read. (ストア %s : 無効なデスティネーションの読  
み取りです。)

Store %s: Failure restoring transaction: %s. (ストア %s : トランザクションの復元  
に失敗しました : %s。)

Store %s: Failure restoring transaction msg: %s. (ストア %s : トランザクション  
メッセージの復元に失敗しました : %s。)

Store %s: Failure restoring transaction ack: %s. (ストア %s : トランザクション確  
認の復元に失敗しました : %s。)

Store %s: Failure resetting topic: %s. (ストア %s : トピックのリセットに失敗し  
ました : %s。)

Store %s: Correct functioning cannot be guaranteed due to mstore failure. (スト  
ア %s : mstore が失敗したため、正しい機能を保証できません。) Exiting. (終了  
します。)

Failed writing to mstore: I/O error or out of disk space. (mstore への書き込みに  
失敗しました : I/O エラーまたはディスクスペースの不足です。)

---

カテゴリ	マルチキャストチャネルに割り振られた帯域幅を超過しました。
説明	マルチキャストチャネルに割り振られた帯域幅を超過したことを示しています。
解決方法	チャネルの maxrate プロパティを増加するか、サーバーの multicast_reserved_rate プロパティを増加することで、マルチキャストチャネル に割り振られた帯域幅を増やすか、フロー制御を有効にするか、パブリッシャ をスローダウンします。
エラー	Multicast channel ¥q%s¥q has exceeded its allotted bandwidth (マルチキャスト チャネル ¥q%s¥q は割り振られた帯域幅を超過しました)

---

カテゴリ	マルチキャストデーモンのステータスコードとエラー
説明	マルチキャストデーモンでエラーが発生しました。
解決方法	マルチキャストデーモンとサーバーの設定に加え、ネットワークの動作状態も確認します。
エラー	<p>Interface IP address: %s ( インタフェース IP アドレス : %s)</p> <p>[%s] Connection created, connid=% PRINTF_LLFMt d ([%s] 接続が作成されました。接続 ID = % PRINTF_LLFMt d)</p> <p>Error: Unable to set channel property %q%s%q=% PRINTF_LLFMt d ( エラー : チャンネルプロパティ %q%s%q=% PRINTF_LLFMt d を設定できません )</p> <p>[%s] Created consumer consid=%PRINTF_LLFMtd connid=%PRINTF_LLFMtd topic=%q%s%q ([%s] 作成されたコンシューマの接続 ID = %PRINTF_LLFMtd 接続 ID = %PRINTF_LLFMtd トピック = %q%s%q)</p> <p>Multicast Daemon Id=%s ( マルチキャストデーモン ID = %s)</p> <p>Statistics enabled on a %d second interval. ( 統計情報は %d の 2 番目のインターバルで有効になりました。 )</p> <p>Statistics disabled. ( 統計情報は無効になりました。 )</p> <p>Rotating log from %s to %s (%s から %s にログをローテーションしています)</p> <p>Memory allocation error, possible data loss. ( メモリの割り当てエラー。データが失われた可能性があります。 )</p> <p>Unrecoverable PGM error rc=%d, reason=%s ( 回復不能 PGM エラー rc = %d、理由 = %s)</p> <p>Could not parse configuration file %q%s%q ( 設定ファイル %q%s%q を解析できませんでした )</p> <p>Interface IP address: %s ( インタフェース IP アドレス : %s)</p> <p>Tracing enabled. ( トレースが有効になりました。 )</p> <p>Tracing disabled. ( トレースが無効になりました。 )</p> <p>refused new connection with existing ID % PRINTF_LLFMt d ( 既存の ID % PRINTF_LLFMt d との新規通信が拒否されました )</p> <p>[%s] Connection destroyed, connid=%PRINTF_LLFMtd ([%s] 接続が破棄されました。通信 ID = %PRINTF_LLFMtd)</p> <p>Error sending to consid=%PRINTF_LLFMtd connid=%PRINTF_LLFMtd from channel %q%s%q: %s ( チャンネル %q%s%q から接続 ID = %PRINTF_LLFMtd 接続 ID = %PRINTF_LLFMtd に送信しているときにエラーが発生しました : %s)</p>

%s, status=%s (%s、ステータス = %s)

Attached channel %q%s¥q to consumer consid=%%PRINTF\_LLFMtd  
connid=%%PRINTF\_LLFMtd ( チャンネル %q%s¥q をコンシューマ接続 ID =  
%%PRINTF\_LLFMtd 接続 ID = %%PRINTF\_LLFMtd に接続しました )

Error attaching channel %q%s¥q to consumer consid=%%PRINTF\_LLFMtd  
connid=%%PRINTF\_LLFMtd ( チャンネル %q%s¥q をコンシューマ接続 ID =  
%%PRINTF\_LLFMtd 接続 ID = %%PRINTF\_LLFMtd に接続しているときに、エ  
ラーが発生しました )

Detaching channel %q%s¥q from consumer consid=%%PRINTF\_LLFMtd  
connid=%%PRINTF\_LLFMtd ( チャンネル %q%s¥q をコンシューマ接続 ID =  
%%PRINTF\_LLFMtd 接続 ID = %%PRINTF\_LLFMtd から切断しました )

Destroying consumer consid=%%PRINTF\_LLFMtd connid=%%PRINTF\_LLFMtd  
( コンシューマ接続 ID = %%PRINTF\_LLFMtd 接続 ID = %%PRINTF\_LLFMtd を破  
棄しています )

Channel configuration from server does not match existing channel %q%s¥q ( サー  
バーからのチャンネル設定が既存のチャンネル %q%s¥q と一致しません )

Ignoring additional PGM receiver created on group %q%s¥q, dport=%d,  
sport=%d, channel=%s ( グループ %q%s¥q、D ポート = %d、S ポート = %d、  
チャンネル = %s で作成された追加 PGM レシーバを無視します )

Created channel: %q%s¥q ( 作成されたチャンネル : %q%s¥q )

Error: %s is not a valid multicast-capable IP address. ( エラー : %s は有効なマル  
チキャスト可能 IP アドレスではありません。 ) Use the -ifc command line  
parameter to specify a valid interface. (-ifc コマンド行パラメータを使用して、有  
効なインタフェースを指定してください。 )

---

カテゴリ	メモリ不足
説明	サーバーは、処理を実行しようとしたときに、メモリの割り当てに失敗しました。
解決方法	オペレーティングシステムに応じて、サーバープロセスが使用しているメモリ 量を確認してください。この値と、ホストが実際に保持しているメモリおよびス ワップ領域の容量を比較します。十分なメモリとスワップ領域がある場合、サー バープロセスに対するオペレーティングシステムの制限を確認し、この制限が エラーの原因かどうかを特定してください。オペレーティングシステムの制限が 最大に設定されていてエラーが発生した場合は、トピックとキューの一部を別 のサーバーに移動することで、このサーバーの負荷を減らします。
エラー	%s trying to recreate persistent message. (%s は永続的なメッセージを再作成しよ うとしています。 )



Error during routed queue configuration, can not create routed queue consumer (ルーティング先キューの設定中にエラーが発生しました。ルーティング先キューコンシューマを作成できません)

Could not initialize monitor (モニタを初期化できませんでした)

Error: out of memory processing admin request (エラー：管理要求を処理するメモリが不足しています。)

Error during route configuration, can not create routed queue consumer, err=%s (ルートの設定中にエラーが発生しました。ルーティング先キューコンシューマを作成できません。エラー = %s)

Configuration error - duplicate group: file=%s, line=%d: ignoring line (設定エラー：重複するグループ：ファイル = %s、行 = %d：行を無視します)

Unable to create admin group: out of memory during initialization (admin グループを作成できません：初期化中にメモリが足りなくなりました)

Error: unable to create alias for %s '%s': no memory (エラー：%s '%s' のエイリアスを作成できません：メモリ不足です)

Error: unable to create alias: out of memory (エラー：エイリアスを作成できません：メモリ不足です)

Unable to create import event for %s '%s' on transport '%s' (トランスポート '%s' に %s '%s' のインポートイベントを作成できません)

Unable to create internal connection, error=(%d) %s (内部接続を作成できません。エラー = (%d) %s)

Unable to create internal connection: out of memory during initialization (内部接続を作成できません：初期化中にメモリが足りなくなりました)

Error: unable to create %s '%s': no memory (エラー：%s '%s' を作成できません：メモリ不足です)

Error: unable to create route while parsing file=%s, line=%d. (エラー：ファイル = %s、行 = %d の解析中にルートを作成できません。)

Unable to create SmartSockets subscriber on transport '%s', %s '%s': out of memory (トランスポート '%s' に SmartSockets サブスクライバを作成できません。%s '%s'：メモリ不足です)

Unable to create temporary destination, out of memory (テンポラリデスティネーションを作成できません。メモリ不足です)

Failed to create reserve memory. (予約メモリの作成に失敗しました。) Exiting. (終了します。)

Failed writing message to '%s': No memory for operation. (メッセージの '%s' への書き込みに失敗しました：処理用のメモリが不足しています。)

Unable to process message imported on transport '%s'. (トランスポート '%s' にインポートされたメッセージを処理できません。)

Fault Tolerant configuration, no memory! (フォールトレランス設定です。メモリ不足です。)

Fault Tolerant error, no memory. (フォールトレランスエラーです。メモリ不足です。)

LDAP initialization failed. (LDAP の初期化に失敗しました。)

No memory. (メモリ不足です。)

No memory authenticating user '%s' (メモリ不足のため、ユーザー '%s' を認証できません)

No memory authenticating via LDAP (メモリ不足のため、LDAP 経由で認証できません)

Out of memory while building admin response message (管理応答メッセージの作成中にメモリが足りなくなりました)

Out of memory while building JNDI response message (JNDI 応答メッセージの作成中にメモリが足りなくなりました)

Out of memory creating global import event on transport '%s' (トランスポート '%s' に対するグローバルインポートイベントの作成中にメモリが足りなくなりました)

Out of memory creating import event for %s '%s' on transport '%s' (トランスポート '%s' に対する %s '%s' のインポートイベントの作成中にメモリが足りなくなりました)

Out of memory creating SS transport %s (SS トランスポート %s の作成中にメモリが足りなくなりました)

No memory creating stalled flows in destination (メモリ不足のため、デスティネーションにストールフローが作成できません)

Out of memory during initialization (初期化中にメモリが足りなくなりました)

Could not set replyto destination exporting SS message. (SS メッセージをエクスポートする replyto デスティネーションを設定できませんでした。)

Could not set destination exporting SS message. (SS メッセージをエクスポートするデスティネーションを設定できませんでした。)

Could not get destination exporting SS message. (SS メッセージをエクスポートするデスティネーションを取得できませんでした。)

Failed to initialize SS message fields exporting SS message. (SS メッセージをエクスポートする SS メッセージ フィールドを初期化できませんでした。)

Out of memory exporting SS message. (SS メッセージのエクスポート中にメモリが足りなくなりました)

Out of memory: unable to process SS message type on export (メモリ不足: エクスポート時に SS メッセージタイプを処理できません)

No memory for creating connection. (メモリ不足のため、接続を作成できません。)

No memory generating dynamic route durable. (メモリ不足のため、動的ルート持続を生成できません。)

Out of memory importing SS message. error=%s. (メモリ不足のため、SS メッセージをインポートできません。エラー = %s。)

No memory in IO thread to create pool. (IO スレッドのメモリ不足のため、プールを作成できません。)

Out of memory while parsing bridges file (ブリッジファイルの解析中にメモリが足りなくなりました)

Out of memory while parsing factories file (ファクトリファイルの解析中にメモリが足りなくなりました)

Out of memory while parsing routes file (ルートファイルの解析中にメモリが足りなくなりました)

No memory performing routing operation. (メモリ不足のため、ルーティング動作を実行できません)

Out of memory processing %s on %s '%s' (%s '%s' 上の %s の処理中にメモリが足りなくなりました)

Out of memory processing administrative request (管理要求の処理中にメモリが足りなくなりました)

Out of memory processing message tracing (メッセージトレース中にメモリが足りなくなりました)

No memory processing purge record. (メモリ不足のため、ページレコードを処理できません。)

No memory while processing route interest (対象ルートの処理中にメモリが足りなくなりました)

Out of memory processing transports (トランスポートの処理中にメモリが足りなくなりました)

Out of memory processing transports configuration (トランスポート設定の処理中にメモリが足りなくなりました)

Out of memory reading configuration. (設定の読み取り中にメモリが足りなくなりました。)

Out of memory restoring routed consumer (ルーティング先コンシューマの復元中にメモリが足りなくなりました)

Out of memory sending monitor message. (監視メッセージの送信中にメモリが足りなくなりました。)

No memory sending topic routing information. (メモリ不足のため、トピックルーティング情報を送信できません。)

%s trying to add message to %s queue. (%s はメッセージを %s キューに追加しようとしています。)

No memory trying to add message to system. (メモリ不足のため、システムに対するメッセージの追加を試行できません。)

No memory trying to cleanup route. (メモリ不足のため、ルートのカリーンアップを試行できません。)

No memory to create ack record. (メモリ不足のため、確認応答レコードを作成できません。)

No memory to create client connection (メモリ不足のため、クライアント接続を作成できません)

No memory to create configured durable '%s%s%s'. (メモリ不足のため、設定された持続 '%s%s%s' を作成できません。)

No memory to create configured durables (メモリ不足のため、設定された持続を作成できません)

No memory to create confirm record. (メモリ不足のため、確認レコードを作成できません。)

No memory to create connection. (メモリ不足のため、接続を作成できません。)

No memory to create consumer. (メモリ不足のため、コンシューマを作成できません。)

No memory trying to create destination. (メモリ不足のため、デスティネーションの作成を試行できません。)

No memory to create destination for consumer or browser. (メモリ不足のため、コンシューマまたはブラウザのデスティネーションを作成できません。)

No memory trying to create global topic destination. (メモリ不足のため、グローバルトピックのデスティネーションの作成を試行できません。)

No memory to create message from store. (メモリ不足のため、ストアからメッセージを作成できません。)

No memory trying to create message producer. (メモリ不足のため、メッセージプロデューサの作成を試行できません。)

No memory to create producer. (メモリ不足のため、プロデューサを作成できません。)

No memory trying to create queue browser. (メモリ不足のため、キューブラウザの作成を試行できません。)

No memory trying to create response message. (メモリ不足のため、応答メッセージの作成を試行できません。)

No memory to create routed consumer (メモリ不足のため、ルーティング先コンシューマを作成できません。)

No memory to create routed queue consumers (メモリ不足のため、ルーティング先キューコンシューマを作成できません)

No memory trying to create routed queue destination. (メモリ不足のため、ルーティング先キューのデスティネーションの作成を試行できません。)

No memory trying to create routed tmp queue destination. (メモリ不足のため、ルーティング先テンポラリキューのデスティネーションの作成を試行できません。)

No memory to create session. (メモリ不足のため、セッションを作成できません。)

No memory trying to create tmp destination for consumer. (メモリ不足のため、コンシューマのテンポラリデスティネーションの作成を試行できません。)

No memory trying to create transaction. (メモリ不足のため、トランザクションの作成を試行できません。)

No memory to create valid messages record. (メモリ不足のため、有効なメッセージレコードを作成できません。)

No memory restoring valid sequence number info. (メモリ不足のため、有効なシーケンス番号情報を復元できません。)

No memory to create zone. (メモリ不足のため、ゾーンを作成できません。)

No memory trying to export message to RV. (メモリ不足のため、RV へのメッセージのエクスポートを試行できません。)

No memory trying to export message to SS. (メモリ不足のため、SS へのメッセージのエクスポートを試行できません。)

No memory trying to import message from RV%s. (メモリ不足のため、RV%s からのメッセージのインポートを試行できません。)

No memory trying to import message from RVCm. (メモリ不足のため、RVCm からのメッセージのインポートを試行できません。)

No memory trying to import message from SS. error=%s. (メモリ不足のため、SS からのメッセージのインポートを試行できません。エラー = %s。)

No memory trying to initialize connection. (メモリ不足のため、接続の初期化を試行できません。)

No memory trying to initialize route connection. (メモリ不足のため、ルート接続の初期化を試行できません。)

No memory trying to parse configured durable. (メモリ不足のため、設定された持続の解析を試行できません。)

No memory trying to process data message. (メモリ不足のため、データメッセージの処理を試行できません。)

No memory trying to process queue message. (メモリ不足のため、キューメッセージの処理を試行できません。)

No memory to process route interest (メモリ不足のため、対象ルート进行处理できません)

No memory trying to process system request. (メモリ不足のため、システム要求の処理を試行できません。)

No memory trying to process topic consumer. (メモリ不足のため、トピックコンシューマの処理を試行できません。)

No memory trying to process topic message. (メモリ不足のため、トピックメッセージの処理を試行できません。)

No memory trying to process xa end. connID=% PRINTF\_LL\_FMT d sessID=% PRINTF\_LL\_FMT d %s (メモリ不足のため、xa 終了の処理を試行できません。 connID=% PRINTF\_LL\_FMT d sessID=% PRINTF\_LL\_FMT d %s)

No memory trying to read message from store. (メモリ不足のため、ストアからのメッセージの読み取りを試行できません。)

Route down while trying to recover routed consumer. (ルーティング先コンシューマの回復を試行しているときに、ルーティングが停止します。)

No memory trying to recover routed consumer. (メモリ不足のため、ルーティング先コンシューマの回復を試行できません。)

No memory trying to recover one msg for routed consumer. (メモリ不足のため、ルーティング先コンシューマのひとつのメッセージの回復を試行できません。)

No memory trying to recover route stall. (メモリ不足のため、ルートストールの回復を試行できません。)

No memory trying to recover route stall, will try again. (メモリ不足のため、ルートストールの回復を試行できません。再試行します。)

No memory to restore messages. (メモリ不足のため、メッセージを復元できません。)

No memory to restore prepared transactions. (メモリ不足のため、準備済みのトランザクションを復元できません。)

No memory trying to retrieve for queue browser. (メモリ不足のため、キューブラウザの取得を試行できません。)

No memory trying to send recover/rollback response. (メモリ不足のため、回復 / ロールバック応答の送信を試行できません。)

out of memory trying to send topic interest to routes (メモリ不足のため、対象トピックのルートへの送信を試行できません)

No memory to set clientID for connection. (メモリ不足のため、接続に clientID を設定できません。)

No memory trying to setup queue route configuration (メモリ不足のため、キュールート設定のセットアップを試行できません)

No memory trying to setup route configuration (メモリ不足のため、ルート設定のセットアップを試行できません)

No memory trying to setup topic route configuration (メモリ不足のため、トピックルート設定のセットアップを試行できません)

Route recovery of destination %s on route from %s will fail: No memory (%s からのルートにおけるデスティネーション %s のルート回復は失敗します：メモリ不足です)

Route recovery of destination %s on route from %s will fail: No memory to create timer (%s からのルートにおけるデスティネーション %s のルート回復は失敗します：メモリ不足のため、タイマーを作成できません)

Route recovery of destination %s on route from %s will fail: %s (%s からのルートにおけるデスティネーション %s のルート回復は失敗します：%s)

Failed to initialize OpenSSL environment: out of memory (OpenSSL 環境の初期化に失敗しました：メモリ不足です)

Out of memory queuing imported message for processing. (メモリ不足のため、処理用にインポートされたメッセージをキューに入れられません。)

Out of memory gathering consumers for incoming message. (受信メッセージ用のコンシューマの収集中にメモリが足りなくなりました。)

Out of memory scheduling message delete. (メモリ不足のため、メッセージ削除のスケジュールを作成できません。)

Out of memory preparing to write message. (メッセージ書き込みの準備中にメモリが足りなくなりました。)

Out of memory assembling list of message to store. (保存するメッセージのリストの構築中にメモリが足りなくなりました)

Out of memory processing route consumer. ( ルートコンシューマの処理中にメモリが足りなくなりました )

Out of memory preparing message for writing. ( 書き込み用のメッセージの準備中にメモリが足りなくなりました。 )

Out of memory creating connection thread list. ( 接続スレッドリストの作成中にメモリが足りなくなりました。 )

Out of memory creating RV gateway thread list. (RV ゲートウェイスレッドリストの作成中にメモリが足りなくなりました。 )

Out of memory creating SmartSockets gateway thread list. error=%s.  
(SmartSockets ゲートウェイスレッドリストの作成中にメモリが足りなくなりました。エラー = %s。 )

Out of memory delaying bridged flow control response. ( ブリッジされたフロー制御応答の遅延中にメモリが足りなくなりました。 )

Out of memory preparing to delay flow control response. ( フロー制御応答の遅延の準備中にメモリが足りなくなりました。 )

Out of memory delaying one flow control response. ( ひとつのフロー制御応答の遅延中にメモリが足りなくなりました )

Out of memory delaying set of flow control responses. ( フロー制御応答のセットの遅延中にメモリが足りなくなりました )

Out of memory trying to clear message hold. ( メモリ不足のため、保持したメッセージのクリアを試行できません。 )

Out of memory trying to delete held message. ( メモリ不足のため、保持したメッセージの削除を試行できません。 )

Unable to update the valid messages record. ( 有効なメッセージレコードを更新できません。 ) Error code: %d - %s. ( エラーコード : %d - %s。 )

No memory scheduling message delete completion, Error code: %d. ( メモリ不足のため、メッセージ削除の完了のスケジュールを作成できません : エラーコード : %d。 )

No memory to build msg properties. ( メモリ不足のため、メッセージプロパティを作成できません。 )

No memory to create prop. ( メモリ不足のため、プロパティを作成できません。 )

No memory to set prop. ( メモリー不足のため、プロパティを設定できません。 )

No memory getting the list of delivered messages. ( メモリ不足のため、配信されたメッセージのリストを取得できません。 ) The JMSXDeliveryCount property of some messages may no longer be accurate. ( 一部のメッセージの JMSXDeliveryCount プロパティは正確ではない可能性があります。 )



No memory getting the list of delivered messages from session %  
 PRINTF\_LL FMT d. The JMSXDeliveryCount property of messages that were sent  
 to this session may no longer be accurate. (メモリ不足のため、セッション %  
 PRINTF\_LL FMT d から配信されたメッセージのリストを取得できません。この  
 セッションに送信されたメッセージの JMSXDeliveryCount プロパティは正確で  
 はない可能性があります。)

No memory getting the list of delivered messages during rollback of transaction  
 with xid: %s. The JMSXDeliveryCount property of messages that were rolled-back  
 may no longer be accurate. (メモリ不足のため、xid を使用したトランザクショ  
 ンのロールバック中に配信されたメッセージのリストを取得できません : %s。  
 ロールバックされたメッセージの JMSXDeliveryCount プロパティは正確ではな  
 い可能性があります。)

Out of memory discarding message. (メモリ不足のため、メッセージを破棄でき  
 ません。)

Out of memory advancing queue pending. (メモリ不足のため、キューの保留を  
 延長できません。)

Out of memory adding message to pending list. (メモリ不足のため、保留中のリ  
 ストにメッセージを追加できません。)

Out of memory returning message to pending list. (メモリ不足のため、保留中の  
 リストにメッセージを戻せません。)

Out of memory trying to re-queue after xa rollback. (メモリ不足のため、xa ロー  
 ルバックの後に再キューを試行できません。)

Out of memory finalizing restored queue: %s. (メモリ不足のため、キューの復元  
 を完了できません : %s。)

Out of memory restoring queue flush state. (メモリ不足のため、キューのフラッ  
 シュステートを復元できません。)

Out of memory detaching message during queue purge. (メモリ不足のため、  
 キューのバージ中にメッセージを切り離せません。)

Out of memory removing message from queue. (メモリ不足のため、キューから  
 メッセージを削除できません。)

Out of memory retrieving message by correlation id. (メモリ不足のため、相関  
 ID によってメッセージを取得できません。)

Out of memory scheduling cleanup of transaction ack: %s. (メモリ不足のため、  
 トランザクション確認応答のクリーンアップをスケジューリングできません :  
 %s。)

Out of memory setting message all acked: %s. (メモリ不足のため、メッセージを  
 すべて確認応答済みに設定できません : %s。)

Out of memory cleaning up transaction: %s. (メモリ不足のため、トランザクションをクリーンアップできません: %s。)

Out of memory updating sent state on ack. (メモリ不足のため、確認応答の送信ステートを更新できません。)

Out of memory updating in-doubt state on ack. (メモリ不足のため、確認応答の未確定ステートを更新できません。)

Out of memory removing message from system. (メモリ不足のため、システムからメッセージを削除できません。)

Out of memory associating ack with data. (メモリ不足のため、データに確認応答を関連付けられません。)

Out of memory associating ack with transaction. (メモリ不足のため、トランザクションに確認応答を関連付けられません。)

Out of memory resetting mstore discard scan: %s. (メモリ不足のため、mstore 破棄スキャンをリセットできません: %s。)

Out of memory recording modified topic. (メモリ不足のため、変更済みトピックをレコードできません。)

Out of memory re-queuing sent messages. (メモリ不足のため、送信されたメッセージを再キューできません。)

---

カテゴリ	プロトコルエラー - XA 要求内に不正な XID が存在する
説明	tibemspd は、セッションが現在使用しているトランザクションとは別のトランザクションを参照したサードパーティ製のトランザクションマネージャから、XA 終了指示を受け取りました。
解決方法	この問題を、トランザクションマネージャのベンダーに報告してください。
エラー	Incorrect xid in xa end (0x%x) request. connID=% PRINTF_LL_FMT d sessID=% PRINTF_LL_FMT d %s (xa 終了 (0x%x) 要求内に不正な xid があります。 connID=% PRINTF_LL_FMT d sessID=% PRINTF_LL_FMT d %s)

---

カテゴリ	プロトコルエラー - 不正な状態のトランザクション
説明	クライアントアプリケーションは、XA トランザクションを開始しようとしたが失敗しました。そのトランザクションはすでに存在し、正しい状態ではありません。

解決方法	たいていの場合、このエラーの原因は、外部トランザクションマネージャがふたつの別のクライアントアプリケーションに同じ XA トランザクション ID (XID) の使用を許可したことです。トランザクションマネージャのマニュアルを参照するか、このエラーをトランザクションマネージャのベンダーに報告してください。
エラー	<p>Cannot process xa start for a session when another transaction is already active on that session. connID=% PRINTF_LLFFMT d sessID=% PRINTF_LLFFMT d %s (セッションで別のトランザクションがすでにアクティブになっているときは、そのセッションの xa 開始を処理できません。connID=% PRINTF_LLFFMT d sessID=% PRINTF_LLFFMT d %s)</p> <p>Cannot process xa start with TMNOFLAGS when the transaction is already active. connID=% PRINTF_LLFFMT d sessID=% PRINTF_LLFFMT d %s (トランザクションがすでにアクティブになっているときは、TMNOFLAGS が付いている XA 開始を処理できません。connID=% PRINTF_LLFFMT d sessID=% PRINTF_LLFFMT d %s)</p>
カテゴリ	プロトコルメッセージ形式エラー
説明	tibemspd が受信したメッセージのデータが不足しているか、または不完全です。
解決方法	エラーおよびエラーが発生した状況の詳細を TIBCO サポートに送信します。
エラー	<p>Unable to confirm session, invalid request. (セッションを確認できません。無効な要求です。)</p> <p>Unable to create consumer, invalid destination. (コンシューマを作成できません。無効なデスティネーションです。)</p> <p>Unable to init session, invalid request. (セッションを初期化できません。無効な要求です。)</p> <p>Unable to process msg for export. error=%s. (エクスポート用のメッセージを処理できません。エラー = %s。)</p> <p>Unable to recover consumer, invalid request. (コンシューマを回復できません。無効な要求です。)</p> <p>Unable to recover consumer, invalid session. (コンシューマを回復できません。無効なセッションです。)</p> <p>Unable to recover one msg for consumer, invalid request. (コンシューマのひとつのメッセージを回復できません。無効な要求です。)</p> <p>Unable to recover one msg for consumer, invalid sequence number. (コンシューマのひとつのメッセージを回復できません。無効なシーケンス番号です。)</p>

Unable to recover one msg for consumer, invalid session. ( コンシューマのひとつのメッセージを回復できません。無効なセッションです。 )

Unable to serve the flow stall recover request from server %s, invalid request. ( サーバー %s からのフローストール回復要求を提供できません。無効な要求です。 )

Unable to start consumer, invalid consumer ( コンシューマを開始できません。無効なコンシューマです )

Unable to server the flow stall recover request from server %s, invalid consumer. ( サーバー %s からのフローストール回復要求を提供できません。無効なコンシューマです。 )

Unable to unsubscribe consumer, invalid client request. ( コンシューマのサブスクライブを解除できません。無効なクライアント要求です。 )

%s: %s failed: illegal to use %s [%s] in standby mode. (%s : %s が失敗しました : スタンバイモードで %s [%s] を使用することはできません。 )

Invalid flag in xa end request. connID=% PRINTF\_LLFMT d sessID=% PRINTF\_LLFMT d %s (xa 終了要求内に無効なフラグがあります。 connID=% PRINTF\_LLFMT d sessID=% PRINTF\_LLFMT d %s)

Invalid flag in xa start request. connID=% PRINTF\_LLFMT d sessID=% PRINTF\_LLFMT d %s (xa 開始要求内に無効なフラグがあります。 connID=% PRINTF\_LLFMT d sessID=% PRINTF\_LLFMT d %s)

Invalid request to delete temporary destination: %s. connID=% PRINTF\_LLFMT d ( テンポラリデスティネーションの削除要求は無効です : %s。接続 ID =% PRINTF\_LLFMT d)

Invalid request to delete temporary destination: not owner connection. ( テンポラリデスティネーションの削除要求は無効です : 接続の所有者ではありません。 )

Invalid xid in commit request. ( 確認要求内に無効な xid があります。 )

Invalid xid in commit transaction record. ( コミットトランザクションレコード内に無効な xid があります。 )

Invalid xid trying to update(%d) transaction record. ( 無効な xid が (%d) トランザクションレコードを更新しようとしています。 )

Invalid xid in rollback request. ( ロールバック要求内に無効な xid があります。 )

Invalid xid in rollback transaction record. ( ロールバックトランザクションレコード内に無効な xid があります。 )

Invalid xid in xa commit request. connID=% PRINTF\_LLFMT d (xa コミット要求内に無効な xid があります。接続 ID =% PRINTF\_LLFMT d)

Invalid xid in xa end request. connID=% PRINTF\_LLFFMT d sessID=% PRINTF\_LLFFMT d (xa 終了要求内に無効な xid があります。接続 ID = % PRINTF\_LLFFMT d セッション ID = % PRINTF\_LLFFMT d)

Invalid xid in xa prepare request. connID=% PRINTF\_LLFFMT d (xa 準備要求内に無効な xid があります。接続 ID = % PRINTF\_LLFFMT d)

Invalid xid in xa rollback request. connID=% PRINTF\_LLFFMT d (xa ロールバック要求内に無効な xid があります。接続 ID = % PRINTF\_LLFFMT d)

Invalid xid in xa start request. connID=% PRINTF\_LLFFMT d sessID=% PRINTF\_LLFFMT d (xa 開始要求内に無効な xid があります。接続 ID = % PRINTF\_LLFFMT d セッション ID = % PRINTF\_LLFFMT d)

Malformed routed message (ルーティング先メッセージの形式が正しくありません)

Problem decoding sequence data in confirm: %s. (確認内のシーケンスデータのデコード中に問題が発生しました: %s。)

Problem decoding sequence data in rollback. (ロールバック内のシーケンスデータのデコード中に問題が発生しました。)

Problem decoding sequence data in xa end. connID=% PRINTF\_LLFFMT d sessID=% PRINTF\_LLFFMT d %s (xa 終了内のシーケンスデータのデコード中に問題が発生しました。connID=% PRINTF\_LLFFMT d sessID=% PRINTF\_LLFFMT d %s)

%s:%s queue browser failed: queue name is missing in request message (%s : %s キューブラウザが失敗しました: 要求メッセージにキュー名が指定されていません)

Received admin request with replyTo not set (受信した管理要求に replyTo が設定されていません)

Received JNDI request with replyTo not set. (受信した JNDI 要求に replyTo が設定されていません)

Received unexpected message type %d (予期しないメッセージタイプ %d を受信しました)

No destination in incoming data message. (受信データメッセージにデスティネーションがありません。)

Invalid %s message (無効な %s メッセージです)

カテゴリ      プロトコルシーケンスエラー

説明      組み込み型以外の Java クライアントが、組み込み JMS 環境の一部である tibemspd に接続しようとしています。

**解決方法** 正規ライセンス版の tibemspd に接続するようにクライアントを再設定します。

**エラー** Invalid client connect detected. ( 無効なクライアント接続が検出されました。 )  
No closure. ( 終了できません。 )

---

**カテゴリ** 回復エラー

**説明** 回復処理中にエラーが発生しました。

**解決方法** 問題を修正できず、システムを再起動する必要がある場合は、ストアファイルのバックアップを作成して、'-forceStart' コマンド行パラメータを使用してサーバーを再起動します。これにより、エラーの内容に関係なく、サーバーは起動を試みます（メモリ不足によるエラーを除く）。このモードでは、（ファイルの破損または他の理由で）問題が生じているアプリケーションメッセージおよび / または内部レコードは削除されます。これによりデータ消失が生じる可能性があるため、このコマンド行パラメータを使用する場合は細心の注意を払い、結果をよく理解してから使用するようにしてください。ストアファイルのコピーを TIBCO サポートに送信して、事後分析を行うことができます。

**エラー** The recovery process stopped while processing a '%s' record (id=% PRINTF\_LLFORMAT d), error: %d - %s. Check the section 'Error Recovery Policy' from chapter 'Running the EMS Server' in the User's Guide before attempting to restart the server ('%s' レコード (id=% PRINTF\_LLFORMAT d) の処理中に、回復プロセスが停止しました。エラー : %d - %s。サーバーの再起動を試行する前に、ユーザーガイドの章「EMS サーバーの実行」のセクション「エラーの回復ポリシー」を確認してください)

The recovery process stopped while processing a '%s' record (id=% PRINTF\_LLFORMAT d) due to an out-of-memory condition. ( メモリ不足のため、'%s' レコード (id=% PRINTF\_LLFORMAT d) の処理中に回復処理が停止しました。 )  
Ensure that the system can allocate sufficient memory to the EMS Server process before restarting it ( システムを再起動する前に、システムが十分なメモリを EMS サーバー処理に割り当てることができることを確認してください )

Unable to get the session's context handle for %s record: %d - %s (%s レコードのセッションコンテキスト処理を取得できません : %d - %s)

Unable to get the list iterator for %s record (%s レコードのリストイテレータを取得できません )

Unable to get next element from list for %s record (%s レコードのリストから次の要素を取得できません )

Unable to create %s object, no memory ( メモリ不足のため、%s オブジェクトを作成できません )

Error occurred while processing %s record id=% PRINTF\_LLFORMAT d (%s) - Unable to reconstruct message: %d - %s (%s レコード ID = % PRINTF\_LLFORMAT d (%s) の処理中に、エラーが発生しました - メッセージを再作成できません : %d - %s)

Unable to recreate zone '%s': %d - %s (ゾーン '%s' を再作成できません : %d - %s)

Unable to add zone '%s' to the system: %d - %s (ゾーン '%s' をシステムに追加できません : %d - %s)

Zone '%s' is defined as type '%s' in configuration but also is defined as type '%s' in meta.db (ゾーン '%s' は設定でタイプ '%s' として定義されていますが、meta.db ではタイプ '%s' としても定義されています)

Unable to recreate connection id=% PRINTF\_LLFORMAT d: %d - %s (接続 ID = % PRINTF\_LLFORMAT d を再作成できません : %d - %s)

Discarding session id=% PRINTF\_LLFORMAT d because the connection id=% PRINTF\_LLFORMAT d was not recovered. (接続 ID = % PRINTF\_LLFORMAT d が回復されなかったため、セッション ID = % PRINTF\_LLFORMAT d を破棄しています。) Recovery continues (回復が続行されます)

Unable to recreate session id=% PRINTF\_LLFORMAT d with connection id=% PRINTF\_LLFORMAT d: %d - %s (接続 ID = % PRINTF\_LLFORMAT d を持つセッション ID = % PRINTF\_LLFORMAT d を再作成できません : %d - %s)

Unable to recreate consumer id=% PRINTF\_LLFORMAT d with connection id=% PRINTF\_LLFORMAT d and session id=% PRINTF\_LLFORMAT d: invalid destination: %s (接続 ID = % PRINTF\_LLFORMAT d およびセッション ID = % PRINTF\_LLFORMAT d を持つコンシューマ ID = % PRINTF\_LLFORMAT d を再作成できません : 無効なデスティネーションです : %s)

No memory to create destination for consumer id=% PRINTF\_LLFORMAT d (メモリ不足のため、コンシューマ ID = % PRINTF\_LLFORMAT d のデスティネーションを作成できません。)

Discarding consumer id=% PRINTF\_LLFORMAT d on destination '%s' because connection id=% PRINTF\_LLFORMAT d was not restored. (接続 ID = % PRINTF\_LLFORMAT d が回復されなかったため、デスティネーション '%s' でコンシューマ ID = % PRINTF\_LLFORMAT d を破棄しています。) Recovery continues (回復が続行されます)

Discarding consumer id=% PRINTF\_LLFORMAT d on destination '%s' and connection id=% PRINTF\_LLFORMAT d because session id=% PRINTF\_LLFORMAT d was not restored. (セッション ID = % PRINTF\_LLFORMAT d が回復されなかったため、デスティネーション '%s' のコンシューマ ID = % PRINTF\_LLFORMAT d および接続 ID = % PRINTF\_LLFORMAT d を破棄しています。) Recovery continues (回復が続行されます)

No memory to recreate consumer id=% PRINTF\_LLFORMAT d (メモリ不足のため、コンシューマ ID = % PRINTF\_LLFORMAT d を再作成できません)



Failed to build import selectors for consumer id=% PRINTF\_LLFMT d: %d - %s  
( コンシューマ ID = % PRINTF\_LLFMT d のインポートセクタを作成できませんでした : %d - %s)

Failed to read import selectors for routed consumer id=% PRINTF\_LLFMT d: %d - %s  
( コンシューマ ID = % PRINTF\_LLFMT d のインポートセクタを読み取れませんでした : %d - %s)

Discarding durable id=% PRINTF\_LLFMT d (connection id=% PRINTF\_LLFMT d) on destination '%s' because the durable name is not specified. ( 持続名が指定されていないため、デスティネーション '%s' の持続 ID = % PRINTF\_LLFMT d ( 接続 ID = % PRINTF\_LLFMT d ) を破棄しています。 ) Recovery continues ( 回復が続行されます )

Unable to recreate producer id=% PRINTF\_LLFMT d with connection id=% PRINTF\_LLFMT d and session id=% PRINTF\_LLFMT d: invalid destination: %s  
( 接続 ID = % PRINTF\_LLFMT d およびセッション ID = % PRINTF\_LLFMT d を持つプロデューサ ID = % PRINTF\_LLFMT d を再作成できません : 無効なデスティネーションです : %s)

No memory to create destination for producer id=% PRINTF\_LLFMT d ( メモリ不足のため、プロデューサ ID = % PRINTF\_LLFMT d のデスティネーションを作成できません。 )

Discarding producer id=% PRINTF\_LLFMT d on destination '%s' because connection id=% PRINTF\_LLFMT d was not restored. ( 接続 ID = % PRINTF\_LLFMT d が回復されなかったため、デスティネーション '%s' でプロデューサ ID = % PRINTF\_LLFMT d を破棄しています。 ) Recovery continues ( 回復が続行されます )

Discarding producer id=% PRINTF\_LLFMT d on destination '%s' with connection id=% PRINTF\_LLFMT d because session id=% PRINTF\_LLFMT d was not restored. ( セッション ID = % PRINTF\_LLFMT d が回復されなかったため、接続 ID = % PRINTF\_LLFMT d を持つ、デスティネーション '%s' のプロデューサ ID = % PRINTF\_LLFMT d を破棄しています。 ) Recovery continues ( 回復が続行されます )

Unable to recreate purge record: invalid destination: %s ( パージレコードを再作成できません : 無効なデスティネーション : %s)

Unable to recreate purge record for destination %s: %d - %s ( デスティネーション %s のパージレコードを再作成できません : %d - %s)

Error creating message for transaction record: %d - %s ( トランザクションレコードのメッセージの作成中にエラーが発生しました : %d - %s)

Error creating message's store structure for transaction record: %d - %s ( トランザクションレコードのメッセージストア構造の作成中にエラーが発生しました : %d - %s)



Unable to recover transaction record: transaction id missing: %d - %s ( トランザクションレコードを回復できません : トランザクション ID が欠落しています : %d - %s)

Unable to recover transaction id=% PRINTF\_LLFMT d: %d - %s ( トランザクション ID = % PRINTF\_LLFMT d を回復できません : %d - %s)

Unable to recover ack record (txid=% PRINTF\_LLFMT d, consid=% PRINTF\_LLFMT d, seqid=% PRINTF\_LLFMT d, location=%s): %d - %s ( 確認応答レコードを回復できません (tx ID = % PRINTF\_LLFMT d、接続 ID = % PRINTF\_LLFMT d、seq ID = PRINTF\_LLFMT d、ロケーション = %s) : %d - %s)

Unable to recover ack record, cannot create message: %d - %s ( 確認応答レコードを回復できません。メッセージを作成できません : %d - %s)

Unable to recover sequence numbers from valid record: %s ( 有効なレコードからシーケンス番号を回復できません : %s)

Unable to recover message, can not create lock: %d - %s ( メッセージを回復できません。ロックを作成できません : %d - %s)

Unable to restore held message from store, (location=%s) no memory ( メモリ不足のため、ストアから保持されているメッセージを回復できません (ロケーション = %s) )

Unable to restore message sequence=% PRINTF\_LLFMT d: (location=%s) %d - %s ( メッセージシーケンス = % PRINTF\_LLFMT d を回復できません (ロケーション = %s) %d - %s)

No memory to create destination for message ( メモリ不足のため、メッセージのデスティネーションを作成できません )

Inconsistency restoring routed message sequence=% PRINTF\_LLFMT d ( ルーティングされたメッセージシーケンス = % PRINTF\_LLFMT d の回復で不整合が生じています )

No memory to restore routed message sequence=% PRINTF\_LLFMT d ( メモリ不足のため、ルーティングされたメッセージシーケンス = % PRINTF\_LLFMT d を回復できません )

Persisted message possibly corrupted: %s ( 永続的メッセージが破損している可能性があります : %s)

Error creating message's store structure: %d - %s ( メッセージストア構造の作成中にエラーが発生しました : %d - %s)

---

**カテゴリ** SSL 経由で TCP ポートに接続しようとしたが、拒否されました

**説明** クライアントアプリケーションは、SSL プロトコルを使用してサーバーの TCP ポートに接続を試みました。

解決方法	クライアントアプリケーションの URL を SSL から TCP に変更するか、サーバーのリスンパラメータを TCP から SSL に変更します。サーバーのリスンパラメータの変更をアクティブ化するには、サーバーの再起動が必要です。
エラー	Rejected attempt to connect via SSL to TCP port (SSL 経由で TCP ポートに接続しようとしたましたが、拒否されました)
カテゴリ	TCP 経由で SSL ポートに接続しようとしたましたが、拒否されました
説明	クライアントアプリケーションは、TCP プロトコルを使用してサーバーの SSL ポートに接続を試みました。
解決方法	クライアントアプリケーションの URL を TCP から SSL に変更するか、サーバーのリスンパラメータを SSL から TCP に変更します。サーバーのリスンパラメータの変更をアクティブ化するには、サーバーの再起動が必要です。
エラー	Rejected attempt to connect via TCP to SSL port (TCP 経由で SSL ポートに接続しようとしたましたが、拒否されました)
カテゴリ	ルートからの接続拒否：ルート内に無効な閉路が存在します
説明	サーバーのマルチホップルートサポートは、閉路の設定をサポートしていません。それにもかかわらず、閉路を作成する設定が検出されました。
解決方法	閉路を作成するルートのいずれかを削除します。
エラー	<p>[%s@%s]: rejected connect from route: invalid cycle in route: %s ([%s@%s] : ルートからの接続が拒否されました：ルート内に無効な閉路が存在します：%s)</p> <p>Illegal, route to '%s' creates a cycle. ( 不正な処理です。'%s' へのルートは閉路を作成します。 ) Terminate the connection ( 接続を終了します )</p> <p>Illegal, route to '%s' creates a cycle. ( 不正な処理です。'%s' へのルートは閉路を作成します。 )</p>
カテゴリ	Rendezvous トランスポートエラー
説明	tibemsd で Rendezvous エラーが発生しました。
解決方法	Rendezvous のマニュアルを参照し、エラーの意味と修正方法の詳細を確認してください。

- エラー**    Unable to create dispatcher for import event for %s '%s' on transport '%s', error is %s (トランスポート '%s' に %s '%s' 用のインポートイベントのディスパッチャを作成できません。エラーは %s です)
- Unable to create inbox for import event for %s '%s' on transport '%s' (トランスポート '%s' に %s '%s' 用のインポートイベントの受信箱を作成できません。)
- Unable to create Rendezvous Certified transport '%s': %s (Rendezvous 保証トランスポート '%s' を作成できません : %s)
- Unable to create Rendezvous Certified transport '%s' because unable to create Rendezvous transport '%s' (Rendezvous トランスポート '%s' を作成できないため、Rendezvous 保証トランスポート '%s' を作成できません)
- Unable to create Rendezvous transport '%s': %s (Rendezvous トランスポート '%s' を作成できません : %s)
- Unable to create TIBCO Rendezvous Certified Listener for %s '%s' on transport '%s': %s (トランスポート '%s' に %s '%s' 用の TIBCO Rendezvous 証明書リスナーを作成できません : %s)
- Failed to confirm RVCN message: %d (%s) (RVCN メッセージを確認できませんでした : %d (%s) )
- Failed to confirm RVCN message sequence % PRINTF\_LLFMt u from cm sender '%s'. (RVCN メッセージシーケンス % PRINTF\_LLFMt u をセNDER '%s' から確認できませんでした。 ) Error: %d (%s) ( エラー : %d (%s) )
- Unable to store trackId % PRINTF\_LLFMt d for RVCN message sequence % PRINTF\_LLFMtu from cm sender '%s'. (RVCN メッセージシーケンス % PRINTF\_LLFMtu のトラック ID % PRINTF\_LLFMt d を cm センダー '%s' から保存できませんでした。 ) Error: %d (%s) ( エラー : %d (%s) )
- Unable to retrieve trackId % PRINTF\_LLFMt d. Error: %d (%s) (トラック ID % PRINTF\_LLFMt d を取得できません。エラー : %d (%s) )
- A problem occurred while importing RVCN message sequence % PRINTF\_LLFMt u from cm sender '%s'. (RVCN メッセージシーケンス % PRINTF\_LLFMt u を cm センダー '%s' からインポートしているときに、問題が発生しました。 ) Expecting a redelivery (再配信を要求しています)
- Unable to queue the request type: %d. Transport '%s', destination '%s', CM Sender '%s', CM Sequence % PRINTF\_LLFMt u . ( 要求タイプをキューできません : %d。 '%s' をトランスポートします。 デスティネーション '%s'、CM センダー '%s'、CM シーケンス % PRINTF\_LLFMt u。 ) Error: %d (%s) ( エラー : %d (%s) )
- Unable to queue the request type: %d. Transport '%s', destination '%s'. ( 要求タイプをキューできません : %d。 '%s' をトランスポートします。 デスティネーション '%s'。 ) Error: %d (%s) ( エラー : %d (%s) )

Failed to disallow Rendezvous Certified Message listener '%s': %s (Rendezvous 保証メッセージリスナー '%s' の許可の取り消しに失敗しました : %s)

Unable to export topic message, error=%s. ( トピックメッセージをエクスポートできません。エラー = %s。 )

Unable to pre-register certified listener '%s' on transport '%s': %s ( トランスポート '%s' に証明書リスナー '%s' を事前登録できません : %s)

Rendezvous send failed on transport '%s', error='%s' ( トランスポート '%s' で Rendezvous 送信に失敗しました。エラー = %s)

Unable to restart the CM Listener for %s '%s' (RVCM Transport '%s'). (%s '%s' (RVCM トランスポート '%s') の CM リスナーを再起動できません。 ) Error code: %d '%s' ( エラーコード : %d '%s')

Unable to create the timer for the restart of the CM Listener for %s '%s' (RVCM Transport '%s'). (%s '%s' (RVCM トランスポート '%s') の CM リスナーの再起動のためのタイマーを作成できません。 ) Error code: %d '%s' ( エラーコード : %d '%s')

Unable to stop the CM Listener for %s '%s' (RVCM Transport '%s'). (%s '%s' (RVCM トランスポート '%s') の CM リスナーを停止できません。 ) Error code: %d '%s' ( エラーコード : %d '%s')

カテゴリ コンシューマの復元の失敗

説明 tibemsd のスタートアップ時に、routes.conf に指定されているルートのゾーンの変更が検出された場合に表示されます。

解決方法 ルートを削除するか、ゾーンバックを変更してから、tibemsd を再起動します。

エラー Restoring consumer failed: Conflicting zone for route to [%s]: The route was initially zone %s type %s, but now %s type %s. Zone change not allowed while there are durable subscribers. ( コンシューマの復元に失敗しました : [%s] へのルートのゾーンが競合しています : ルートは、最初はゾーン %s、タイプ %s でしたが、現在は %s タイプ %s です。持続サブスクライバが存在する場合、ゾーンを変更することはできません。 ) Please delete the route first and create new one. ( ルートを削除してから、新しいルートを作成してください。 )

カテゴリ 予約メモリの使用

説明 tibemsd のメモリが足りなくなったため、予約メモリを使用していることを示す警告です。

**解決方法**      メッセージをコンSUMするか、キューからメッセージを削除することによって、tibemsd のメモリ要件を低くします。または、コンピュータ上の他のプロセスをシャットダウンするか、コンピュータのメモリを増設することによって、tibemsd が利用可能なメモリ容量を増やします。

**エラー**      Running on reserve memory, ignoring new message. ( 予約メモリを使用しています。新しいメッセージを無視します。 )

Running on reserve memory, no more send requests accepted. ( 予約メモリを使用しています。新しい送信要求は受け入れられません。 ) Pending msg count = % PRINTF\_LL\_FMT d ( 保留中のメッセージ数 = % PRINTF\_LL\_FMT d )

Pending msg count = % PRINTF\_LL\_FMT d ( 保留中のメッセージ数 = % PRINTF\_LL\_FMT d )

---

**カテゴリ**      フォールトトレランスセットアップでの実行時エラー

**説明**      フォールトトレランスのセットアップで、実行時にエラーが発生します。

**解決方法**      両方のサーバー（プライマリ、スタンバイ）のステータスを確認します。両方のサーバーがアクティブである場合、ファイルストアデータがすでに破損している場合があります。両方のサーバーをシャットダウンして、状況を調査することをお勧めします。

**エラー**      Fault-tolerance error: Dual-Active server detected at: '%s' ( フォールトトレランスエラー：次の場所でデュアルアクティブサーバーが検出されました：'%s' )

The primary EMS server does not hold the lock on meta store ( プライマリ EMS サーバーはメタストアにロックを保持していません )

The standby EMS server could not find the specified meta store. ( スタンバイ EMS サーバーは指定されたメタストアを検出できませんでした。 )

The primary EMS server name is %s while the standby EMS server name is %s. The names must be the same ( プライマリ EMS サーバー名は %s ですが、スタンバイ EMS サーバー名は %s です。名前は同じである必要があります )

A backup EMS server (%s) is already connected to the primary EMS server ( バックアップ EMS サーバー (%s) はすでにプライマリ EMS サーバーに接続されています )

Fault Tolerant error (%s), can't create connection to '%s'. ( フォールトトレランスエラー (%s)。'%s' への接続を作成できません。 )

---

**カテゴリ**      SmartSockets トランスポートエラー

説明	tibemspd で SmartSockets エラーが発生しました。
解決方法	SmartSockets のマニュアルを参照し、エラーの意味と修正方法の詳細を確認してください。
エラー	<p>Unable to create SmartSockets subscriber on transport '%s': failed to convert %s '%s', error=%s ( トランスポート '%s' に SmartSockets サブスクライバを作成できません。 %s '%s' の変換に失敗しました。 エラー = %s)</p> <p>Unable to import SmartSockets message on transport %s: failed to convert subject '%s', error=%s ( トランスポート '%s' の SmartSockets メッセージをインポートできません。サブジェクト '%s' の変換に失敗しました。 エラー = %s)</p> <p>Unable to import SmartSockets message on transport %s: failed to tokenize subject '%s' ( トランスポート %s で SmartSockets メッセージをインポートできません : サブジェクト '%s' をトークン化できませんでした )</p> <p>Unable to import SmartSockets message on transport %s: failed to convert reply subject '%s', error=%s ( トランスポート '%s' の SmartSockets メッセージをインポートできません。応答サブジェクト '%s' の変換に失敗しました。 エラー = %s)</p> <p>Unable to import SmartSockets message on transport %s: no destination found '%s' ( トランスポート %s で SmartSockets メッセージをインポートできません : デスティネーションが見つかりませんでした '%s')</p> <p>Unable to export EMS message into SmartSockets on transport '%s'. ( トランスポート '%s' の SmartSockets に EMS メッセージをエクスポートできません。 )</p> <p>Failed to convert subject '%s', error=%s. ( サブジェクト '%s' の変換に失敗しました。 エラー = %s。 )</p> <p>Unable to export EMS message into SmartSockets on transport '%s'. ( トランスポート '%s' の SmartSockets に EMS メッセージをエクスポートできません。 )</p> <p>Failed to convert reply subject '%s', error=%s. ( 応答サブジェクト '%s' の変換に失敗しました。 エラー = %s。 )</p> <p>Error translating EMS message body into SS message. (EMS メッセージ本文を SS メッセージに変換中にエラーが発生しました。 ) Status=%s ( ステータス = %s)</p> <p>Error translating EMS message headers into SS message. (EMS メッセージヘッダーを SS メッセージに変換中にエラーが発生しました。 ) Status=%s ( ステータス = %s)</p> <p>Error translating EMS message properties into SS message. (EMS メッセージプロパティを SS メッセージに変換中にエラーが発生しました。 ) Status=%s ( ステータス = %s)</p> <p>Unable to confirm SS message. (SS メッセージを確認できません。 ) %s (%s)</p> <p>Unable to connect to SmartSockets RTserver via transport: '%s': %d - %s ( トランスポート経由で SmartSockets RTserver に接続できません : '%s' : %d - %s)</p>

Unable to register GMD failure callback: '%s': %d - %s (GMD 障害コールバックを登録できません: '%s': %d - %s)

Unable to create open callback on transport: '%s': %d - %s ( トランスポートのオープンコールバックを作成できません: '%s': %d - %s)

Unable to create default callback on transport: '%s': %d - %s ( トランスポートでデフォルトコールバックを作成できません: '%s': %d - %s)

Unable to create SS callback for %s '%s' on transport '%s' SS error: %s ( トランスポート '%s' SS エラーの %s '%s' の SS コールバックを作成できません: %s)

Unable to create SS message type on export ( エクスポート時に SS メッセージタイプを作成できません )

Unable to create SmartSockets subscriber for %s '%s' on transport '%s', error: %s ( トランスポート '%s' の %s '%s' の SmartSockets サブスクリバを作成できません。エラー: %s)

Unable to create SmartSockets transport '%s': %d - %s (SmartSockets トランスポート '%s' を作成できません: %d - %s)

Failed to confirm SS message. error=%s. (SS メッセージの確認に失敗しました。エラー = %s。)

Failed to create SmartSockets transport %s (SmartSockets トランスポート %s の作成に失敗しました)

Unable to handoff confirm SS message: %s. ( 確認 SS メッセージをハンドオフできません: %s。)

Unable to import SS message. (SS メッセージをインポートできません。)  
Error=%d, %s. ( エラー = %d、'%s'。)

Unable to import SS message data fields. (SS メッセージデータフィールドをインポートできません。)  
Error=%d, %s. ( エラー = %d、'%s'。)

Unable to import SS message headers. (SS メッセージヘッダをインポートできません。)  
Error=%d, %s. ( エラー = %d、'%s'。)

Unable to import SS message, failed to create message destination. (SS メッセージをインポートできません。メッセージのデスティネーションを作成できませんでした。)

Unable to import SS message, failed to create reply destination. (SS メッセージをインポートできません。応答のデスティネーションを作成できませんでした。)

Unable to import SS message, error retrieving delivery mode. (SS メッセージをインポートできません。配信モードの取得中にエラーが発生しました。)

Unable to import SS message, error setting imported property. error=%s. (SS メッセージをインポートできません。インポートされたプロパティの設定中にエラーが発生しました。エラー = %s。)



Unable to import SS message, error setting message extentions property.  
error=%s. (SS メッセージをインポートできません。メッセージ拡張子プロパ  
ティの設定中にエラーが発生しました。エラー = %s。)

Unable to import SS message, failed to create message wire. error=%s. (SS メッ  
セージをインポートできません。メッセージのワイヤーを作成できませんでした。  
エラー = %s。)

Unable to import SS message, error retrieving number of fields. (SS メッセージを  
インポートできません。フィールド数の取得中にエラーが発生しました。)

Unable to initialize SmartSockets transport '%s': error=%d: %s (SmartSockets トラ  
ンスポート '%s' を初期化できません : エラー = %d : %s)

Unable to set SmartSockets Dispatcher for transport: '%s': %d - %s ( トランスポー  
ト '%s' の SmartSockets ディスパッチャを設定できません : %d - %s)

Unable to set SS message type on export ( エクスポート時に SS メッセージタイプ  
を設定できません )

Unable to set Username/Password for SmartSockets transport '%s': %d - %s  
(SmartSockets トランスポート '%s' のユーザ名 / パスワードを設定できません :  
%d - %s)

Unable to import SmartSockets message on transport %s: failed to retrieve SS  
subject. ( トランスポート '%s' に対する SmartSockets メッセージをインポートで  
きません。SS サブジェクトの取得に失敗しました。)

SS Subject CB destroy Failed: for '%s' on transport '%s' SS error: %s (SS サブジェク  
ト CB の破棄に失敗しました : トランスポート '%s' の '%s' の SS エラー : %s)

SS Subject CB lookup Failed: for '%s' on transport '%s' SS error: %s (SS サブジェク  
ト CB の照合に失敗しました : トランスポート '%s' の '%s' の SS エラー : %s)

SmartSockets TipcMsgSetDeliveryMode failed, '%s' (SmartSockets  
TipcMsgSetDeliveryMode が失敗しました。'%s')

SmartSockets TipcMsgSetLbMode failed, '%s' (SmartSockets TipcMsgSetLbMode  
が失敗しました。'%s')

SmartSockets TipcSrvConnFlush failed, '%s' (SmartSockets TipcSrvConnFlush が  
失敗しました。'%s')

SmartSockets TipcSrvConnMsgSend failed, '%s' (SmartSockets  
TipcSrvConnMsgSend が失敗しました。'%s')

SS Unsubscribe failed: for '%s' on transport '%s' SS error: %s (SS のサブスクライ  
ブ解除に失敗しました : トランスポート '%s' の '%s' の SS エラー : %s)

GMD delivery failed on transport '%s', SS message seq=%d, reason='%s' for  
process '%s' (GMD 配信がトランスポート '%s' で失敗しました。SS メッセージ  
シーケンス = %d、理由 = プロセス '%s' の '%s')



Unable to process undelivered SS GMD message, can not register EMS message, error='%s', tport='%s', GMD seq=%d ( 未配信の SS GMD メッセージを処理できません。EMS メッセージを登録できません。エラー = '%s'、トランスポート = '%s'、GMD シーケンス = %d)

Unable to process undelivered SS GMD message, can not add to undelivered EMS queue, error='%s', tport='%s', GMD seq=%d ( 未配信の SS GMD メッセージを処理できません。未配信の EMS キューに追加できません。エラー = '%s'、トランスポート = '%s'、GMD シーケンス = %d)

Unable to process undelivered SS GMD message, failed to build EMS message, error='%s', tport='%s', GMD seq=%d ( 未配信の SS GMD メッセージを処理できません。EMS メッセージの作成に失敗しました。エラー = '%s'、トランスポート = '%s'、GMD シーケンス = %d)

Unable to convert undelivered SS GMD message into EMS message, error='%s', tport='%s', GMD seq=%d ( 未配信の SS GMD メッセージを EMS メッセージに変換できません。エラー = '%s'、トランスポート = '%s'、GMD シーケンス = %d)

---

カテゴリ	SSL の初期化の失敗
説明	サーバーは、OpenSSL ライブラリを初期化しようとして失敗しました。
解決方法	OpenSSL エラーを調べて、EMS ユーザーガイドの SSL の使用について説明している章を確認します。
エラー	<p>Failed to process ft ssl password (ft ssl パスワードの処理に失敗しました )</p> <p>Failed to process ssl password (ssl パスワードの処理に失敗しました )</p> <p>Ignoring SSL listen port %s (SSL リスポート %s を無視します )</p> <p>Failed to initialize SSL: can not load certificates and/or private key and/or CRL file(s) and/or ciphers. (SSL を初期化できませんでした : 証明書、秘密鍵、CRL ファイル、および / または暗号を読み込めません。)</p> <p>Failed to initialize OpenSSL environment: error=%d, message=%s. (OpenSSL 環境の初期化に失敗しました : エラー = %d、メッセージ = %s。)</p> <p>Failed to initialize SSL. (SSL の初期化に失敗しました。 ) Error=%s ( エラー = %s)</p> <p>Failed to initialize SSL: unable to obtain password (SSL の初期化に失敗しました : パスワードを取得できません )</p> <p>Failed to initialize SSL: server certificate not specified. (SSL の初期化に失敗しました : サーバー証明書が指定されていません。)</p> <p>Failed to initialize SSL: server private key not specified. (SSL の初期化に失敗しました : サーバーの秘密鍵が指定されていません。)</p>

カテゴリ	ストアファイル形式の不一致
説明	指定されたストアファイルは、このバージョンでサポートされていない異なるバージョンの EMS から作成されました。
解決方法	ストアファイルを作成したバージョンの EMS を使用するように戻すか、ストアファイルの変換ツールを見つけて、それを使用してストアファイルをこのバージョンに変換します。
エラー	Unsupported store format: %s (%d) ( サポートされていないストア形式 : %s (%d) )

カテゴリ	errno 駆動型のシステムコールエラー
説明	下位レベルのシステム関数に失敗しました。
解決方法	エラーをシステム管理者に報告し、問題の修正を依頼します。
エラー	<p>Accept() failed: too many open files. (Accept() が失敗しました : 開いているファイルが多過ぎます。 ) Please check per-process and system-wide limits on the number of open files. ( 開いているファイル数に関して、プロセスごとおよびシステム全体の制限を確認してください。 )</p> <p>Accept() failed: %d (%s) (Accept() に失敗しました : %d (%s) )</p> <p>Select() failed: %d (%s) (Select() に失敗しました : %d (%s) )</p> <p>Epoll_wait() failed: %d (%s) (Epoll_wait() に失敗しました : %d (%s) )</p> <p>Epoll_ctl() %s on fd %d failed: %d (%s) (fd %d での Epoll_ctl() %s に失敗しました : %d (%s) )</p> <p>ioctl() on /dev/poll failed: %d (%s) (/dev/poll での ioctl() に失敗しました : %d (%s) )</p> <p>write() %s update /dev/poll on fd %d failed: %d (%s) (fd %d で write() %s による /dev/poll の更新に失敗しました : %d (%s) )</p> <p>Cannot retrieve user name of the current process. ( 現在のプロセスのユーザー名を取得できません。 )</p> <p>Client connection not created, %s. ( クライアント接続は作成されませんでした。 %s。 )</p> <p>Could not obtain hostname ( ホスト名を取得できませんでした )</p>

Could not resolve hostname '%s'. ( ホスト名 '%s' を解決できません。 ) Possibly default hostname is not configured properly while multiple network interfaces are present. ( 複数のネットワークインタフェースが存在している場合、デフォルトのホスト名が適切に設定されてない可能性があります。 )

Unable to listen for connections: %d (%s). ( 接続をリスンできません : %d (%s)。 )

Unable to open socket for listening: %d (%s). ( リスン用にソケットを開くことができません : %d (%s) )

Closing SSL connection from %s due to timeout, exceeded handshake\_timeout of %d. ( タイムアウトにより、%s からの SSL 接続を閉じています。 %d の handshake\_timeout を超過しました。 )

---

カテゴリ	前のアクションが完了しない間のトランザクションアクション。
説明	ステート変更アクションが、別の同様のアクションが処理されているトランザクションで要求されています。
解決方法	エラーおよびエラーが発生した状況の詳細を TIBCO サポートに送信します。
エラー	<p>Cannot request second state change for transaction while the first request is in progress (%d, %d) %s. ( 最初の要求が進行中である間は (%d, %d)、トランザクションに対して 2 番目のステート変更を要求できません。 %s。 )</p> <p>Unexpected request to roll xa txn forward with previous operation (%d) incomplete: %s. ( 前の処理 (%d) が未完了の状態で、xa txn をロールして転送するための予期しない要求が行われました : %s。 )</p> <p>Unexpected request to roll xa txn back with previous operation (%d) incomplete: %s. ( 前の処理 (%d) が未完了の状態で、xa txn をロールバックするための予期しない要求が行われました : %s。 )</p> <p>Unexpected request to prepare xa txn with previous operation (%d) incomplete: %s. ( 前の処理 (%d) が未完了の状態で、xa txn を準備するための予期しない要求が行われました : %s。 )</p> <p>Unexpected request to commit xa txn with previous operation (%d) incomplete: %s. ( 前の処理 (%d) が未完了の状態で、xa txn をコミットするための予期しない要求が行われました : %s。 )</p> <p>Unexpected request to commit session with previous operation (%d) incomplete. ( 前の処理 (%d) が未完了の状態で、セッションをコミットするための予期しない要求が行われました。 )</p>

---

カテゴリ	トランザクションのタイムアウト。
------	------------------

説明	タイムアウトの前に、トランザクションが完了しませんでした。問題のトランザクションは破棄されます。
解決方法	大部分の場合、トランザクションマネージャエラーによって、適切なタイミングでこのトランザクションの進行が回避されます。トランザクションマナーの正しい処理を確認します。
エラー	Rollback due to timeout on unprepared transaction. connID=% PRINTF_LL_FMT d %s ( 準備ができていないトランザクションでタイムアウトが生じたため、ロールバックします。connID=% PRINTF_LL_FMT d %s)
カテゴリ	不要なメッセージまたは重複するメッセージ
説明	tibemspd が受信したメッセージのデータが不足しているか、または不完全です。
解決方法	エラーおよびエラーが発生した状況の詳細を TIBCO サポートに送信します。
エラー	Error processing xa start request, %s. connID=% PRINTF_LL_FMT d sessID=% PRINTF_LL_FMT d (xa 開始要求の処理中にエラーが発生しました。 %s。接続 ID = % PRINTF_LL_FMT d セッション ID = % PRINTF_LL_FMT d)  Error trying to enter standby for '%s', %s. ('%s' のためにスタンバイ状態に移行しようとして、エラーが発生しました。 %s。 )
カテゴリ	認識できないオプション
説明	サーバーのコマンドラインに、認識できないオプションが含まれています。
解決方法	-help オプションを指定してサーバーを実行し、認識できないオプションが含まれているコマンドラインと比較します。
エラー	Unrecognized option: '%s'. ( 認識できないオプション : '%s'。 )

# 索引

## 記号

.NET

アセンブリのバージョン [320](#)

プログラマのチェックリスト。 [319](#)

\$sys.redelivery.delay [66](#)

## A

acl.conf ファイル [229](#)

add member コマンド [122](#)

addprop factory コマンド [122](#)

addprop queue コマンド [122](#)

addprop route コマンド [123](#)

addprop topic コマンド [123](#)

admin

connect [124](#)

パスワード [111](#)

ユーザー [120](#)

admin ユーザー [111](#)

anonymous

ユーザーとセキュリティ [111](#)

authorization パラメータ [189](#)

AUTO\_ACKNOWLEDGE モード [38](#), [328](#)

autocommit コマンド [123](#)

## B

bridges.conf ファイル [230](#)

## C

C

プログラマのチェックリスト。 [313](#)

c#

アセンブリのバージョン [320](#)

プログラマのチェックリスト。 [319](#)

channel プロパティ [56](#)

channels.conf ファイル [231](#)

CLIENT\_ACKNOWLEDGE モード [38](#), [39](#)

client\_heartbeat\_server パラメータ [201](#)

client\_timeout\_server\_connection パラメータ [202](#)

client\_trace パラメータ [211](#)

clock\_sync\_interval パラメータ [189](#)

cm\_name パラメータ [394](#)

commit コマンド [123](#)

compact コマンド [124](#)

compliant\_queue\_ack パラメータ [189](#)

connect

admin [124](#)

connect コマンド [124](#)

console\_trace パラメータ [211](#)

consumers

詳細統計情報 [449](#)

create bridge コマンド [125](#)

create durable コマンド [125](#)

create factory コマンド [125](#)

create group コマンド [125](#)

create jndiname コマンド [125](#)

create queue コマンド [126](#)

create route コマンド [126](#)

create rvcmlistener コマンド [126](#)

create topic コマンド [127](#)

create user コマンド [127](#)

## D

daemon パラメータ 394  
 dbstore\_classpath パラメータ 296  
 dbstore\_driver\_dialect パラメータ 297  
 dbstore\_driver\_name パラメータ 297  
 default\_ttl パラメータ 394  
 delete all コマンド 127  
 delete bridge コマンド 127  
 delete connection コマンド 128  
 delete durable コマンド 128  
 delete factory コマンド 128  
 delete group コマンド 128  
 delete jndiname コマンド 128  
 delete message コマンド 128  
 delete queue コマンド 129  
 delete route コマンド 129  
 delete rvcmlistener コマンド 129  
 delete topic コマンド 129  
 delete user コマンド 129  
 destination\_backlog\_swapout パラメータ 197  
 detailed\_statistics パラメータ 213  
 disconnect コマンド 130  
 disconnect\_non\_acking\_consumers パラメータ 190  
 DTC 13  
 DUPS\_OK\_ACKNOWLEDGE モード 38, 40  
 durables.conf ファイル 234

## E

echo コマンド 130  
 EMS サーバー  
   起動 102  
   停止 105  
 EMS サーバーの起動 102  
 EMS サーバーの停止 105  
 emsntsrc 106  
 exclusive プロパティ 56  
 exit コマンド 130  
 expiration プロパティ 57  
 EXPLICIT\_CLIENT\_ACKNOWLEDGE モード 39, 39,  
   39, 39  
 EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE

モード 40, 40

explicit\_config\_only パラメータ 395  
 export  
   トピックのプロパティ 58  
 export プロパティ 58  
 export\_headers パラメータ 395  
 export\_properties パラメータ 395

## F

factories.conf ファイル 235  
 FIPS  
   fips140-2 パラメータ 218  
 FIPS 140-2 471  
 flow\_control パラメータ 190  
 flowControl プロパティ 58, 83  
 ft\_activation パラメータ 203  
 ft\_active パラメータ 203  
 ft\_heartbeat パラメータ 203  
 ft\_reconnect\_timeout パラメータ 204  
 ft\_ssl\_auth\_only パラメータ 204  
 ft\_ssl\_ciphers パラメータ 206  
 ft\_ssl\_expected\_hostname パラメータ 206  
 ft\_ssl\_identity パラメータ 204  
 ft\_ssl\_issuer パラメータ 204  
 ft\_ssl\_password パラメータ 205  
 ft\_ssl\_private\_key パラメータ 205  
 ft\_ssl\_rand\_egd パラメータ 205  
 ft\_ssl\_trusted パラメータ 205  
 ft\_ssl\_verify\_host パラメータ 206  
 ft\_ssl\_verify\_hostname パラメータ 206

## G

global プロパティ 59  
 grant admin コマンド 132  
 grant queue コマンド 130  
 grant topic コマンド 131  
 groups.conf ファイル 239

## H

handshake\_timeout パラメータ 197  
 help コマンド 132

## I

ibrvcn.conf ファイル 247  
 import  
   キューのプロパティ 59  
   トピックのプロパティ 59  
 import プロパティ 59  
 info コマンド 132

## J

jaas\_classpath パラメータ 225  
 jaas\_config\_file パラメータ 225  
 jaas\_login\_timeout パラメータ 225  
 jaas.conf ファイル 240  
 jaci\_class パラメータ 226  
 jaci\_classpath パラメータ 225  
 jaci\_timeout パラメータ 226  
 Java  
   プログラマのチェックリスト。312  
 JMS\_TIBCO\_SS\_CORRELATION\_ID プロパティ 428  
 JMS\_TIBCO\_SS\_DELIVERY\_MODE プロパティ 428  
 JMS\_TIBCO\_SS\_EXPIRATION プロパティ 428  
 JMS\_TIBCO\_SS\_LB\_MODE プロパティ 428  
 JMS\_TIBCO\_SS\_MESSAGE\_ID プロパティ 428  
 JMS\_TIBCO\_SS\_PRIORITY プロパティ 428  
 JMS\_TIBCO\_SS\_SENDER プロパティ 427  
 JMS\_TIBCO\_SS\_SENDER\_TIMESTAMP プロパ  
   ティ 428  
 JMS\_TIBCO\_SS\_SEQ\_NUM プロパティ 428  
 JMS\_TIBCO\_SS\_TYPE\_NUM プロパティ 428  
 JMS\_TIBCO\_SS\_USER\_PROP プロパティ 428  
 jre\_library パラメータ 226  
 jre\_option パラメータ 227  
 JVM  
   パラメータ 226

## L

LDAP 269  
 ldap\_all\_groups\_filter パラメータ 223  
 ldap\_all\_users\_filter パラメータ 222  
 ldap\_cache\_enabled パラメータ 219  
 ldap\_cache\_ttl パラメータ 219  
 ldap\_conn\_type パラメータ 219  
 ldap\_credential パラメータ 219  
 ldap\_dynamic\_group\_attribute パラメータ 224  
 ldap\_dynamic\_group\_class パラメータ 224  
 ldap\_dynamic\_member\_url\_attribute パラメータ 224  
 ldap\_group\_base\_dn パラメータ 222  
 ldap\_group\_filter パラメータ 223  
 ldap\_group\_scope パラメータ 222  
 ldap\_principal パラメータ 219  
 ldap\_static\_group\_attribute パラメータ 223  
 ldap\_static\_group\_class パラメータ 223  
 ldap\_static\_member\_attribute パラメータ 224  
 ldap\_tls\_cacert\_dir パラメータ 220  
 ldap\_tls\_cacert\_file パラメータ 220  
 ldap\_tls\_cert\_file パラメータ 221  
 ldap\_tls\_cipher\_suite パラメータ 220  
 ldap\_tls\_key\_file パラメータ 221  
 ldap\_tls\_rand\_file パラメータ 220  
 ldap\_url パラメータ 218  
 ldap\_user\_attribute パラメータ 221  
 ldap\_user\_base\_dn パラメータ 221  
 ldap\_user\_class パラメータ 221  
 ldap\_user\_filter パラメータ 222  
 ldap\_user\_scope パラメータ 222  
 ledger\_file パラメータ 394  
 log\_trace パラメータ 210  
 logfile パラメータ 210  
 logfile\_max\_size パラメータ 211

## M

MapMessage 22, 341  
   定義 22, 341  
 max\_client\_msg\_size パラメータ 197  
 max\_connections パラメータ 197  
 max\_msg\_field\_print\_size パラメータ 191

max\_msg\_memory パラメータ 198  
 max\_msg\_print\_size パラメータ 191  
 max\_stat\_memory パラメータ 214  
 maxbytes プロパティ 60  
 maxmsgs プロパティ 61  
 maxRedelivery プロパティ 61  
 DTC 13  
 MS DTC 13  
 msg\_pool\_block\_size パラメータ 198  
 msg\_swapping パラメータ 199  
 mstore 30  
   間隔 32  
 multicast パラメータ 207  
 multicast\_channels パラメータ 207  
 multicast\_daemon\_default パラメータ 208  
 multicast\_statistics\_interval パラメータ 208

## N

network パラメータ 393  
 NO\_ACKNOWLEDGE モード 39, 39, 39  
 npsend\_check\_mode パラメータ 192

## O

overflowPolicy プロパティ 62

## P

PGM 5  
 policy.1.0 320  
 prefetch プロパティ 64  
 processor\_ids パラメータ 193  
 producers  
   詳細統計情報 449  
 purge all queues コマンド 133  
 purge all topics コマンド 133  
 purge durable コマンド 133  
 purge queue コマンド 133

purge topic コマンド 133

## Q

queue\_import\_dm パラメータ 395  
 queues.conf ファイル 240

## R

rate\_interval パラメータ 213  
 redeliveryDelay プロパティ 66  
 remove member コマンド 133  
 removeprop factory コマンド 134  
 removeprop queue コマンド 134  
 removeprop route コマンド 134  
 removeprop topic コマンド 134  
 request\_old パラメータ 394  
 reserve\_memory パラメータ 199  
 resume route コマンド 134  
 revoke admin コマンド 134  
 revoke queue コマンド 135  
 revoke topic コマンド 135  
 rotatelog コマンドの実行 136  
 routes.conf ファイル 241  
 rv\_tport パラメータ 394  
 RVCM パラメータ 394

## S

Schema Export Tool 302  
 secure プロパティ 67  
 secure プロパティとアクセス権限 67  
 selector\_logical\_operator\_limit パラメータ 194  
 sender\_name プロパティ 68  
 sender\_name\_enforced プロパティ 68  
 server パラメータ 194  
 server\_heartbeat\_client パラメータ 202  
 server\_heartbeat\_server パラメータ 202  
 server\_rate\_interval パラメータ 212



server\_timeout\_client\_connection パラメータ 201  
 server\_timeout\_server\_connection パラメータ 202  
 service パラメータ 393  
 set password コマンド 136  
 set server コマンド 137  
 setprop factory コマンド 142  
 setprop queue コマンド 142  
 setprop route コマンド 143  
 setprop topic コマンド 143  
 show bridge コマンド 143  
 show bridges コマンド 144  
 show config コマンド 146  
 show connections コマンド 149  
 show db コマンド 152  
 show durable コマンド 152  
 show durables コマンド 153  
 show factories コマンド 154  
 show factory コマンド 154  
 show group コマンド 155  
 show groups コマンド 155  
 show jndiname コマンド 154  
 show jndinames コマンド 155  
 show members コマンド 155  
 show message コマンド 156  
 show messages コマンド 156  
 show parents コマンド 156  
 show queue コマンド 157  
 show queues コマンド 158  
 show route コマンド 159  
 show routes コマンド 160  
 show rvcmlisteners コマンド 161  
 show rvcmltransportledger コマンド 160  
 show server コマンド 161  
 show stat コマンド 161  
 show store コマンド 162  
 show stores コマンド 164  
 show topic コマンド 164  
 show topics コマンド 166  
 show transaction コマンド 168  
 show transactions コマンド 169  
 show transport コマンド 170  
 show transports コマンド 170  
 show user コマンド 171  
 show users コマンド 171  
 showacl admin コマンド 171

showacl group コマンド 171  
 showacl queue コマンド 171  
 showacl topic コマンド 172  
 showacl user コマンド 172  
 shutdown コマンド 172  
 socket\_receive\_buffer\_size パラメータ 200  
 socket\_send\_buffer\_size パラメータ 200  
 SSL  
     サーバーのパラメータ 214  
 ssl\_auth\_only パラメータ 218  
 ssl\_cert\_user\_specname パラメータ 215  
 ssl\_crl\_path パラメータ 217  
 ssl\_crl\_update\_interval パラメータ 218  
 ssl\_dh\_size パラメータ 214  
 ssl\_password パラメータ 216  
 ssl\_rand\_egd パラメータ 217  
 ssl\_require\_client\_cert パラメータ 214  
 ssl\_server\_ciphers パラメータ 214  
 ssl\_server\_identity パラメータ 216  
 ssl\_server\_issuer パラメータ 217  
 ssl\_server\_key パラメータ 216  
 ssl\_server\_trusted パラメータ 217  
 ssl\_use\_cert\_username パラメータ 215  
 startup\_abort\_list パラメータ 195  
 statistics 448  
 statistics パラメータ 213  
 statistics\_cleanup\_interval パラメータ 213  
 store パラメータ 196  
 store プロパティ 69  
 stores.conf  
     データベースストアの設定 297  
 stores.conf ファイル 243  
 suspend route コマンド 172  
 sync\_ledger パラメータ 394

## T

tcp 91, 124, 324  
 temp\_destination\_timeout パラメータ 396  
 TIBCO\_HOME xxix  
 tibemsadmin  
     オプション 118  
 tibemsd 314, 319

tibemsd.conf

データベースストアの設定 296

tibemsd.conf ファイル 177

tibemsmcd 363

tibemsMsg\_GetStringProperty 342, 342

tibemsMsg\_SetBooleanProperty 342, 342

tibrv\_transports パラメータ 209

tibss\_config\_dir パラメータ 210

tibss\_transports パラメータ 209

time コマンド 173

topic\_import\_dm パラメータ 395

topics.conf ファイル 247

trace プロパティ 70

trace\_client\_host パラメータ 212

track\_correlation\_ids ばらめーた 207

track\_message\_id パラメータ 207

transaction commit コマンド 173

transaction rollback コマンド 173

transports.conf ファイル 248

type パラメータ 393

## U

UNIX システム

ユーザー認証に使用する 269

updatecrl コマンド 173

URL 形式 236

user\_auth パラメータ 196

users.conf ファイル 252

## W

whoami コマンド 174

## X

xa\_default\_timeout パラメータ 196

## Z

確認応答 38

間隔

mstore 32

## あ

アーキテクチャ

マルチキャスト 376

アクセス権限

secure プロパティ 67

保護 263

圧縮、メッセージ 37

暗号スイート

.NET クライアント 469

Java クライアント 466

## え

永続的メッセージ 26

キュー 26

同期ファイルストレージ 27

トピック 26

フェイルオーバー

非共有ステート 480

エラーの回復ポリシー 109

## お

オプション

tibemsadmin 118

## か

鍵

ファイル名 457

拡張可能なセキュリティ

パラメータ [224](#)

メッセージ

確認応答 [38](#)

確認応答なし受信モード [39](#)

カスタマサポート [xxxii](#), [xxxii](#)

型の変換 [44](#)

管理者

パスワードの割り当て [120](#)

## き

規則

命名 [121](#)

キュー

遅延メッセージの再配信 [66](#)

静的 [50](#)

テンポラリ [50](#)

動的 [50](#)

未配信メッセージ [20](#)

ルーティング先 [516](#)

キューの **import** プロパティ [59](#)

キューの制限ポリシー [392](#)

キューのプロパティ [55](#)

キューのプロパティのリスト [55](#)

共有ステート [483](#)

プロセス [476](#)

## く

クライアントによるトレース [211](#)

グループ [269](#)

## け

継承 [76](#)

プロパティ [76](#)

## こ

コア

割り当て [115](#)

コマンドラインオプション

マルチキャスト [364](#)

## さ

サーバー

起動 [102](#)

停止 [105](#)

削除、サブスクライバ [128](#)

サブジェクトの競合 [399](#)

サブスクライバ [274](#), [274](#)

削除 [128](#)

持続 [133](#), [133](#)

サポートへの連絡方法 [xxxii](#), [xxxii](#)

サンプル

コンパイル [89](#)

サンプルのコンパイル [89](#)

サンプルファイル [88](#), [88](#)

## し

詳細統計情報 [449](#)

持続サブスクライバ [5](#), [133](#), [133](#)

拡張可能なセキュリティ [288](#)

実装に関する注意事項

マルチキャスト [370](#)

証明書

ファイル名 [457](#)

## す

ストア

複数ストア [29](#)

## ストアファイル

- mstore の間隔 32
- show store コマンド 162
- show stores コマンド 164
- 複数ストアの設定 31
- データベースストア 294
- データベースストアの設定 295
- デスティネーションデフラグ 244
- デフォルト 30
- ファイルベースのストア 29

## スループット 115

## スレッド 115

- processor\_ids パラメータ 193
- フロー制御 85

## スワップ

- msg\_swapping パラメータ 199

## せ

## 静的キュー 50

## 静的トピック 50

## セキュリティ

- anonymous ユーザー 111
- 主要設定ファイル 265
- 無効化 110

## セキュリティの無効化 110

## 接続

- ネットワーク I/O 116
- マルチキャスト 370

## 接続ファクトリ 323

- パラメータ 235

## 設定

- LDAP 269
- 認証に使用する外部ディレクトリ 269

## セクタ、メッセージ 41

## そ

## ゾーン 502

## た

## 帯域幅

- マルチキャストの管理 372

## ち

## チャンネル

- 詳細統計情報 449

## 調整

- パフォーマンス 115

## て

## データ型の変換 44

## データベースストア 294

- stores.conf 297

- tibemsd.conf 296

- 設定 295

## データベースストアファイル

- Schema Export Tool 302

## テクニカルサポート xxxii, xxxii

## デスティネーション

- ブリッジ 78

## デスティネーションデフラグ機能

- file\_destination\_defrag 244

## デスティネーションのデフラグ 244

## デスティネーションの動的作成

- ワイルドカード 75

## デスティネーションのプロパティ 55

## デスティネーションブリッジ

- フロー制御 85

## デッドロック

- フロー制御 85

## デュアルステートフェイルオーバー 481

## テンポラリキュー 50

## テンポラリトピック 50

## と

- 動的デスティネーション 50
  - 作成 331
- トピック
  - 静的 50
  - テンポラリ 50
  - 動的 50
  - ルーティング先 511
- トピックの **export** プロパティ 58
- トピックの **import** プロパティ 59
- トピックのプロパティ 55
- トピックのプロパティのリスト 55
- トランザクション 13
- トレース 440
  - クライアント 211
  - トレースオプション 436

## な

- 長さの制限
  - 命名規則 121
- 名前の長さの制限 121

## ね

- ネットワーク I/O 接続
  - 調整 116

## は

- 管理者
  - パスワードの割り当て 120
- パスワード
  - admin 111
  - 設定 120, 136
- パフォーマンス調整 115

- パブリッシュとサブスクライブ
  - 例 93
  - メッセージングモデル 4

## ひ

- 非共有ステート 475
  - URL の指定 496
  - クライアントの設定 495
  - サーバーの設定 489
  - 実装 495
  - プロセス 480

## ふ

- ファイル、サンプル 88, 88
- ファイルベースのストア 29
- ファイル名
  - 証明書と鍵 457
- フェイルオーバー
  - 共有ステートの概要 474
  - 共有ステート 483
  - デュアルステート 481
  - 非共有ステート 475
- フォールトトレランス 12, 473
- フォールトトレランス URL 236
- フォールトトレランス スイッチオーバー 27, 493, 507
- 負荷分散 4, 57, 78, 239, 348, 417
- 負荷分散 URL 236
- 複数ストア 29
- ブリッジ 78
- フロー制御 83
  - 実行 83, 83
  - スレッドとデッドロック 85
  - デスティネーションブリッジの使用 85
  - マルチキャストの使用 84
  - ルートの使用 84
- プログラマ用チェックリスト 312
  - C 313
  - C# 319
  - Java 312

## プロパティ 16

- channel 56
- exclusive 56
- expiration 57
- export 58, 58
- flowControl 58
- global 59
- import 59, 59
- maxbytes 60
- maxmsgs 61
- maxRedelivery 61
- overflowPolicy 62
- prefetch 64
- redeliveryDelay 66
- secure 67
- sender\_name 68
- sender\_name\_enforced 68
- store 69
- trace 70
- 継承 50
- 定義 55
- キュー 55
- 継承 76
- トピック 55

プロパティタグ 16, 16, 16

プロパティの継承 50

プロパティの定義 55

分散トランザクション 13

## へ

変換、データ型 44

## ほ

ポイントツーポイント

例 92

メッセージングモデル 3

保護アクセス権限 263

## ま

前のリリースの『TIBCO Enterprise Message Service  
ユーザーガイド』からの変更点 xxvi

マルチキャスト

daemon 363

下位互換性 361

実装例 376

例 98

アーキテクチャ 376

コマンドラインオプション 364

サーバーエラー 388

実装に関する注意事項 370

接続 370

帯域幅の管理 372

デーモンエラー 387

トラフィックの制限 372

ネットワーク容量の決定 381

フロー制御 84

メッセージングモデル 5

ワイルドカード 74

マルチキャストデーモン

エラー 387

マルチキャストトラフィックの制限 372

マルチキャストのトラブルシューティング 385

一般的なヒント 385

接続 386

データ消失 386

## み

未配信メッセージキュー 20

## め

命名規則 121

## メッセージ

最大サイズ [22](#)

圧縮 [37](#)

確認応答 [38](#)

セレクタ [41](#)

トレース [440](#)

メッセージブール [198](#)

メッセージリスナー [45](#)

メッセージングモデル

パブリッシュとサブスクライブ [4](#)

ポイントツーポイント [3](#)

マルチキャスト [5](#)

メモリの予約 [199](#)

## ゆ

ユーザー [268](#)

admin [111](#)

外部で認証される [269](#)

ユーザー admin [120](#)

## ら

ライブラリファイル [314](#)

ラウンドロビン型キュー（非排他） [57](#)

## り

リンク [314](#)

## る

ルート

詳細統計情報 [449](#)

ゾーン [502](#)

フロー制御 [84](#)

## れ

例外リスナー [329](#)

## わ

ワイルドカード [73](#)

キュー [74](#)

動的に作成したデスティネーションでの使用 [75](#)

トピック [74](#)

mstore

間隔 [32](#)