UMDS (Version 6.13.1)

UMDS Release Notes

Contents

1 Introduction **5**

2 UMDS Version 6.13.1 **7**

2.1 Enhancements for UMDS 6.13.1 7

2.2 Fixed Limitations for UMDS 6.13.1 7

3 UMDS Version 6.12 **9**

3.1 Enhancements for UMDS 6.12 9

3.2 Fixed Limitations for UMDS 6.12 9

4 UMDS Version 6.0 **11**

4.1 Enhancements for UMDS 6.0 11

4.2 Fixed Limitations for UMDS 6.0 12

5 UMSD Version 6.0.200 **15**

5.1 Enhancements for 6.0.200 15

5.2 Fixed Limitations for 6.0.200 15

6 Known Problems and Limitations for UMDS **17**

Chapter 1

Introduction

This document describes the important changes made to the UMDS product. This document is cumulative from UMDS version 6.0 till now.

For policies and procedures related to Ultra Messaging Technical Support, see UM Support.

Note

> The UMDS product is based on the Ultra Messaging (UM) product. Users of UMDS are assumed to have familiarity with the UM product line. See the Ultra Messaging Documentation Set. In particular, review the UM Release Notes documentation for the UM version you are using.

Chapter 2

UMDS Version 6.13.1

The most-significant updates to UMDS version 6.13.1 is the introduction of persistence and client/server encryption (TLS).

2.1 Enhancements for UMDS 6.13.1

The following new features and enhancements apply to the UMDS product.

- **Persistence.** UMDS now supports receiving Ultra Messaging persisted messages. See **Using UMDS Persistence**
- **Client Encryption.** UMDS now supports TLS encryption between clients and the UMDS server. See **Using UMDS Client Encryption**.

2.2 Fixed Limitations for UMDS 6.13.1

The following bug fixes apply to the UMDS product.

Change Request	Description
11040	**FIXED:** *The* **Daemon Statistics** *feature has a slow memory leak.*

Chapter 3

UMDS Version 6.12

The most-significant update to UMDS version 6.12 is the introduction of Daemon Statistics to the UMDS server.

3.1 Enhancements for UMDS 6.12

The following new features and enhancements apply to the UMDS product.

- **UMDS Daemon Stats.** A new method of monitoring the UMDS server. The same information which is currently available on the server' web monitor can now be published on a normal UM topic. See **Daemon Statistics**.

Note

The Windows .NET client package is not offered in UMDS 6.12. This is because no changes were made to the client for version 6.12. **Users are directed to continue using the UMDS version 6.0 .NET client.** Be aware that although the Java client files are included in the UMDS 6.12 Linux package, no changes were made to the Java client as well. The files are included to have a consistent package.

3.2 Fixed Limitations for UMDS 6.12

The following bug fixes apply to the UMDS product.

Change Request	Description
10382	**FIXED:** *Running the UMDS server with a UM version of 6.11 or later generates the following benign warning message:* `Core-5688-28: WARNING: receiver config variable use_transport_thread is deprecated. Has no effect.` The warning was caused because the UMDS server is incompatible with Transport Threads (MTT), and actively disables it during initialization. This generates the warning message in UM 6.11 and beyond. The UMDS server now detects the version of the underlying UM and only disables MTT for versions that support it.

Change Request	Description
10002	**FIXED:** *Server can fail with a fatal assert:* `failed assertion "worker_conn==umdsd_src->worker_conn"`
9935	**FIXED:** *Server can fail with a segmentation fault in* `umdsd_stats_queue_process_cmds`.

Chapter 4

UMDS Version 6.0

UMDS version 6.0 includes all functionality and fixed limitations of version 6.0.200. Version 6.0 should be considered the *next* version after 6.0.200. Think of the ".200" version number as being like a negative number; 6.0.200 is less than 6.0.

The most-significant update to UMDS version 6.0 is the availability of Late Join for UMDS receive and sending client applications.

4.1 Enhancements for UMDS 6.0

The following new features and enhancements apply to the UMDS product.

- You can now use Ultra Messaging Late Join with UMDS Client receive applications, as well as with UMDS Client send applications.
- The UMDS Web Monitor user interface now includes more detailed displays with help tooltips, server and connection control buttons, and a configuration file display. The Web Monitor also displays more detailed message statistics, with messages broken out by type and number of bytes sent and received. The new Web Monitor has improved design and readability.
- The UMDS 6.0 Server is backward compatible with UMDS 1.1.1 clients.
- The UMDS Server console log now includes the IP addresses and port numbers of client connections.
- The example applications now monitor the number of source and receiver close acknowledgements. The example applications also monitor the number of requests that have timed out at the server. These applications log to the console, as appropriate, the number of requests sent and request cancellations received, the source and receiver closes sent, and close ACKs received. The example applications do not wait indefinitely, but continue after an arbitrary time. You can adjust the time limits in these examples as needed.
- You cannot use, and are no longer required to use, permissions options. UMDS now ignores all permission configuration settings.
- You can now use the Windows UMDS Client installer, `UMDS_6.0_client-install.exe`, for either 32-bit or 64-bit Windows.

4.2 Fixed Limitations for UMDS 6.0

The following bug fixes apply to the UMDS product.

Change Request	Description
7499	FIXED: *A UMDS Client cannot close a source or receiver while being disconnected from the server.* As a solution, sources and receivers can now call `close()` while disconnected from the server. When the client reconnects and re-established the closed element, the client library automatically issues the source or receiver delete command, completing the close process.
8168	FIXED: *The server might crash after taking too long to service requests to refresh a Web Monitor display, or to create or delete a client connection.*
8218	FIXED: *Authentication by application name fails.*
8418	FIXED: *You cannot specify the size of a request independently of the message size in the* `umdsrequest` *sample application.*
8482	FIXED: *The UMDS Server log inserts an empty line after each log message.*
8500	FIXED: *When a UMDS send fails and does not return an EWOULDBLOCK, the UMDS Server does not log an error message.*
8524	FIXED: *Wildcard Receiver throughput performance degrades with high numbers of topic matches.*
8530	FIXED: *The UMDS Server does not set the retransmission flag for OTR recovered messages before forwarding those messages to a UMDS Client.*
8542, 8547	FIXED: *With transport LBT-SMX you cannot receive messages or send requests.* As a solution, you now cannot configure transport LBT-SMX with UMDS.
8545	FIXED: *If you enable multithreaded transports, the UMDS Server encounters multiple error conditions.* As a solution, you can no longer enable multithreaded transports within the UMDS Server.
8561, 8822	FIXED: *UMDS Client applications might crash due to a situation where the application cannot create a debug log file, but then tries to write log entries to the nonexistent debug log file.*
8619	FIXED: *The UMDS Server cannot send error messages reported by a UMDS client before that UMDS Client starts to send.* As a solution, the following new states track which commands the UMDS Client has sent to the UMDS Server and which commands the UMDS Server has responded to. • SOURCE_CREATE • RECEIVER_CREATE • SOURCE_DELETE • RECEIVER_DELETE • CONNECT • LOGIN_COMPLETE • LOGOUT_COMPLETE
8656	FIXED: *The UMDS Web Monitor truncates longer topic names.*

Change Request	Description
8697	FIXED: *If you configure more workers than the host machine can support, the UMDS Server crashes with the following message:* `[emergency] FATAL: failed assertion [err==0] at line 1645 in ../../../../../src/umds/umdsd/umdsd_worker.c` As a solution, if you configure too many workers, you see a warning message.
8700	FIXED: *The UMDS Server needlessly issues the following warning:* `Core-5688-1793: WARNING: Requested receiver attributes will be ignored, previous receiver for topic [s] has already defined the attributes.`
8753, 8754	FIXED: *The UMDS Server might crash when receiving malformed packets.*
8777	FIXED: *The UMDS Server might crash when UMDS Client applications send at a high rate.*
8786	FIXED: *UMDS Clients might unexpectedly disconnect and reconnect due to a race condition.*
8912	FIXED: *The UMDS Server might crash when an enabled Web Monitor receives requests for details about an invalid connection.*
8930	FIXED: *The Java UMDS Client closes `System.err` when creating a debug log file.*
8951	FIXED: *If a message queue size limit is set to 0, the UMDS Server fails to discard messages older than the configured message age limit.*
8985	FIXED: *Example application server authentication fails if you connect and supply a username and password without an application name.* As a solution, you can now use command line option `-A` with the example applications to suppress sending their application name.
8986	FIXED: *The UMDS Client .NET example `umdssend` application reports statistics with erratic timing intervals.*
9006	FIXED: *The UMDS Server might crash under conditions of heavy loss when a receiver client disconnects.*
9057	FIXED: *When a UMDS Server disconnects from a Java or .NET UMDS Client, the client issues a NULL pointer exception.*
9058	FIXED: *While a UMDS Client connection is closing, you might see the following warning message:* `WARNING :Invalid state CLOSING for message 71`

Chapter 5

UMSD Version 6.0.200

UMDS 6.0.200 is a "Limited Availability" (LA) release which was made *prior* to version 6.0 "General Availability" (GA), and does not include the enhancements and fixed limitations of 6.0. Think of the ".200" version number as being like a negative number; 6.0.200 is less than 6.0.

The most-significant update for UMDS version 6.0.200 is the introduction of Request/Response capability.

5.1 Enhancements for 6.0.200

The following new features and enhancements apply to the UMDS product.

- Clients now have Request/Response capability.
- The UMDS Web Monitor has a `Quit Server` button and a `Disconnect this Client` button. These buttons are disabled by default. You can enable or disable these buttons with the `<server allow-shutdown-via-webmon=[0|1]>` option. This option applies to both types of button.
- The UMDS sample applications, libraries, and server report versions to the debug log. The sample applications also report the version in the `-h` help text and the server reports its version in the console output after startup.
- UMDS provides a single `anycpu` DLL for the .NET client library for use with both 32-bit and 64-bit applications.
- The UMDS Web Monitor displays version numbers for both the UMDS server and the installed Ultra Messaging version that the server has dynamically loaded.
- You can trigger a clean UMDS server shutdown with the UNIX signal SIGTERM or SIGINT.
- The UMDS 6.0.200 server is backward compatible with UMDS 1.1.1 clients.
- This release includes only .NET and Java APIs for UMDS clients. **The C client API and C++ wrapper are no longer available.**

5.2 Fixed Limitations for 6.0.200

The following bug fixes apply to the UMDS product.

Change Request	Description
8233	FIXED: *The UMDS Server exits when connection to a UMDS Client takes more than 120 seconds to complete.*
8497	FIXED: *For UMDS clients configured for arrival order delivery, received messages do not verify correctly.* With this fix, **you can NO LONGER configure a UMDS client for arrival order delivery**. If you attempt to configure arrival order delivery, the UMDS server resets to ordered delivery.

8522	FIXED: *The defaults values for server and client keepalive options are incorrect.*
	They now have the following values:
	`server-ka-interval 2,000 client-ka-threshold 10,000`
	`client-ka-interval 3,000 server-ka-threshold 11,000`

8664	FIXED: *A UMDS server configured to use Ultra Messaging configuration option `resolver_unicast_daemon (context)` might, upon startup, crash with the following message:* `[emergency] FATAL: failed assertion [src->res_ucast_daemons!=NULL] at line 2386 in ../../../../src/lib/lbm/lbmctx.c`
8683	FIXED: *A client might experience an exception with the following message:* `System.NullReferenceException: Object reference not set to an instance of an object...`

Chapter 6

Known Problems and Limitations for UMDS

The following open limitations applies to the UMDS product.

You should also review the known limitations for your underlying Ultra Messaging version. See the "Known Problems and Limitations" section in the UM Release Notes documentation for your UM version's, available at Informatica Ultra Messaging Documentation Sets.

Change Request	Description
8411	License keys including the UMSM product option are not recognized as valid when running a version of Ultra Messaging earlier than 6.5. The UMSM license is only required to run the System Monitoring components. Enabling monitoring of applications and other products does not require a license containing UMSM.
8579	The `UMDSMessage` `low_seqnum` field is not the low sequence number of a fragmented message as documented. Instead, it contains the `seqnum` field value.
8581	Sending request messages on a topic subscribed by a pre-6.0 UMDS client causes the client to disconnect from the UMDS server with the following log message: `WARNING :Error reading parameter 49:`**`com.latencybusters.umds.UMDS`**`+UMDSException: Bad data hdrid in UMDSMessageFormatter.parse()`
8582	The UMDS web monitor incorrectly displays `can-reqresp=1` on the client details page for older UMDS clients that do not have the capability to participate in request/response.
8586	If UM automatic monitoring is enabled on the server with an invalid monitoring configuration, the server does not properly report the configuration error, but instead issues log messages such as the following example: `Warning: Attempt to send data to tidx 0 when source not initialized`
8587	For a source that matches a wildcard receiver pattern, creates a per-topic queue, then ends, the per-topic queue is not deleted until the wildcard receiver gets deleted.
8695	If the UMDS Server is running with Ultra Messaging 6.7 or earlier, UMDS Client receivers do not correctly format responses to request messages that have traversed a UM Router from sources in a different Topic Resolution Domain. Instead of delivering such response messages, the originating requester logs the following warning message: `Core-6259-25: Deserialize Response: Context in domain 1 received response with no domain:...`

Change Request	Description
9078	The UMDS Server might crash under the following conditions: • LBM_DEBUG_MASK is set to a value that includes the 0x10 topic resolution bit. • The UMDS Server is running with Ultra Messaging 6.7 or earlier. To avoid this problem, do one of the following: • Do not set LBM_DEBUG_MASK to a value that includes the 0x10 topic resolution bit. • Upgrade to a version of Ultra Messaging later than 6.7.
9118	The `umdsresponse` sample application erroneously requires an argument for the wildcard command-line option `-W`. To work around this limitation, either supply a dummy argument, such as "<tt>-W dummy</tt>" or add a `case 'W':` near line 90 of the `umdsresponse.cs` file.
9129	You cannot set option `umq_queue_name`. If you set this option, instead of flagging error during the source create and exiting the source create, Ultra Messaging attempts to send and then logs the following warning message: `[warning] Umds-8499-2: LBM error while sending message: CoreApi-5688-707: Not currently registered with enough UMQ queue instances`
9130	You cannot use Request-Response with MIM.
9131	If you send a Request on a topic where no topic queue is configured, Ultra Messaging creates a per-topic queue anyway. This queue is unbound in size, although Requests do time out. For full control, configure a topic queue with size limit on these topics.
9132	If a wildcard receiver matches a topic for which no per-topic queue is configured, Ultra Messaging creates a per-topic queue anyway, but then still uses the default queue to deliver that topic's messages, and not the per-topic queue.
9133	If option `<server-reconnect>` is at the default settings of `0` and `client-write=yes`, and you have not set the UMDS Client to reconnect with the `setProperty()` API, a disconnected UMDS Client does not reconnect as expected. To insure proper behavior, use the `setProperty()` API to configure the UMDS Client for the desired reconnect mode.
9134	Server reports the client authentication indicated by the client not the authentication method used by the server. When the server is configured for authentication none, and the client supplies an application name and / or a user name and password, the server will report that authentication type basic was used, when it was not.