



# *UDISpec Changes from 0.86 to 0.90*

---

*1*

## ***1.1 Overview***

This document describes the changes to the UDI Specifications that occurred between Revision 0.86 and the current Revision 0.90.

The change history listed here is intended simply as a guide to the reader and does not document each change in detail. Instead, it refers to the sections that changed or were added and describes the changes briefly. The reader must refer to the actual specifications for details.

While the authors have attempted to make this a comprehensive list, there may be omissions. The 0.90 specifications themselves must be considered the ultimate authority.

Only normative changes are listed. Formatting and wording changes are not listed unless the previous version was so unclear as to likely have been misinterpreted by implementors.

## ***1.2 Changes That Affect All Specifications***

1. All mnemonic constants are now specified with actual values.
2. A terminology change was made that affected all specs: “interface operations” are now called “channel operations”; “interface indexes” are “ops indexes”; `xxx_interface_init` functions are now `xxx_ops_init`; `if_idx` is `ops_idx`.
3. Callee-side typedefs (`xxx_op_t`) for channel operations were changed to not include the “\*” for indirection. The ops arguments must now be static globals and do not have copy semantics, so stack variables cannot be used.
4. Header files in subdirectories (`meta/*.h` and `bus/*.h`) were moved to the top level. (Implementation-specific component header files should be moved to a subdirectory.)
5. Added `!locator` enumeration attribute and corresponding definitions for all metalanguages and bus bindings, as well as clarification of all other enumeration and filter attributes for metalanguages and bus bindings.

## 1.3 *UDI Core Specification Changes*

1. Compile-time versioning (`UDI_VERSION`) was added to a new General Requirements chapter.
2. The `cb_idx` passed to `_cb_init` operations cannot be zero.
3. Several new status codes were added.
4. `udi_version` (from the Initialization chapter) was eliminated, since versioning is now covered by Static Driver Properties.
5. Missing fields were added to `udi_limits_t`.
6. `udi_primary_region_init` in the Initialization chapter was modified to show the linkage between `child_bind_interface/parent_bind_interface` and `child_meta/parent_meta` in the Static Driver Properties.
7. Updated alignment requirements for `udi_mem_alloc` to require most restrictive alignment of the platform's natural alignments for long and pointer data types.
8. Clarified behavior of `udi_constraints_attr_set` w.r.t. `UDI_BUF_HEADER_SIZE` and `UDI_BUF_TRAILER_SIZE` (previously named `UDI_XFER_HEADER_SIZE` and `UDI_XFER_TRAILER_SIZE`; now moved to Buffer Management chapter).
9. Noted that if `udi_constraints_attr_set` fails that the *new\_constraints* shall return a handle to a constraints object specifying the same constraints values as the original passed in.
10. Removed *constraints* argument from callback for the `udi_constraints_propagate` operation.
11. The definition of the `UDI_BUF_INSERT` macro was fixed to actually do what it says and a new macro, `UDI_BUF_DUP`, was added.
12. `udi_buf_mark_for_return` was eliminated.
13. A “Buffer Tags” section was added to the Buffer Mgmt chapter. Various tag types were defined for network support and driver-specified regions. Tag utilities `udi_buf_tag_compute` and `udi_buf_tag_apply` were also added to the Buffer Mgmt chapter.
14. `udi_time_later` was removed.
15. Some new instance attribute classes were added (“@” for parent-visible attributes, “#” for message strings). Enumeration attributes now have a “!” prefix.
16. Updated WARNING on `udi_channel_close` to note that management channels are illegal for this operation. Clarified description of pending operations and `event_ind` deliveries (relative to anchoring).
17. `UDI_HANDLE_ID` was added to Tracing & Logging.
18. Trace classes and trace events were combined into a single trace events bitmask, of type `udi_trevent_t`, and several unneeded events were dropped. Metalanguage masks were replaced with metalanguage indexes.

19. Added **trace\_event** argument to `udi_log_write`; this allows `udi_log_write` to also call `udi_trace_write` to simultaneously log and trace a single event without requiring two driver calls to effect the same functionality.
20. Some `udi_snprintf` format string changes: `%b/%B` changed to `%a/%A`; new `%b[xXdu]` added. Added `udi_vsnprintf` (and corresponding ISO C varargs references in the types chapter).
21. A number of changes were made to the endian utilities, including dropping `udi_endian` and adding a number of bit-field and multi-byte helper macros.
22. `udi_channel_event_ind_cb_t` was renamed `udi_channel_event_cb_t` and some of its fields were changed.
  - 1) The union was removed as this is not possible in the MEI specification
  - 2) Added a `UDI_BUFFER_RECOVERED` event.
23. A number of changes to the Mgmt Meta were made, for hot plug and enumeration filters, including the addition of device management, child release, and final cleanup. `udi_trace_mod_req` was combined with resource usage information and renamed `udi_usage_ind`.
24. Renamed `udi_secondary_region_attach_ind` to `udi_region_attach_ind` and `udi_unbind_from_parent_req` to `unbind_parent_req`.
25. Renamed enumeration context to `child_context` and `bind_context` to `parent_context`.
26. The following convenience stub operation functions were added to the Mgmt Meta: `udi_region_attach_unused`, `udi_enumerate_no_children`, `udi_child_release_ind_unused`.
27. `udi_bind_cb_t` in the Management Metalanguage was modified: a `meta_idx` member was added and the `MA_context` was removed. Clarified for all Mgmt operations that the same control block must be returned in the response.
28. Removed assertions in second description paragraph of `udi_unbind_parent_req` regarding the statement that children will have been unbound by the time this operation is received. The new device management changes this position. Also removed flags argument; channel closed indications provide the corresponding information instead.
29. The `udi_unbind_parent_req` is not allowed to fail, therefore the status value was removed from `udi_unbind_parent_ack`.
30. Changed the `attr_value` of `udi_instance_attr_list_t` to `udi_ubit8_t` to prevent sign extension and added 32-bit value SET/GET macros. Also changed `attr_length` to `udi_ubit8_t` since value size is limited to 64 bytes.
31. The Generic I/O Metalanguage and an associated “Diagnostics Support” chapter were added to Section 4. Previously there was just a placeholder chapter.
32. `udi_mei_typespec_t` was renamed `udi_layout_t` and moved to the Fundamental Types chapter in the Core Spec, with some changes. The rest of MEI was moved into a new section in the Core Spec.
33. Most of the MEI interfaces changed to one degree or another.

## *Changes from 0.86*      *UDI Core Specification Changes*

---

34. MEI stub implementation macros were added (and the UDI\_MIN/UDI\_MAX macros were removed).
35. The Static Driver Properties, Packaging & Distribution Format, and Build & Packaging Tools chapters were added to Section 5.
36. A chapter was added to summarize the items that need to be specified in an ABI binding.

# *UDI Physical I/O Specification Changes Changes from*

---

## **1.4 UDI Physical I/O Specification Changes**

1. An introductory section was added that includes UDI\_PHYSIO\_VERSION and extensions to Static Driver Properties.
2. Clarifications were made in the specification and usage of DMA block vector segments.
3. The DMA constraints attribute, UDI\_DMA\_DATA\_ALIGNMENT\_BITS, was replaced with a per-element UDI\_DMA\_ELEMENT\_ALIGNMENT\_BITS.
4. udi\_dma\_prealloc was renamed udi\_dma\_prepare and the flags definition was changed.
5. The flags argument was dropped from udi\_dma\_buf\_unmap and the “sync” rules for udi\_dma\_buf\_map were adjusted.
6. A NULL\_PIO\_HANDLE was added.
7. A UDI\_PIO\_UNALIGNED PIO attribute was added.
8. A new service call, udi\_pio\_abort\_sequence, was added in support of “region kill”.
9. Several changes and clarifications to udi\_pio\_trans\_t:
  - a. The shift count for UDI\_PIO\_SHIFT\_LEFT and UDI\_PIO\_SHIFT\_RIGHT must not exceed 32.
  - b. The maximum tran\_size is now 32 bytes.
  - c. The UDI\_PIO\_LABEL op was added, and UDI\_PIO\_BRANCH changed to branch to a label (with the same operand) instead of an absolute index.
  - d. Various nonsensical combinations of CSKIP, REPEAT, BRANCH, END, and LABEL ops were made explicitly illegal.
  - e. Last op must always be END.
  - f. Clarification: pacing applies only to IN, OUT, IN\_IND, and OUT\_IND ops.
  - g. Clarification: arithmetic operations wraps around modulo the operand size.
10. The pio\_offset of udi\_pio\_map was changed to base\_offset and assertions about alignment requirements relative to the members of the corresponding transaction list were made.
11. Added constants to define the logical registers for the PIO trans list operations.
12. The udi\_pio\_probe service call was added.
13. Many of the names in the Bus Bridge Metalanguage were changed.
14. Clarified return status codes and meanings for udi\_intr\_attach\_ack.
15. The definition of pci\_unit\_address in the PCI Bus Binding was changed.

## ***1.5 UDI Network Driver Specification Changes***

1. Significant updates to the network metalanguage based on review comments from SCO, IBM, and Compaq; highlights follow but many miscellaneous changes were also made.
2. Receive flow control implemented in a similar (but reversed) fashion to the transmit flow control mechanism.
3. Added description of checksum handling vis-a-vis the corresponding buffer tags added to the core specification.
4. Moved all typed structures to be explicit control block values rather than passed by buffer.
5. Reduced control operations to generic network control operations; any technology or adapter specific parameters should be handled via instance attributes.
6. Updated statistics lists to allow SNMP MIB-II support for most technologies.
7. Added support for ALLMULTI multicast reception.
8. Updated tracing and logging events and descriptions per standard extensions defined as part of 0.86.

# *UDI SCSI Driver Specification Changes Changes from*

## **1.6 UDI SCSI Driver Specification Changes**

1. Fixed problems from 0.86 with no-longer-valid control block types (`xxx_req_cb_t`), and naming (changed `_con` to `_ack`).
2. Added `UDI SCSI_VERSION` preprocessor symbol definition.
3. Added SCSI enumeration attributes, filter attributes, locator attribute, and parent-visible attributes.
4. Added support for multi-lun PDs (PDs which access multiple LUNs on a single bind channel):
  - a. Added semantics to the use of the CDB memory in the `scsi_io_request` (see the `cdb_ptr` definition in the `udi_scsi_io_cb_t` reference page)
  - b. Added a “!scsi\_multi\_lun” enumeration attribute which the system can set when it’s doing directed enumeration to indicate that a particular PD instance has this property.
5. Moved all SCSI-specific status code values to the front of the chapter, in section 2.7, and fine-tuned the set of available status codes.
6. Added `unbind_ack_op` and `io_nak_op` to `udi_scsi_pd_ops_t`; added `unbind_req_op` to `udi_scsi_hd_ops_t`.
7. Renamed `max_cdb_size` parameter to `cdb_mem_size` in `udi_scsi_io_cb_init`.
8. Changed `udi_scsi_bind_req` and `udi_scsi_bind_cb_t` as follows:
  - a. Moved `events` parameter in `udi_scsi_bind_req` to `udi_scsi_bind_cb_t`, and changed the name of an event code from `UDI SCSI_EVENT_AEN SCSI` to `UDI SCSI_EVENT_AEN`. Also, added constant definitions for each of the event codes.
  - b. Added `bind_flags` and `aen_buf_size` parameters to the `scsi_bind_req` operation, and reordered the rest of the parameters.
  - c. Added exclusive access support so a PD can request exclusive access to its LUN: added two new bind flags - `UDI SCSI_BIND_EXCLUSIVE` and `UDI SCSI_TEMP_BIND_EXCLUSIVE`.
  - d. Changed `timeout_granularity` from a `udi_ubit32_t` to a `udi_time_t`.
9. Added `scsi_unbind_req/ack` operations.
10. Changed `udi_scsi_io_req/ack` and `udi_scsi_io_cb_t` as follows:
  - a. Added constant values for the I/O request flags and the Task Attributes. Added the `UDI SCSI_OVERRUN` flag.
  - b. Changed the way the `data_buf` and `data_len` fields in I/O request and event operations are used.
  - c. Added/removed/changed status code names (see item #4 above).
  - d. Added a `scsi_io_nak` operation (i.e., split `scsi_io_ack` into `scsi_io_ack` and `scsi_io_nak` to streamline the normal path).
  - e. Changed `udi_scsi_io_con` to `udi_scsi_io_ack`, and removed the `status` and `sense_buf` parameters (they only exist on the `scsi_io_nak`).

11. Changed `udi_scsi_ctl_req/ack` and `udi_scsi_ctl_cb_t` as follows:
  - a. Removed the union from `udi_scsi_ctl_cb_t` (leaving the embedded members).
  - b. Added constant values for the `ctrl_func` codes
  - c. Changed `UDI SCSI_CTL_CLEAR_CA` to `UDI SCSI_CTL_CLEAR_ACA`.
  - d. Changed the corresponding status codes (see item #4 above).
12. Changed `udi_scsi_event_ind/res` and `udi_scsi_event_cb_t` as follows:
  - a. Removed the union from `udi_scsi_event_cb_t` (leaving the embedded member).
  - b. Added the `udi_scsi_event_ind_unused` convenience stub function.

# *UDI SCSI Driver Specification Changes Changesfrom*