UDI Spec 0.90 Errata *1*

1.1 Overview

This document describes substantive normative corrections to the Revision 0.90 UDI Specifications. Only normative errors that would affect the developers of UDI environment implementations or drivers will be noted in this document. Formatting and wording errors are not listed unless the 0.90 version was so unclear as to likely have been misinterpreted by implementors.

[Initials in square brackets indicate owners for making corresponding specification changes.]

1.2 UDI Core Specification Corrections

1. UDI_SPECIFIC_STATUS_MASK, defined in section 8.8.1, should be `0x7FFF` instead of `0xEFFF`. *[DONE]*

2. In 8.8.2, "Data Layout Specifier", the type for `udi_layout_t` should be "`udi_ubit8_t`", not "`udi_ubit8_t *`". The following sentence should read, "A data layout specifier consists of an array of one or more `udi_layout_t` layout elements." *[DONE]*

3. In Table 8-4, UDI_DL_TIMESTAMP_T should be removed. Timestamps are not transferable. *[MWE]*

4. At the end of the DESCRIPTION for **udi_constraints_attr_combine** (page 12-11), a new paragraph should be added:

   "`udi_constraints_attr_combine` can also be used to create a new constraints object with all attributes set to default values. This is done by setting both `src_constraints` and `dst_constraints` to NULL_CONSTRAINTS. It is illegal to set `src_constraints` to NULL_CONSTRAINTS if `dst_constraints` is not also null." *[KDG]*

5. On page 13-21, **udi_tagtype_t**, the value for UDI_BUFTAG_DRIVER8 should be "`(1U<<31)`". *[LW]*

6. In Table 15-1 and throughout the Instance Attributes chapter, the prefix character for Private Persistent attributes should be "%" and there should be no prefix for Enumeration attributes. *[KDG]*

7. For `udi_instance_attr_set` and `udi_instance_attr_get`, ***`enumeration_context`*** should be ***`child_context`***. *[KDG]*

8. On page 16-6, in the DESCRIPTION of `udi_channel_spawn`, the second sentence of the second paragraph should be replaced with, "The ***`callback`*** routine is called once the local end of the channel has been created and, if specified, anchored. The other region may or may not have completed its end of the spawn at this point."

   The first sentence on the next page should read, "It is the drivers' responsibility to ensure that both ends have completed spawning and are anchored and ready to go before invoking any operations on the channel." *[JP]*

9. The first sentence of the WARNINGS for `udi_channel_close` is unclear. It could incorrectly be interpreted to imply that loose ends may not be closed. This sentence should read: "`udi_channel_close` most not be used with a channel handle that has been passed to another region." *[JP]*

10. On page 17-3, **udi_trevent_t**, the value for UDI_TREVENT_LOG should be "`(1U<<31)`". *[LW]*

11. A new utility function is available, `udi_offsetof()`, which behaves as ISO C `offsetof()`, but has a value of type `udi_size_t` instead of `size_t`. *[DONE]*

12. In the example at the end of 19.5.1.1, the two references to `UDI_MDEP` should be `UDI_MBSET`. In 19.5.2.2, on page 19-29, the reference to `UDI_MBDEP` should be `UDI_MBSET`. *[MWE]*

13. In 19.5.2.2, "Multi-Byte Macros", on page 19-30, the definitions of UDI_MBGET_3 and UDI_MBGET_4 are each missing a backslash at the end of their second lines. *[MWE]*

14. In the DESCRIPTION of UDI_ENDIAN_SWAP_16/32 on page 19-32, the functional interfaces should be shown as:

```
udi_ubit16_t UDI_ENDIAN_SWAP_16 (
                 udi_ubit16_t data16 );

udi_ubit32_t UDI_ENDIAN_SWAP_32 (
                 udi_ubit32_t data32 );
```

    [MWE]

15. On pg 21-30, UDI_ATTR32_SET should be UDI_ATTR32_INIT. The new UDI_ATTR32_SET is as follows:

```
#define UDI_ATTR32_SET(aval,v) \
        { udi_ubit32_t vtmp = (v); \
          (aval)[0] = (vtmp) & 0xff; \
          (aval)[1] = ((vtmp) >> 8) & 0xff; \
          (aval)[2] = ((vtmp) >> 16) & 0xff; \
          (aval)[3] = ((vtmp) >> 24) & 0xff; }
```

    [KDG]

16. In the MEMBERS block of **udi_enumerate_cb_t** on page 21-33, the first sentence for ***attr_list*** should read, "***attr_list*** is a pointer to an array of `udi_instance_attr_list_t` structures, allocated and owned by the MA." *[JP]*

17. A new argument should be added to `udi_enumerate_ack` (page 21-37): `udi_index_t` ***meta_idx***, defined as follows, "***meta_idx*** indicates (one of) the metalanguages usable with this child device. If a driver calls `udi_enumerate_ack` multiple times for the same ***child_context*** but different ***meta_idx*** values, the environment may bind child drivers to any or all such metalanguages. *[JP]*

18. In 21.8.3, the reference to "pio handles, dma handles, etc" should be removed. *[JP]*

19. In the Generic I/O Metalanguage, the type of the ***params_layouts*** argument to the `udi_gio_xfer_cb_init` function should be "`udi_layout_t *`" instead of "`udi_layout_t`". The corresponding entry in the ARGUMENTS section should start with ***params_layout*** instead of ***params_typespec***. *[DONE]*

20. In the Generic I/O Metalanguage, in the DESCRIPTION of **udi_gio_xfer_ack**, the first sentence of the third paragraph should read, "If ***op*** includes `UDI_GIO_DIR_READ`, ***data_len*** must be set to the amount of data actually read; otherwise, if ***op*** includes UDI_GIO_DIR_WRITE, ***data_len*** must be set to the amount of data actually written; if neither direction flag is included, ***data_len*** is ignored." *[LW]*

21. In 26.2, "Structures", in the MEI chapter, the ***cb_layout*** member of the `udi_mei_op_template_t` structure should be named ***visible_layout***. *[DONE]*

22. In 26.2, “Structures”, in the MEI chapter, the ***op_templates*** member of the `udi_mei_ops_template_t` structure should be a pointer, not an array:

```
const udi_mei_op_template_t *op_templates;
```

[LW]

23. In 26.4, “Initialization Interfaces”, in the MEI chapter, the type of the ***inline_layouts*** argument to the `udi_mei_cb_init` function should be “`udi_layout_t **`” instead of “`udi_layout_t *`”. *[DONE]*

24. In 26.5, “Front-End Stub Interfaces”, in the MEI chapter, the type of the ***meta_ID*** argument to the `udi_mei_call` function should be “`udi_mei_meta_id_t`” instead of “`void *`”. *[LW]*

25. In 26.6, “MEI Stub Implementation Macros”, in the MEI chapter, the following line:

```
_UDI_ARG_LIST_##argc args ); \
```

should be:

```
_UDI_ARG_VARS_##argc ); \
```

[LW]

26. In 26.6, “MEI Stub Implementation Macros”, in the MEI chapter, both occurrences of the following line:

```
(*op_name##_op_t)op) ( \
```

should be:

```
(*op_name##_op_t *)op) ( \
```

[LW]

27. In 26.6, “MEI Stub Implementation Macros”, in the MEI chapter, the `UDI_ARG_LIST_0` through `UDI_ARG_LIST_7` macros should be removed. *[LW]*

28. In 26.6, “MEI Stub Implementation Macros”, in the MEI chapter, the `_UDI_ARG_LIST_1` through `_UDI_ARG_LIST_7` macros should be defined as:

```
#define _UDI_ARG_LIST_1(a)              ,a arg1
#define _UDI_ARG_LIST_2(a,b)            ,a arg1,b arg2
#define _UDI_ARG_LIST_3(a,b,c)          \
         ,a arg1,b arg2,c arg3
#define _UDI_ARG_LIST_4(a,b,c,d)        \
         ,a arg1,b arg2,c arg3,d arg4
#define _UDI_ARG_LIST_5(a,b,c,d,e)      \
         ,a arg1,b arg2,c arg3,d arg4,e arg5
#define _UDI_ARG_LIST_6(a,b,c,d,e,f)    \
         ,a arg1,b arg2,c arg3,d arg4,e arg5,f arg6
#define _UDI_ARG_LIST_7(a,b,c,d,e,f,g) \
         ,a arg1,b arg2,c arg3,d arg4,e arg5,f arg6,g arg7
```

[LW]

29. At the bottom of 28.6.3, the first two bullet items should be replaced with:

    - Orphan driver instances are never bound to parents, so `udi_bind_to_parent_req` is never called in an orphan driver. *[KDG]*

30. In 28.6.5, the “device” declaration syntax should be:

```
device <msgnum> <meta_idx> \
     { <attr_name> <attr_type> <attr_value> }
```

    and the second sentence of the second following paragraph should read, “The set of valid enumeration attribute names is specified by the instance attribute bindings of the selected parent metalanguage, as indicated by a “parent_meta” declaration with matching `<meta_idx>`. The last sentence of this paragraph should be removed. Another paragraph should be added, that reads:

    “If two ‘device’ declarations for the same driver use the same `<msgnum>`, the environment may bind multiple parents of different types (as indicated by the `<meta_idx>` values, which must be different for each of these ‘device’ declarations) to this driver.” *[KDG]*

31. In the Sample Static Driver Properties File (page 28-21), the following declarations are missing:

```
parent_meta 1 udi_physio
child_meta 2 udi_network
```

    [KDG]

32. In the Sample Static Driver Properties File (page 28-21), the “device” declaration should read:

```
device 5 1 bus_type string pci pci_vendor_id ubit32 0x1234 \
     pci_device_id ubit32 19
```

    [KDG]

33. In the Packaging & Distribution chapter, on page 29-2, the directory hierarchy listing (starting with `/udi-dist.1`) should show `udiprops.txt` in the `src` subdirectory rather than directly in `<comp_zzz>`. *[KDG]*

1.3 UDI Physical I/O Specification Corrections

1. On page 2-8, **udi_scgth_t**, the value for UDI_SCGTH_DRIVER_MAPPED should be "`(1U<<7)`". *[JP]*

2. On page 2-26, **udi_dma_mem_alloc**, the value for UDI_DMA_NEVERSWAP should be "`(1U<<7)`". *[JP]*

3. On pages 3-15 through 3-19 (**udi_pio_trans_t**), `tran_size` now encodes the transaction size as a power of two; that is, the transaction size, in bytes, equals `(2^tran_size)`. All `tran_size` values must be between 0 and 5, inclusive. For UDI_PIO_BRANCH, UDI_PIO_LABEL, and UDI_PIO_END, `tran_size` must be zero. *[KDG]*

4. The description of UDI_PIO_CSKIP in Table 3-3 should now read, "Conditional skip. The value in `reg` is compared with zero. The `operand` value selects a condition code from Table 3-7 to determine the type of comparison. If the condition is TRUE, the following trans list element will be skipped."

   The description of UDI_PIO_CSKIP preceding Table 3-7 is superceded by the above definition.

   Table 3-7, PIO Condition Codes, is replaced with the following:

Mnemonic	Value	Condition Description
UDI_PIO_Z	`0`	`reg == 0`
UDI_PIO_NZ	`1`	`reg != 0`
UDI_PIO_NEG	`2`	`reg < 0` [signed]
UDI_PIO_NNEG	`3`	`reg >= 0` [signed]

   [KDG]

5. On pages 3-15 through 3-19 (**udi_pio_trans_t**), the three opcodes UDI_PIO_REP, UDI_PIO_REP_ZP, and UDI_PIO_REP_ZA, should be replaced with UDI_PIO_REP_IN_IND (0xFA), UDI_PIO_REP_OUT_IND (0xFB) and UDI_PIO_BARRIER (0xFC).

   See `http://stage.sco.com/cgi-bin/udi/show.cgi/23` for details on the new form of repeat operation. All ordering restrictions with respect to repeat operations are removed.

   The new UDI_PIO_BARRIER replaces the special-case use of UDI_PIO_SYNC and UDI_PIO_SYNC_OUT with `tran_size` of zero. If ***`operand`*** is 0, the barrier is affected by all PIO device transactions; if it is UDI_PIO_OUT, the barrier is affected only be PIO device output transactions. *[KDG]*

6. On page 3-19, in the description of UDI_PIO_END, the first two sentences should read, "The last element of a trans list must be either a UDI_PIO_END operation or a UDI_PIO_BRANCH operation. Upon reaching a UDI_PIO_END operation anywhere in the list, processing of the trans list is terminated." The second-to-last paragraph on that page should have the reference to reaching the end of the trans list removed. *[KDG]*

7. On pages 3-20 and 3-21, references to UDI_HW_PROBLEM should be replaced with UDI_STAT_HW_PROBLEM. *[KDG]*

8. For **udi_bus_device_ops_init**, page 4-3, there should be another member of `udi_bus_device_ops_t`, after ***bus_bind_ack_op***: "`udi_bus_unbind_ack_op_t *`***bus_unbind_ack_op***". This new operation is defined as:

   ```
   void udi_bus_unbind_ack (
         udi_channel_t target_channel,
         udi_bus_bind_cb_t *bus_bind_cb );
   ```

   [JP]

9. For **udi_bus_bridge_ops_init**, page 4-5, there should be another member of `udi_bus_bridge_ops_t`, after ***bus_bind_req_op***: "`udi_bus_unbind_req_op_t *`***bus_unbind_req_op***". This new operation is defined as:

   ```
   void udi_bus_unbind_req (
         udi_channel_t target_channel,
         udi_bus_bind_cb_t *bus_bind_cb );
   ```

   [JP]

10. In the description of ***interrupt_idx*** for `udi_intr_attach_cb_t`, ***enumeration_context*** should be ***child_context***. The first sentence of this paragraph should read, "***interrupt_idx*** is used to select one of possibly several interrupt sources from the interrupt handler's device that are managed by a particular interrupt dispatcher." *[JP]*

11. In section 4.5, the name of the enumeration attribute should be "`bus_type`", not "`!bus_type`" (no exclamation point). All enumeration, filter, and locator attributes listed in the Introduction to Bus Bindings and PCI Bus Bindings chapters should have no exclamation point prefix. *[KDG]*

12. In Table 6-1, PCI Enumeration Attributes, the SIZE for "`bus_type`" should be 4. *[KDG]*

1.4 UDI Network Driver Specification Corrections

1. On page 1-57, **udi_net_rx_cb_t**, the value for UDI_NET_RX_OTHER_ERR should be "`(1U<<7)`". *[KWQ]*

2. All enumeration, filter, and locator attributes listed in Section 1.6 should have no exclamation point prefix. *[KWQ]*

1.5 UDI SCSI Driver Specification Corrections

1. All enumeration, filter, and locator attributes listed in Section 2.6 should have no exclamation point prefix. The TYPE for `scsi_inquiry` should be UDI_ATTR_ARRAY8. The TYPE for `scsi_vendor_id` and `scsi_product_id` should be UDI_ATTR_STRING; the SIZE for `scsi_vendor_id` should be "1..9"; the SIZE for `scsi_product_id` should be "1..17".

   The paragraphs for `scsi_vendor_id` and `scsi_product_id` should read:

   The "`scsi_vendor_id`" attribute provides an up to 9-byte vendor_id string from the SCSI INQUIRY data as a null-terminated string without trailing spaces.

   The "`scsi_product_id`" attribute provides an up to 17-byte product_id string from the SCSI INQUIRY data as a null-terminated string without trailing spaces. *[MWE]*

2. In the example on page 2-8, `ddd_scsi_pd_ops` should have another member, `ddd_scsi_unbind_ack`, after `ddd_scsi_bind_ack`. *[MWE]*

3. In the example on page 2-9, `ddd_scsi_hd_ops` should have another member, `ddd_scsi_unbind_req`, after `ddd_scsi_bind_req`. *[MWE]*