Errata for IMPI version 0.0

last updated: March 26, 2003

This document contains additions, deletions, clarification, and corrections to the IMPI version 0.0 specification document. The IMPI version 0.0 specification is available at `http://impi.nist.gov/IMPI` and is also published in the *Journal of Research of the National Institute of Standards and Technology*, May-June 2000, Vol. 105 No. 3, pp 349–428, which can be down-loaded at:

`http://nvl.nist.gov/pub/nistpubs/jres/105/3/cnt105-3.htm`

Changes are in order by page number in the IMPI version 0.0 specification document. The page numbers in the *Journal of Research* (JoR) version of the IMPI specification, shown in parenthesis below, are offset by 354 from the stand-alone version of the IMPI specification. Each entry in this errata is dated so that new entries can more easily be found.

- Section 2.3.5, page 17 (JoR p. 371), lines 11–12 (Sept 2000)

Change:

IMPI_H_ACKMARK Each client's array contains the `IMPI_Pk_ackmark` values for its hosts. Each value is an `IMPI_Uint4`. Section 3.9 mandates some restrictions on this value.

To:

IMPI_H_ACKMARK Each client's array contains the ackmark values for its hosts. Each value is an `IMPI_Uint4`. The smallest value specified by any host determines the value of `IMPI_Pk_ackmark` (see Section 3.6 Packet Protocol). Section 3.9 mandates some restrictions on this value.

- Section 2.3.5, page 17 (JoR p. 371), lines 14–16 (Sept 2000)

Change:

IMPI_H_HIWATER Each client's array contains the `IMPI_Pk_hiwater` values for its hosts. Each value is an `IMPI_Uint4`. Section 3.9 mandates some restrictions on this value.

To:

IMPI_H_HIWATER Each client's array contains the hiwater values for its hosts. Each value is an `IMPI_Uint4`. The smallest value specified by any host determines the value of `IMPI_Pk_hiwater` (see section 3.6 Packet Protocol). Section 3.9 mandates some restrictions on this value.

- Section 2.5, page 22 (JoR p. 376), lines 10–22 (Mar 2, 2001)

Change:

IMPI_CLIENT_SIZE returns as the keyvalue the number of IMPI clients included in the communicator.

IMPI_CLIENT_COLOR returns as the keyvalue a number between 0 and (IMPI_CLIENT_SIZE -1. The value returned indicates with which client the querying rank is associated. The relative ordering of colors corresponds to the ordering of host ranks in MPI_COMM_WORLD.

IMPI_HOST_SIZE returns as the keyvalue the number of IMPI hosts included in the communicator.

IMPI_HOST_COLOR returns as the keyvalue a number between 0 and IMPI_HOST_SIZE-1. The value returned indicates with which client the querying rank is associated. The relative ordering of colors corresponds to the ordering of host ranks in MPI_COMM_WORLD.

To:

IMPI_CLIENT_SIZE returns as the attribute value the number of IMPI clients included in the communicator.

IMPI_CLIENT_COLOR returns as the attribute value an integer between 0 and `num_clients - 1`, where `num_clients` is value of the IMPI_CLIENT_SIZE attribute. This value indicates with which client the querying rank is associated. The relative ordering of colors corresponds to the ordering of host ranks in MPI_COMM_WORLD.

IMPI_HOST_SIZE returns as the attribute value the number of IMPI hosts included in the communicator.

IMPI_HOST_COLOR returns as the attribute value an integer between 0 and `num_hosts - 1`, where `num_hosts` is the value of the IMPI_HOST_SIZE attribute. This value indicates with which host the querying rank is associated. The relative ordering of colors corresponds to the ordering of host ranks in MPI_COMM_WORLD.

Each of these attribute values is returned directly as an `INTEGER` in the Fortran binding of MPI and indirectly as a pointer (`int *`) in the C binding of MPI.

- Section 2.5, page 22 (JoR p. 376), line 26 (Oct 12, 2000)

Change:

IMPI_COMM_SPLIT

To:

MPI_COMM_SPLIT

- Section 3.6, page 26 (JoR p. 380), line 31 (Sept 2000)

Change:

requiring a protocol ACK to be send back.

To:

requiring a protocol ACK to be sent back.

- Section 4.2, page 32 (JoR p. 386), line 27 (Oct 12, 2000)

Change:

int local_master_rank(int n, MPI_Comm comm)

To:

int local_master_rank(int r, MPI_Comm comm)

- Section 4.8, page 38 (JoR pp. 392), line 12 (Feb 25, 2003)

Change:

```
MPI_Group_size(lcomm, &lgsize)
```

To:

```
MPI_Comm_size(lcomm, &lgsize)
```

- Section 4.8, page 38 (JoR pp. 392), line 23,24 (Feb 25, 2003)

Change:

```
MPI_INT
```

To:

```
IMPI_Uint8
```

- Section 4.11, page 44 (JoR p. 398), lines 9–10 (Feb 26, 2001)

Change:

```
mynum = master_num(myrank, comm);
rootnum = master_num(root, comm);
```

To:

```
mynum = local_master_num(myrank, comm);
rootnum = local_master_num(root, comm);
```

- Section 4.10, page 42 (JoR pp. 396), lines 18, 21, 37, 43 (Jan 22, 2003)

Change:

```
rank_master
```

To:

```
master_rank
```

- Section 4.10, page 43 (JoR p. 397), line 3 (Jan 22, 2003)

Change:

```
peer = rank | mask;
```

To:

```
peer = mynum | mask;
```

- Section 4.10, page 43 (JoR pp. 397), line 5 (Jan 22, 2003)

Change:

```
rank_master
```

To:

```
master_rank
```

- Section 4.12, page 46 (JoR p. 400), line 26 (Mar 2, 2001)

Change:

```
MPI_Get_count(status, MPI_BYTE, &count);
```

To:

```
MPI_Get_count(&status, MPI_BYTE, &count);
```

- Section 4.12, page 47 (JoR p. 401), line 11 (Mar 2, 2001)

Change:
```
MPI_Get_count(status, MPI_BYTE, &count);
```
To:
```
MPI_Get_count(&status, MPI_BYTE, &count);
```

- Section 4.13, page 49 (JoR p. 403), line 19 (Mar 2, 2001)

Change:
```
MPI_Get_count(status, MPI_BYTE, &count);
```
To:
```
MPI_Get_count(&status, MPI_BYTE, &count);
```

- Section 4.13, page 50 (JoR p. 404), line 1 (Mar 2, 2001)

Change:
```
MPI_Get_count(status, MPI_BYTE, &count);
```
To:
```
MPI_Get_count(&status, MPI_BYTE, &count);
```

- Section 4.13, page 50 (JoR p. 404), line 32 (Jun 18, 2001)

Change:
```
MPI_Send(sbuf, scount, sdtype, root, IMPI_GATHER_TAG, comm);
```
To:
```
MPI_Send(sbuf, scount, sdtype, root, IMPI_GATHERV_TAG, comm);
```

- Section 4.16, page 56 (JoR p. 410), lines 22-26 (Jan 22, 2003)

Change:
```
        MPI_Recv(tmpbuf, count, dtype, master_rank(i, comm),
              IMPI_REDUCE_TAG, comm, &status);
     }

     call reduction function op on tmpbuf (invec)
        and rbuf (inoutvec);
```

To:
```
        MPI_Recv(tmpbuf, count, dtype, master_rank(i, comm),
              IMPI_REDUCE_TAG, comm, &status);
        call reduction function op on tmpbuf (invec)
           and rbuf (inoutvec);
     }
```

The above change moves the call to the reduction function to be within the `for` loop.

- Section 4.17, page 58 (JoR p. 412), lines 22-25 (March 26, 2003)

Change:

```
if (myrank == 0) {
   for (i = 0; i < (nprocs - 1); i++) {
      count += rcounts[i];
   }
```

To:

```
for (i = 0; i < (nprocs - 1); i++) {
   count += rcounts[i];
}
if (myrank == 0) {
```

The above change moves the initialization of the variable `count` to be outside of the `if (myrank == 0)` clause.

- Section 4.20, page 60 (JoR pp. 396), line 3 (Feb 10, 2003)

Change:

```
int i, total_rcount;
```

To:

```
int i, total_rcount, mynum;

MPI_Comm_rank(comm, &mynum);
```

- Section 4.20, page 60 (JoR p. 414), line 10 (Feb 26, 2001)

Change:

```
MPI_Bcast(rbuf, total_rcount, rdtype, 0, comm);
```

To:

```
if (mynum == 0) {
    Concatenate, in rank order, all of the process send buffers in rbuf
    into a temporary buffer, tmpbuf.
}
MPI_Bcast(tmpbuf, total_rcount, rdtype, 0, comm);
if (mynum != root) {
    Unpack tmpbuf into rbuf using offsets in displs.
}
```

/*
Notice that instead of packing and unpacking using the temporary buffer `tmpbuf`, a derived datatype could be used. This is an implementation decision. The requirement specified here is only that a single broadcast of the concatenated set of send buffers, from rank 0, is used to complete the Allgatherv operation.
*/

- Section 4.16, page 55 (JoR p. 409), line 35 (Oct 13, 2000)

Change:

`MPI_op`

To:

`MPI_Op`

- Section 4.16, page 55 (JoR p. 409), line 44 (Oct 13, 2000)

Change:

`MPI_op`

To:

`MPI_Op`

- Section 4.16, page 57 (JoR p. 411), line 30 (Feb 28, 2001)

Change:

`MPI_Comm_root`

To:

`MPI_Comm_rank`

- Section 4.16, page 57 (JoR p. 411), line 33 (Oct 17, 2000)

Change:

`return(MPI_Send` ...

To:

`return MPI_Send` ...

- Section 4.21, page 60 (JoR p. 414), line 17 (Oct 13, 2000)

Change:

`MPI_op`

To:

`MPI_Op`