

MPIR-2

Loading the mpir dll

- `const char** mpir_dll_locations`: null-terminated array of null-terminated strings
- Mpi implementation needs to set this ptr in an “atomic” way. I.e., the ptr is either null, or points to a valid dll locations
 - Mpi implementation needs to make up its mind!!! No changes allowed
- Cannot generate an `MPIR_Breakpoint` event, because it would require reading `MPIR_Debug_state`, but the callback to read `MPIR_Debug_state` is not ready yet.
- Need `mpir_dll_locations_ready()` so debugger can plant a breakpoint
 - Can this be called more than once? What if you call it the second time and the debugger was in the process of reading the `mpir_dll_locations` – how can the mpi implementation know when to free it ? Two solutions: dbg copy the vector, or mpi just frees it after the debugger resumes the starter process
 - Can this function take a parameter? John: It’s very platform dependent, not reliable

Initialization

- All functions should return `mpir_status`
- `mpir_get_version(int* version)`
 - Determine the mpir api dll version
- `mpir_get_version_string(const char** version_string);`
- ~~`mpir_version_compatibility(int *version)` – Do not need this func~~
 - ~~Determine which debugger version the dll is compatible with~~
- `mpir_initialize(const mpir_callback_t* callbacks)`
 - Debugger provides dll with a pointer to a callback table
 - Does the debugger need to maintain this pointer like in mqd? Ompd does not require it - John: doesn't matter, it is a static table in TV (Anh: same as in MS debugger)

mpir_callbacks_t

- typedef struct mpir_callbacks_t

```
{
```

```
    mpir_alloc_ft          mpir_alloc_fp;  
    mpir_free_ft           mpir_free_fp;  
    mpir_read_memory_ft    mpir_read_memory_fp;  
    mpir_write_memory_ft   mpir_write_memory_fp;  
    mpir_find_symbol_ft    mpir_find_symbol_fp;  
    mpir_get_type_sizes_ft mpir_get_type_sizes_fp;  
    mpir_target_to_host_ft  mpir_target_to_host_fp;  
    mpir_host_to_target_ft  mpir_host_to_target_fp;
```

```
}
```

DLL's provided functions

- `mpir_get_debug_state(mpir_debug_state_t* debug_state)`
MPIR_DEBUG_NULL
MPIR_DEBUG_SPAWNED
MPIR_DEBUG_ABORTED
- `mpir_get_world_size(int* size)`
 - Need different name because we want to think about supporting dynamic processes