

One of the difficulties encountered as an implementer of MPI2 One-Sided was due to the possibility that multiple different synchronization methods might be used on a given window. If the implementer knew that a window would only be used with one synchronization type, there could be optimizations made. Propose the following change to window creation:

`MPI_WIN_CREATE(base, size, disp_unit, info, assert, comm, win)`

'assert' may contain only one (or none) of the following values:

<code>MPI_MODE_FENCEONLY</code>	The window will only use <code>MPI_Win_fence</code> synchronization
<code>MPI_MODE_PSCWONLY</code>	The window will only use <code>MPI_Win_post/start/complete/wait</code> synchronization
<code>MPI_MODE_LOCKONLY</code>	The window will only use <code>MPI_Win_lock/unlock</code> synchronization

(the above is MPI2-centric and probably needs to change for MPI3 proposal?)

The implementer could insert a function table in the window object (e.g. in `MPID_DEV_WIN_DECL`) and populate that function table based on asserts given in the `MPI_WIN_CREATE` call. This proposal is that these be asserts rather than values in `MPI_Info` because their nature is more that of an assert, although the 'info' object could be used if need be. The main idea is to make these official definitions, even if implementations are free to ignore them, to promote use in applications.