# Example: Upgrading Wasm actor

In this example, we show how you can write an upgrade migration which upgrades the code of an existing WASM actor that has been deployed on chain.

More specifically, in this example we want to replace the code of thechainmetadata actor which was deployed at genesis. This actor is used to store blockhashes of the previous blocks on chain. For instance, suppose we intend to enhance this actor to store additional information beyond block hashes. The specifics of the new version's functionality are irrelevant; we focus solely on replacing the actor's code.

Inside the migration function, we first must have access to the WASM binary of the new actor. Here, we simply copied the source code of thechainmetadata actor to a new location, made relevant changes to the source code of that new actor and compiled it to thefendermint_actor_chainmetadata_v2.wasm target.

To replace the existingchainmetadata actor that we deployed at genesis with this new v2 version, we need store the new WASM code in the blockstore, then updatecode of the actor state associated with thechainmetadata actor withcode_cid of the new WASM actor.

Our migration function is defined as follows:

```

Copy // The WASM binary of the new version of the chainmetadata actor. staticWASM_BIN:&[u8]=include_bytes!("../output/fendermint_actor_chainmetadata_v2.wasm");

pubfnupgrade_wasm_actor_func( state:&mutFvmExecState, )->anyhow::Result<()> { letstate_tree=state.state_tree_mut();

// get the ActorState from the state tree // letactor_state=matchstate_tree.get_actor(CHAINMETADATA_ACTOR_ID)?{ Some(actor)=>actor, None=>{ returnErr(anyhow!("chainmetadata actor not found")); } }; println!( "chainmetadata code_cid: {:?}, state_cid: {:?}", actor_state.code, actor_state.state );

// store the new wasm code in the blockstore and get the new code cid // letnew_code_cid=state_tree.store().put( Code::Blake2b256, &Block{ codec:IPLD_RAW, data:WASM_BIN, }, )?; println!("new chainmetadata code_cid: {:?}", new_code_cid);

// next we update the actor state in the state tree // state_tree.set_actor( CHAINMETADATA_ACTOR_ID, ActorState{ code:new_code_cid, state:actor_state.state, sequence:actor_state.sequence, balance:actor_state.balance, delegated_address:actor_state.delegated_address, }, );

Ok(()) }

```

Once we have finished writing ourUpgrade migration, we can add it to theUpgradeScheduler :

```

Copy letmutscheduler=UpgradeScheduler::new(); letupgrade=Upgrade::new(chain_name, block_height, app_version, upgrade_wasm_actor_func); scheduler.add(upgrade);

// when initializing the FvmMessageInterpreter, specify the upgrade schedule letinterpreter=FvmMessageInterpreter:::new( ... scheduler, );

```

Previous Example: Patching actor state Next Deploying an explorer Last updated9 days ago On this page Was this helpful? Edit on GitHub