

---

Posted by [shalamyansky](#) on Mon, 15 May 2023 14:19:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

UdrCppEngine.h

```
FB_BOOLEAN* FB_UDR_PLUGIN_ENTRY_POINT(::Firebird::IStatus* status, FB_BOOLEAN*
theirUnloadFlag, ::Firebird::IUdrPlugin* udrPlugin);
```

UdrCppEngine.h

```
extern "C" FB_DLL_EXPORT FB_BOOLEAN*
FB_UDR_PLUGIN_ENTRY_POINT(::Firebird::IStatus* status, \
FB_BOOLEAN* theirUnloadFlag, ::Firebird::IUdrPlugin* udrPlugin) \
{\
::Firebird::Udr::FactoryRegistration::finish(status, udrPlugin); \
\
class UnloadDetector \
{\
public: \
UnloadDetector(FB_BOOLEAN* aTheirUnloadFlag, ::Firebird::IUdrPlugin* aUdrPlugin) \
: myUnloadFlag(FB_FALSE), \
theirUnloadFlag(aTheirUnloadFlag), \
udrPlugin(aUdrPlugin) \
{\
}\
\
~UnloadDetector() \
{\
```

```

    if (!myUnloadFlag) \
        *theirUnloadFlag = FB_TRUE; \
    } \
    \
    FB_BOOLEAN myUnloadFlag; \
    FB_BOOLEAN* theirUnloadFlag; \
    ::Firebird::IUdrPlugin* udrPlugin; \
}; \
\
static UnloadDetector unloadDetector(theirUnloadFlag, udrPlugin); \
\
return &unloadDetector.myUnloadFlag; \
}

```

UdrEngine.cpp

```

UdrPluginImpl* Engine::loadModule(ThrowStatusWrapper* status, IRoutineMetadata* metadata,
PathName* moduleName, string* entryPoint)
{
...
    UdrPluginImpl* udrPlugin = FB_NEW UdrPluginImpl(*moduleName, module);
    udrPlugin->theirUnloadFlag = entryPoint(status, &udrPlugin->myUnloadFlag, udrPlugin);
...
}

class UdrPluginImpl : public Versionedface<IUdrPluginImpl<UdrPluginImpl,
ThrowStatusWrapper> >

```

```

{
public:
  UdrPluginImpl(const PathName& aModuleName, ModuleLoader::Module* aModule)
    : moduleName(*getDefaultMemoryPool(), aModuleName),
      module(aModule),
      myUnloadFlag(FB_FALSE),
      theirUnloadFlag(NULL),
      functionsMap(*getDefaultMemoryPool()),
      proceduresMap(*getDefaultMemoryPool()),
      triggersMap(*getDefaultMemoryPool())
  {
  }

  ~UdrPluginImpl()
  {
  if (myUnloadFlag)
    return;

  *theirUnloadFlag = FB_TRUE;

  {
  GenericMap<Pair<Left<string, IUdrFunctionFactory*> > >::Accessor accessor(&functionsMap);
  for (bool cont = accessor.getFirst(); cont; cont = accessor.getNext())
    accessor.current()->second->dispose();
  }

  {
  GenericMap<Pair<Left<string, IUdrProcedureFactory*> > >::Accessor
accessor(&proceduresMap);
  for (bool cont = accessor.getFirst(); cont; cont = accessor.getNext())
    accessor.current()->second->dispose();
  }

  {
  GenericMap<Pair<Left<string, IUdrTriggerFactory*> > >::Accessor accessor(&triggersMap);
  for (bool cont = accessor.getFirst(); cont; cont = accessor.getNext())
    accessor.current()->second->dispose();
  }
}

FB_BOOLEAN myUnloadFlag;
FB_BOOLEAN* theirUnloadFlag;
};

```

