

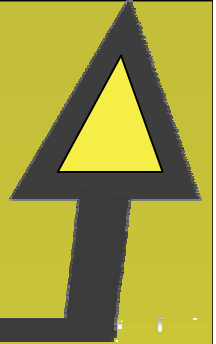


***Minimum CORBA
Tradeoffs***

Ken Black
Founder and President
Highlander Engineering

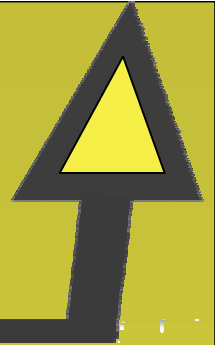
HIGHLANDER™

minimumCORBA



- Subset designed for static systems with limited resources
 - Reduces memory requirement 26-35% with Highlander VisiBroker-RT
- Retains all features required by Real-Time CORBA
- Adopted 1998
- Incorporated into CORBA 2.4

Interoperability and Portability



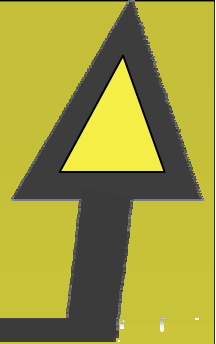
- Only impacts interfaces available to applications using minimumCORBA ORB
- Use is transparent to remote ORBs
 - 100% interoperable with full CORBA (GIOP, IIOP)
- MinimumCORBA applications can be run unchanged on a full CORBA ORB
- Supports multiple/child POAs with subset of policies



***minimumCORBA
Differences***

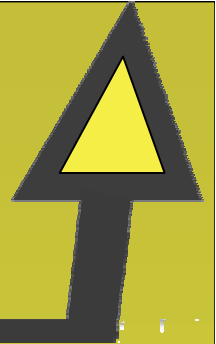
HIGHLANDER™

Object Access and Implementation



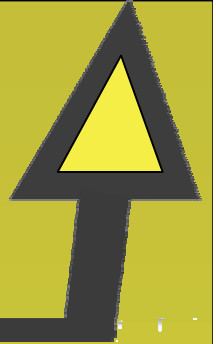
- Full CORBA has two principal mechanisms
 - Through stubs and skeletons generated by IDL compiler
 - Dynamically at run-time
 - Interface Repository (IR) maintains database of types from IDL
 - Clients use Dynamic Invocation Interface (DII) to construct requests
 - Servants use Dynamic Skeleton Interface (DSI)
- minimumCORBA omits dynamic interfaces (IR, DII, DSI)
 - IDL must be known at compile time

Object Identification and Activation



- Full CORBA provides three mechanisms for finding the servant for an object
 - Active Object Map — ORB-maintained table
 - Servant Manager — User-provided mechanism
 - Default Servant — Handles all requests for a POA
- Servant Manager and Default Servant allow objects to be created when used
- minimumCORBA only has Active Object Map
 - All objects must be activated before use
 - Per-object overhead

Other Features Omitted from minimumCORBA

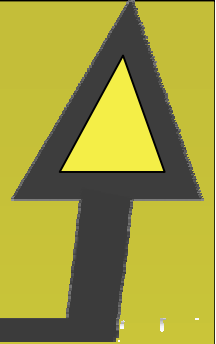


- Implicit activation
- Interceptors
- Value Types (Objects by Value)



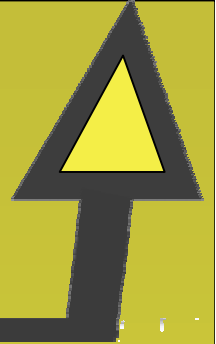
HIGHLANDER™

minimumCORBA Benefits Applications that...



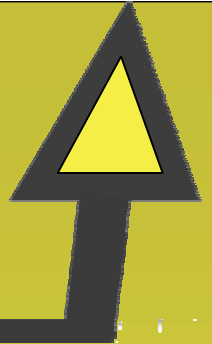
- Know at compile time the interfaces (IDL) to all objects implemented and used
- Have few objects
 - Since each object has overhead whether actively used or not
- Do not need interceptors (e.g., for security) or Value Types

Full CORBA Benefits Applications that...



- Do not know the interfaces (IDL) to all objects at compile time
- Have many objects
 - Since objects can be activated as needed
- Need interceptors or value types
- Want to leverage software originally written for enterprise systems

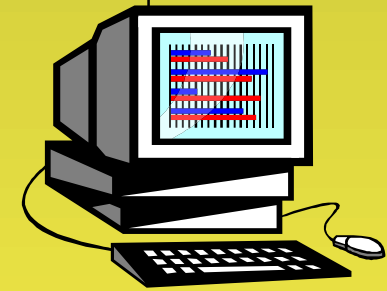
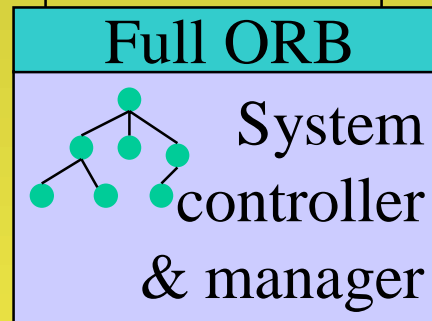
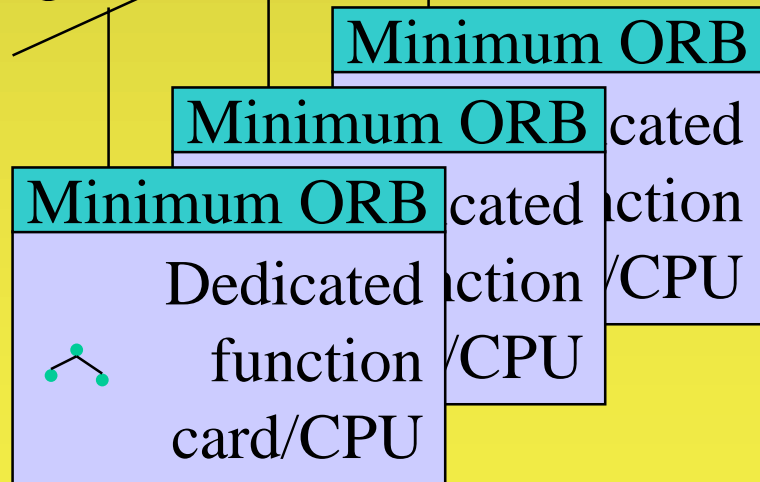
Full and minimum CORBA Co-existence



Embedded Application

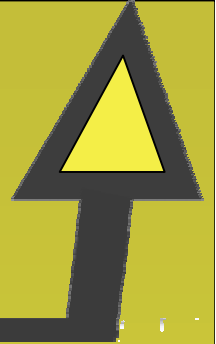
IIOP / Ethernet

xIOP
GIOP over internal bus



- Uses IR, DII, DSI to discover and proxy interfaces on function cards
- Many objects: all on function cards plus system-level

Summary



- minimumCORBA reduces an ORB's static memory requirement
- Most embedded applications can take advantage of minimumCORBA
- Some embedded applications will benefit from full CORBA
 - Reduced dynamic memory use
 - Need omitted features
- Full and minimumCORBA can co-exist