

Object XRef Architecture and SDK in Max 8

Attila Szabo
Software Developer
Autodesk Media & Entertainment Division

Agenda

- 1. What are XRefs?**
- 2. Object XRef Architecture**
- 3. Object XRef SDK**
- 4. QA**

What are XRefs?

Definition

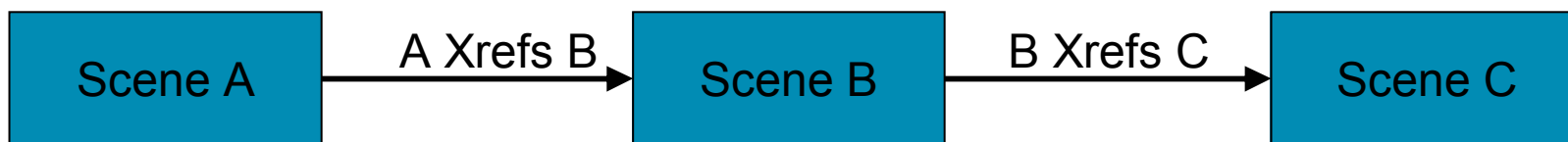
- eXternal References = “XRefs”
- XRefs = data from an external scene + live-link to the external scene

Terminology

- Source Data: the xrefed data
- Master Scene: the host of the xrefed data

XRef design philosophy in 3ds Max:

- Master saves moniker\placeholder to source
- Source is **not** saved with Master
- Source cannot be edited in Master
- Master can be updated with latest version of Source
- XRefs can be nested



What are Object XRefs?

Definition:

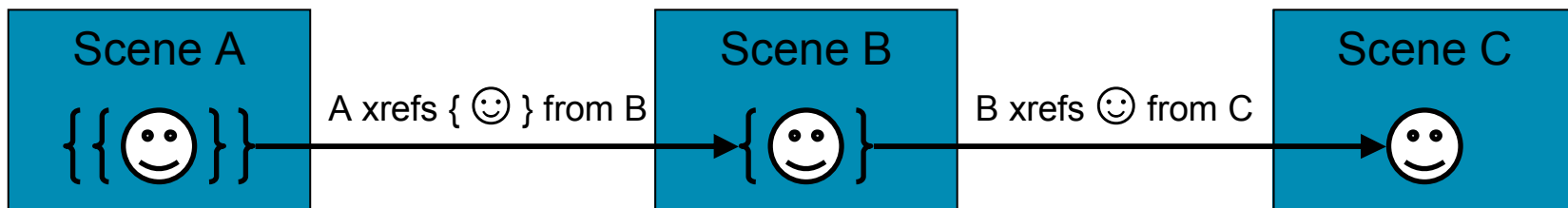
- Object XRefs = XRefs to individual scene entities

Terminology:

- Source Item: the xrefed scene entity
- Master Scene: the host of the xrefed scene entity
- XRef Item: placeholder of the Source Item within the Master Scene

Object XRef design philosophy in 3ds Max:

- XRefed scene entities can be enhanced within the master Scene. Enhancements are saved in the Master Scene
- Xrefing maintains relationships between xrefed scene entities



Object XRef Architecture

- 1. Components of the architecture**
- 2. Nested XRefs Items**
- 3. Types of XRef Items**
- 4. XRef Items in the Scene**
- 5. Strengths and Limitations of the architecture**

Object XRef Architecture (1)

Object XRef Manager

- Creates, destroys, modifies xref items and records
- Owns XRef Records

XRef Record

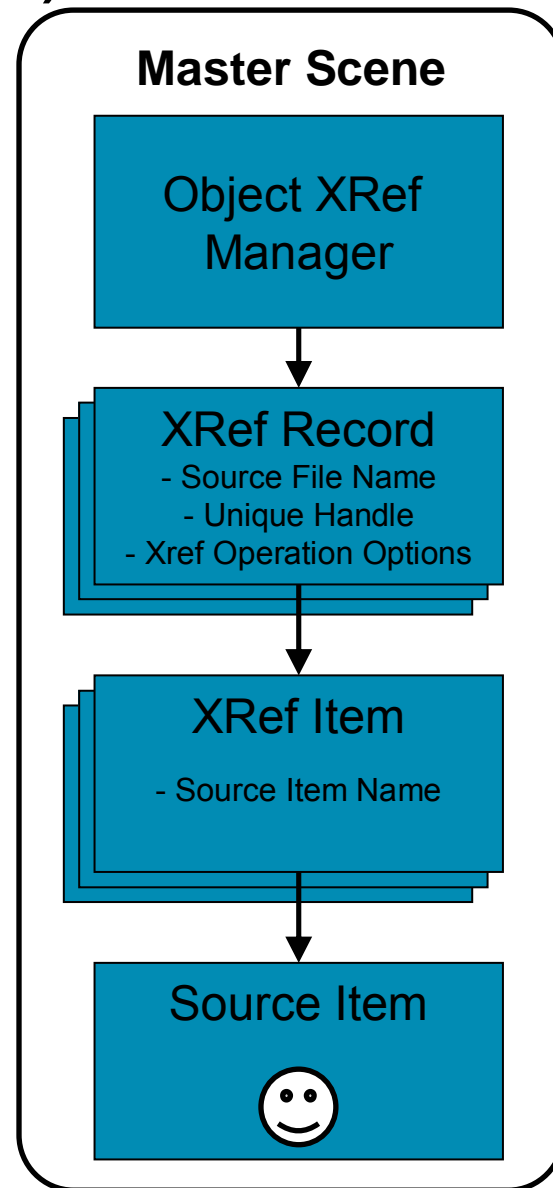
- Record of an xref operation
- Owns the XRef Items created by an xref operation
- Saved with Master Scene

XRef Item

- Moniker\placeholder of a source item
- Wrapper for source item
 - Allow read-only access to source item
 - Act as pass-through for requests they cannot satisfy
- Saved with Master Scene

Source Item

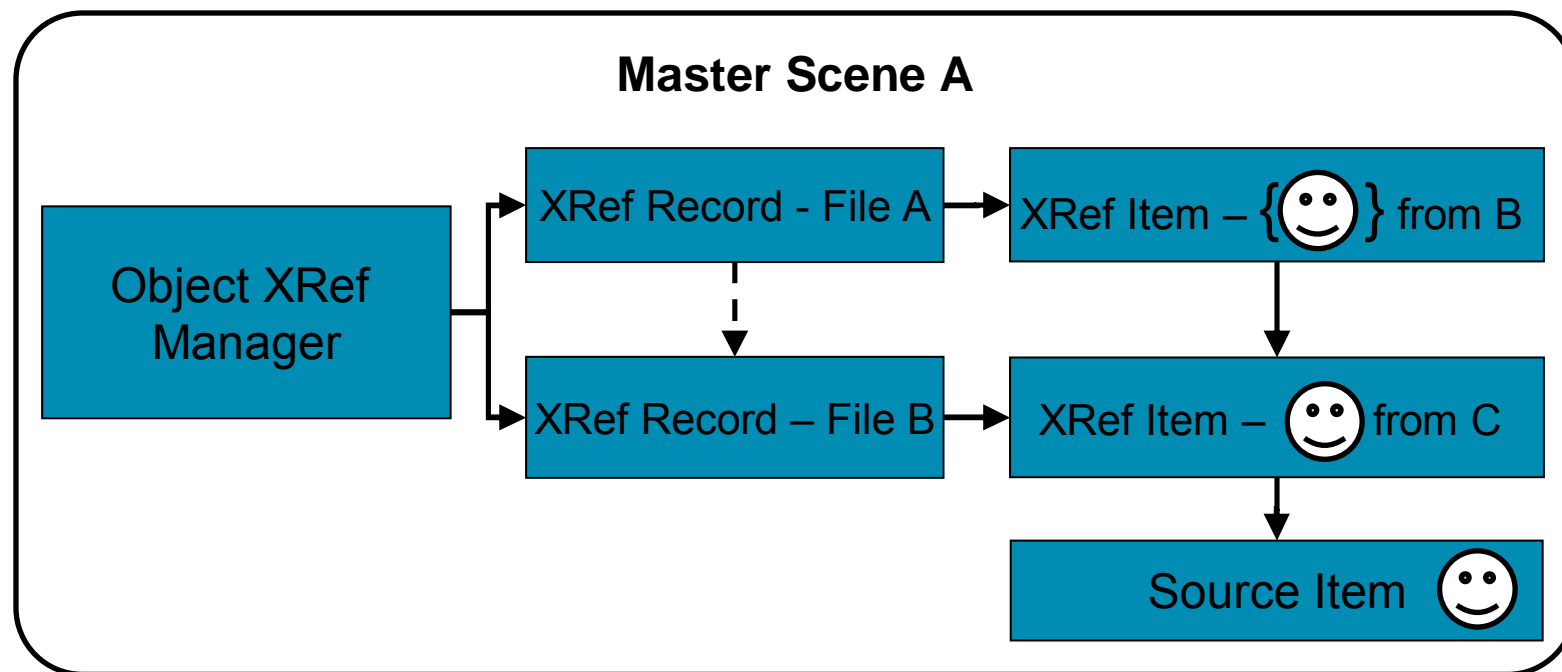
- A copy of the scene entity from the source file
- **Not** saved with Master Scene



Object XRef Architecture (2)

Nested XRef Items

- The source item of an xref item is another xref item
- Nesting of xref items causes nesting of xref records



Object XRef Architecture (3)

Types of XRef Items

▪ XRef Object

- Xrefs the geometry pipeline of a node
- Supports proxy xref item

▪ XRef Atmospherics

- Xrefs an atmospheric effect assigned to a gizmo object
- “Subordinated” to XRef Objects
 - Cannot be xrefed directly; xrefed when its gizmos are xrefed
 - Automatically deleted when all its xrefed gizmos are deleted from master scene
 - Cannot be assigned to other gizmos in the master scene

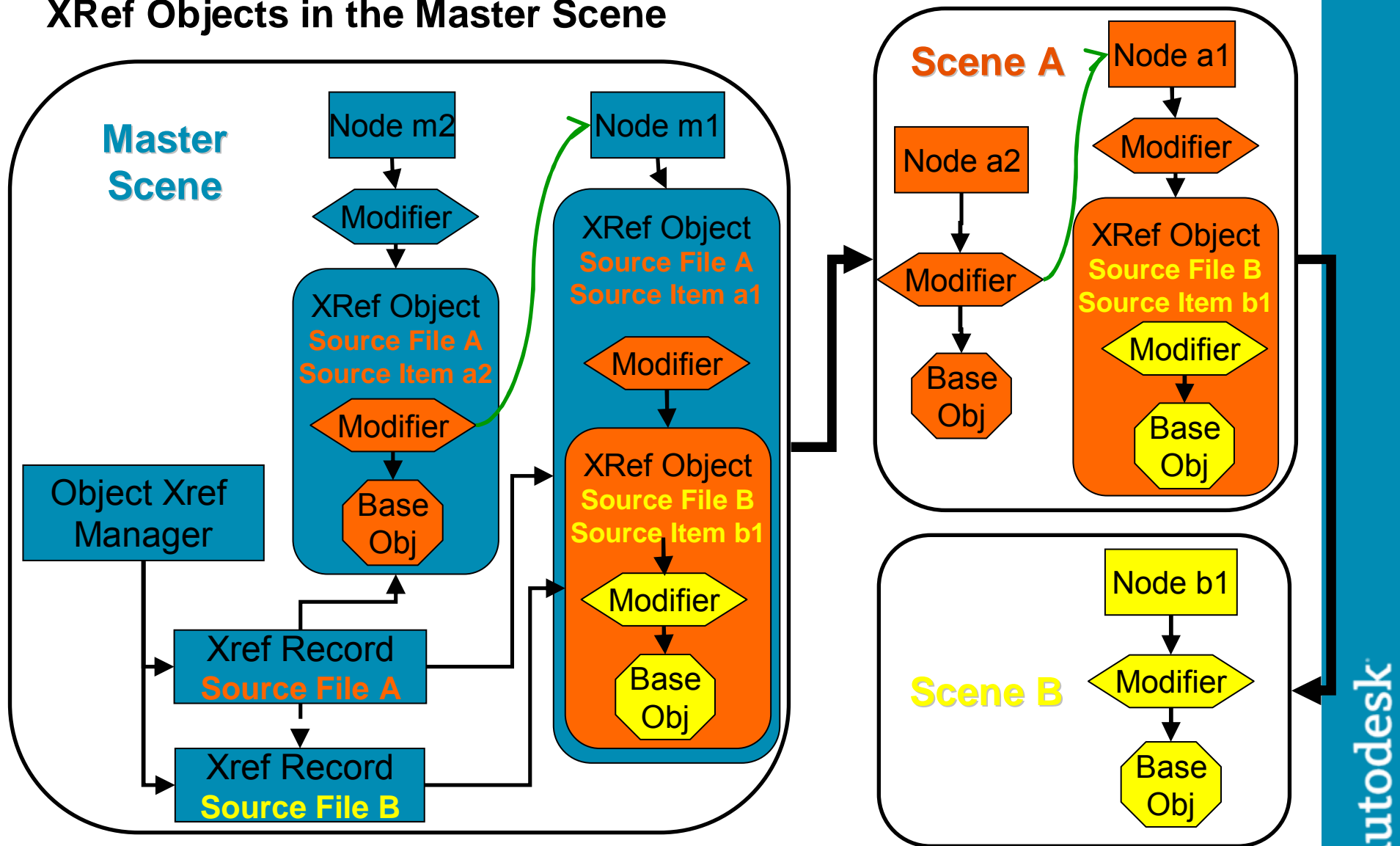
▪ XRef Material ✦

- Xrefs the material and all of its sub-materials assigned to a node
- Can be used as a sub-material in a master scene material

Note: many scene entities related to the xrefed ones are merged rather than xrefed: nodes, controllers, textures, layers, etc ...

Object XRef Architecture (4.1)

XRef Objects in the Master Scene



Object XRef Architecture (5)

Strengths

- **Excels in asset aggregation workflows**
- **Extensible – allows for supporting new xref item types**
- **Prevents data loss by allowing read-only access to source items**
- **Handle based design increases robustness**

Limitations

- **Does not support animation aggregation workflows (yet!)**
- **No per xref item updates**
- **Include All not honoured with nested xref records**
- **Incomplete undo\redo support**
- **No “in-place” editing of source items**

Object XRef SDK

- 1. Highlights of the Max 8 Object XRef SDK**
- 2. Object XRef Classes**
- 3. How to make your plugin object xref-able?**
- 4. How to create xref items?**
- 5. How to modify xref items?**
- 6. How to delete xref items?**
- 7. How to work with xref records?**

Object XRef SDK (1)

Highlights of the Max 8 Object XRef SDK

- **New set of classes with rich behaviour**
 - IXRefItem, IObjXRefRecord, IObjXRefManager8, etc
- **New design paradigm**
 - Xref items and records expose read-only interfaces
- **Performance and functionality enhancements**
 - Xref record update performance increased
 - Xref objects node name dependency fixed
- **Full exposure of object xref system to maxscript**
 - objXrefMgr core interface, IXRefRecord, IXRefItem, etc
- **Backward compatibility with legacy\deprecated xref classes**
 - IXRefObject::SetUpdateMats, IXRefObject::SetIgnoreAnim – no-ops

Object XRef SDK (3)

How to make your plugin object xref-able?

- **Nothing special to do!**
 - All object, material and atmospheric plugins are automatically supported by Max's object xref system
- **General guidelines for all plugins**
 - Program against interfaces rather than class\super class ids
 - `IXRefObject8* IXRefObject8::GetInterface(InterfaceServer& is)`
 - Prefer re-using available algorithms rather than coding your own
 - `int EnumGeomPipeline(GeomPipelineEnumProc *gpep,)`
 - Avoid dangling reference pointers
 - Initialize them to NULL or a valid reference target address

Object XRef SDK (4.1)

How to create xref items?

a) Add new xref items to a new xref record

```
virtual IObjXRefRecord* IObjXRefManager8::AddXRefItemsFromFile(  
    const TCHAR* srcFileName,  
    bool promptObjNames,  
    const Tab<TCHAR*>* objNames,  
    unsigned int xrefOptions);
```

- Xrefs the objects specified via `objNames` and their materials, unless `XREF_MERGE_MATERIALS` is specified via `xrefOptions`
- Allows for xrefing all objects from the source file: `promptObjNames=false`, `objNames=NULL`
- Prevents circular xrefs from being created
- Maintains node hierarchies
- Objects dependent on xrefed ones are automatically xrefed. Exception: param-wire dependencies.
- Atmospherics are xrefed by xrefing their gizmos

Object XRef SDK (4.2)

How to create xref items?

b) Add new xref items to an existent xref record

```
virtual IObjXRefRecord* IObjXRefManager8::AddXRefItemsToXRefRecord(
    IObjXRefRecord& xrefRecord,
    bool promptObjNames,
    const Tab<TCHAR*>* objNames);
```

- The xref options recorded with the xref record are used instead

Xref Options:

<code>XREF_AS_PROXY</code>	- Xref objects are created as proxies
<code>XREF_XREF_MODIFIERS</code>	- Modifiers are xrefed; i.e. won't be accessible in master file
<code>XREF_DROP_MODIFIERS</code>	- Modifiers are neither xrefed nor merged
<code>XREF_MERGE_MODIFIERS</code>	- Modifiers are merged, i.e will be accessible in the master file
<code>XREF_MERGE_MANIPULATORS</code>	- Manipulators are merged rather than xrefed
<code>XREF_SELECT_NODES</code>	- The nodes of xrefed items are selected
<code>XREF_MERGE_MATERIALS</code>	- Materials are merged rather than xrefed

Object XRef SDK (4.3)

How to create xref items?

- Handling duplicated node and material names

```
virtual void IObjXRefManager8::SetDupObjNameAction(unsigned int action);  
virtual void IObjXRefManager8::SetDupMtlNameAction(unsigned int action);
```

- Duplicated node name actions

MERGE_DUPS_PROMPT - Prompt user when duplicate node names are encountered
MERGE_DUPS_MERGE - Merge nodes and keep old ones in the scene
MERGE_DUPS_SKIP - Do not merge nodes with duplicated names
MERGE_DUPS_DELOLD - Merge nodes and delete old ones from the scene
MERGE_LIST_NAMES - Not supported
MERGE_DUPS_RENAME - Merge nodes and automatically rename them

- Duplicated material name actions

MERGE_DUP_MTL_PROMPT - Prompt user when duplicate material names are found
MERGE_DUP_MTL_USE_MERGED - Merge material and replace all scene materials with the same name with the merged material
MERGE_DUP_MTL_USE_SCENE - Don't merge material, use scene material with same name
MERGE_DUP_MTL_RENAME - Merge material and automatically rename them

Object XRef SDK (5)

How to modify xref items?

- By modifying the xref item's source item name
- By modifying the xref item's source file name (re-pathing xref item)
- By modifying the xref record's source file name (re-pathing xref record)

```
virtual bool IObjXRefManager8::SetXRefItemSrcName(
    ReferenceTarget& xrefItem,
    const TCHAR* srcItemName);

virtual bool IObjXRefManager8::SetXRefItemSrcFile(
    ReferenceTarget& xrefItem,
    const TCHAR* srcFileName);

virtual bool IObjXRefManager8::SetRecordSrcFile(
    IObjXRefRecord& xrefRecord,
    const TCHAR* srcFileName);
```

- These methods can be called in any order
- Calling these methods will trigger an update of the whole xref record
- Xref atmospherics cannot be modified; they depend on their gizmos

Object XRef SDK (6.1)

How to delete xref items?

a) Delete nodes from the scene

- Xref materials referenced by the material editor are not deleted

b) Delete them from their xref records

```
virtual bool IObjXRefManager8::RemoveXRefItemFromScene(  
    Tab<ReferenceTarget*>& xrefItems);
```

- Xref objects - all nodes that reference the xref object are removed
- Xref atmospherics - are removed automatically when all their xrefed gizmos have been removed
- Xref materials – are merged into the scene rather than removed

Object XRef SDK (6.2)

How to delete xref items?

c) Merge xref items into the scene

```
virtual bool IObjXRefManager8::MergeXRefItemIntoScene(  
    Tab<ReferenceTarget*>& xrefItems);
```

- Source item becomes part of master scene
- Wrapper xref item is removed from reference hierarchy; all references to the xref item are transferred to the source item
- Xref objects - all the nodes that reference the xref stay in the scene
- Xref atmospherics - are merged automatically when all their xrefed gizmos have been merged
- Xref materials – the source material continues to be assigned to the same nodes that had the xref material assigned to them

Object XRef SDK (7.1)

How to update xref records?

```
virtual bool IObjXRefRecord::Update( );
```

- Updates the record and all its child\nested records
- Reloads the record's source file even if it hasn't changed

```
virtual bool IObjXRefManager8::UpdateAllRecords( );
```

- Only records that are out-of-date are updated.

How to bring new nested scene entities into a master scene?

- Include All does not work recursively!!!
- All intermediate master files must be update first and saved, starting with the most nested one

Object XRef SDK (7.2)

Xref record handles

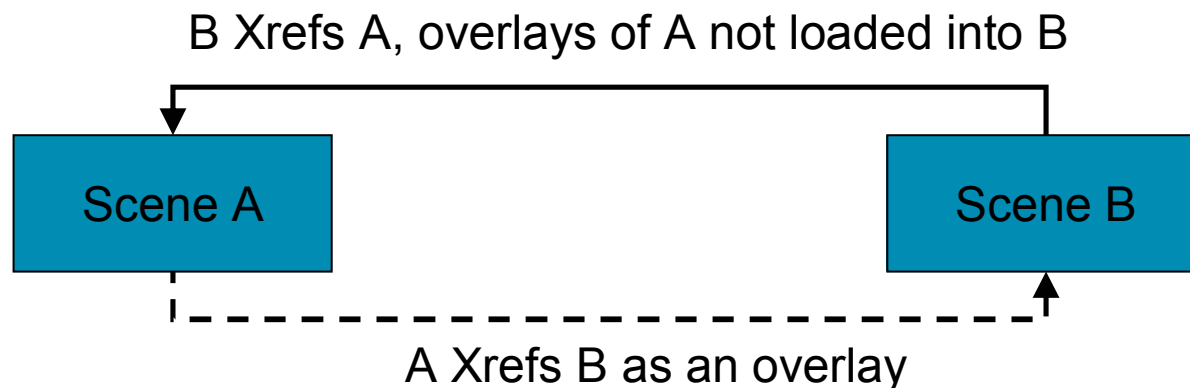
- Xref record pointers could become invalid. Don't cache them. Use Xref record handles instead
- Xref record handles:
 - Can be cached temporarily and used for retrieving xref records in a safe way
 - Are guaranteed to be unique for all xref records within the current scene
 - Are not persisted with the scene; are not unique across scenes

```
typedef unsigned long XRefRecordHandle;  
const XRefRecordHandle kNullXRefRecordHandle = 0;  
virtual IObjXRefRecord* IObjXRefManager8::FindRecord(  
XRefRecordHandle xrefRecHandle) const;
```

Scene XRef bonus topic

Overlay Scene XRefs

- Allows for two or more scenes to scene xref each other without creating circular xrefs



```
#define XREF_SCENE_OVERLAY (1<<12)
virtual void INode::SetXRefFlags(int i, DWORD flag, BOOL onOff);
```

Q&A

End of Presentation Slides

<http://sparks.discreet.com>

See you there...