

The Bungie logo is centered on a black background. It features the word "BUNGIE" in a white, sans-serif font. A grey arc curves over the letters "N" and "G". The letter "i" is lowercase and has a blue dot. A small trademark symbol (TM) is located to the upper right of the letter "E".

BUNGiE™

# Ten Years of Keeping People Working [At Bungie]

Mat Noguchi  
Bungie Studios





- Halo: Combat Evolved
- Halo 2
- Halo 3
- Halo 3: ODST
- Halo: Reach

# Why so angry?

- **It's my job**
  - (I'm not actually that angry. Maybe stressed.)
- **Halo is big and awesome**
- **Halo is literally big and awesome**
  - Halo: Reach == 30 Halo: CEs!
  - Still has to fit on a DVD
- **And we never actually planned for this**



# Well known bottlenecks

- **Disk/storage footprint**
  - DVD or BluRay or downloadable package
- **Memory**
  - 512 MB for current generation
- **I/O**
  - 1 or 2 devices, limited bandwidth
  - Internet (much more limited bandwidth)



# It's also about the people!

- Someone has to make all that content
- Someone has to program the game to use that content
- And everyone has to make it awesome



# It starts with programmers

- **Programmers should only work with content in a simple way**
  - Not as raw memory
  - Not as files
  - Not as I/O
- **Otherwise, they will mess it up.**
  - Even me. (DOH!)



# content == tags

- **Programmers think in terms of tags**
  - Containers, not memory
  - Dependencies, not files
  - Don't think about I/O at all!





# Tag system: Halo 1

## Features

- **Tag build (development)**
  - Hot loading for all tags
- **Cache build (ship)**
  - Optimized memory layout
  - Monolithic file

## API

- `long tag_load(  
    const char *name,  
    tag group_tag,  
    dword flags);`
- `void *tag_get(  
    long tag_index);`



# Tag system: Halo 2

## More Features

- **Tag build**
  - Append only versioning
- **Cache build**
  - Automatic cross-map sharing
  - Automatic immutable data sharing

## API

- `long tag_load(  
const char *name,  
tag group_tag,  
dword flags);`
- `void *tag_get(  
long tag_index);`



# Tag system: Halo 3

## Even More Features

- **Tag build**
  - Automatic and explicit tag versioning
  - Arbitrary paging support
  - Monolithic files on devkit
- **Cache build**
  - Whole disk optimizations
  - Can run without HDD

## API

- `long tag_load(  
    const char *name,  
    tag group_tag,  
    dword flags);`
- `void *tag_get(  
    long tag_index);`



# More technical details

- [The Technology of Halo 2 \(2004\) - Chris Butcher](#)
- [Content Management for Halo 2 and Beyond \(2005\) - Mat Noguchi](#)
- [New Dog, Old Tricks: Running Halo 3 without a Hard Drive \(2008\) - Mat Noguchi](#)



# It's simple!

- Every in-game feature is driven by tags!
- Encourages communication!



# PORKCHOP SANDWICHES!



# Shaders tags, my nemesis

- **19 shader types**
  - The HLSL is tag driven (woo 😊)
- **310,718,936 permutations**
  - Only use 1,659 (boo 😞)
- **Can generate on-demand**
  - Usually several seconds to a minute
  - But we can't cache them on the devkit
  - More than 10 and artists get mad. Really mad.
- **Build shaders periodically on our build farm**
  - [Life on the Bungie Farm ... - Luis Villegas and Sean Shypula](#)
- **Let's not have to do that next time.**



# Personal growth

- From 2002-ish to 2007
  - Level editor
  - Tag system
  - Tag editor
  - Audio





But then...



# And now...

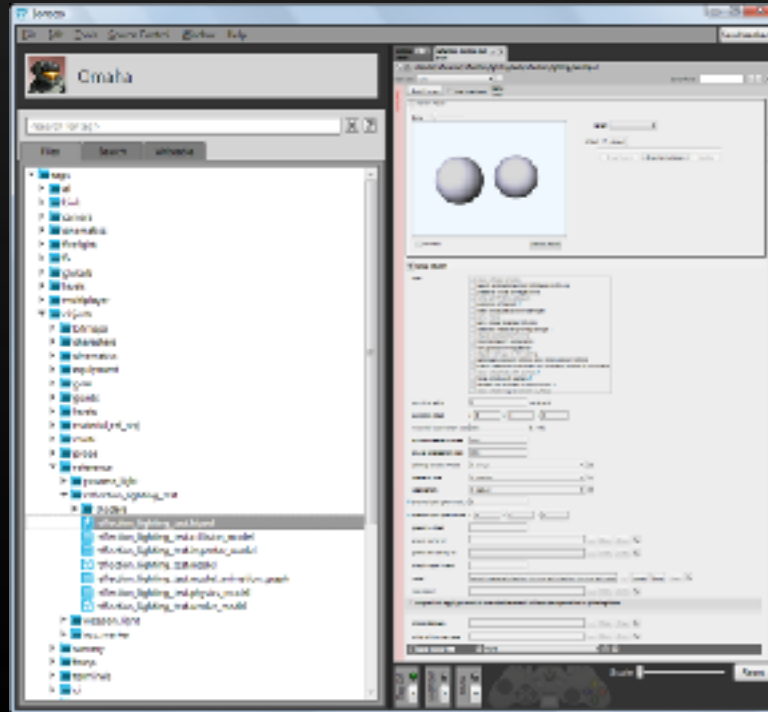


# Today

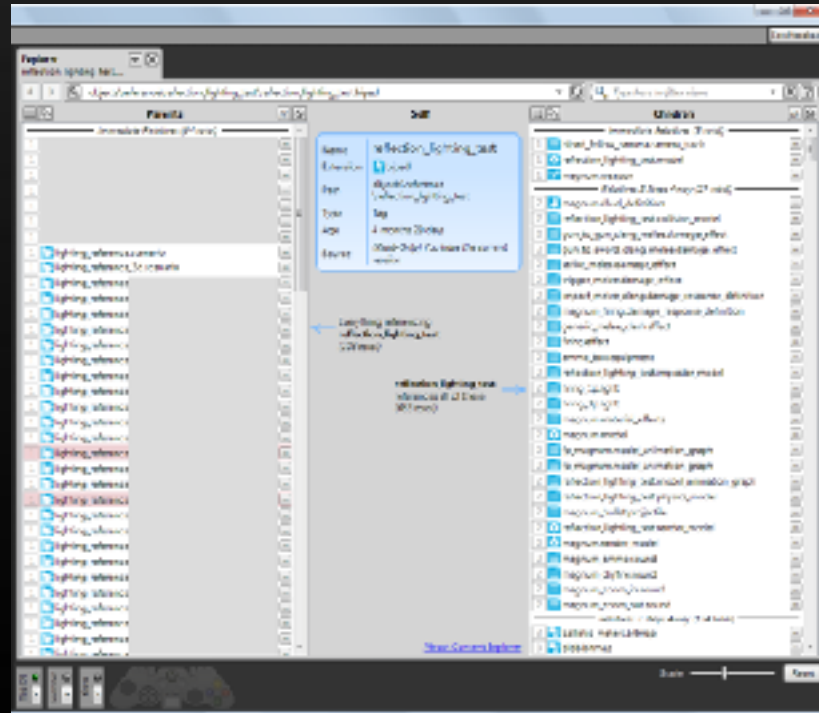
- **I own**
  - Tag system
- **Other people own**
  - Level editor
  - Audio
- **Tools team**
  - Tag editor



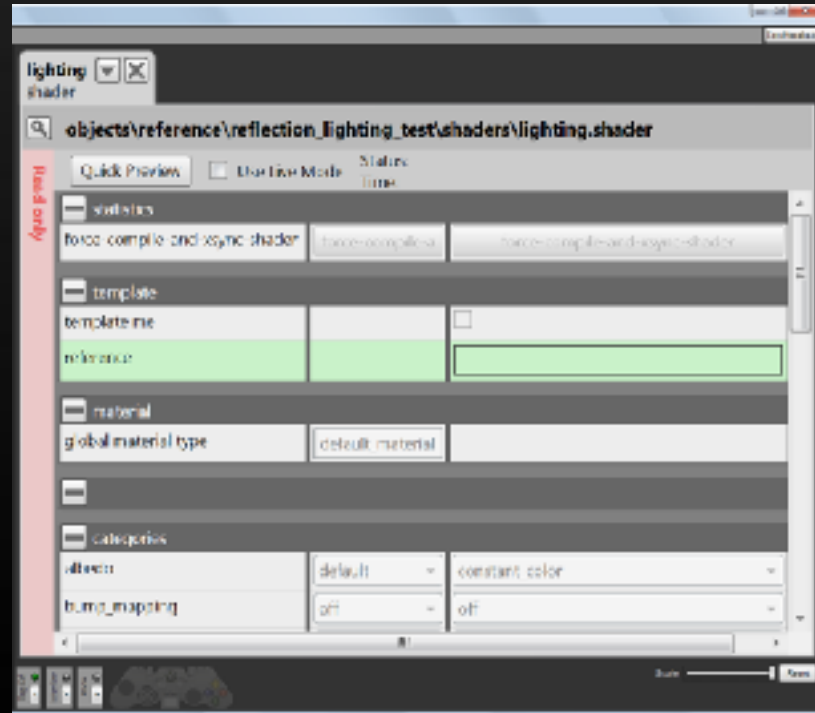
# Tag editor evolved: Bonobo



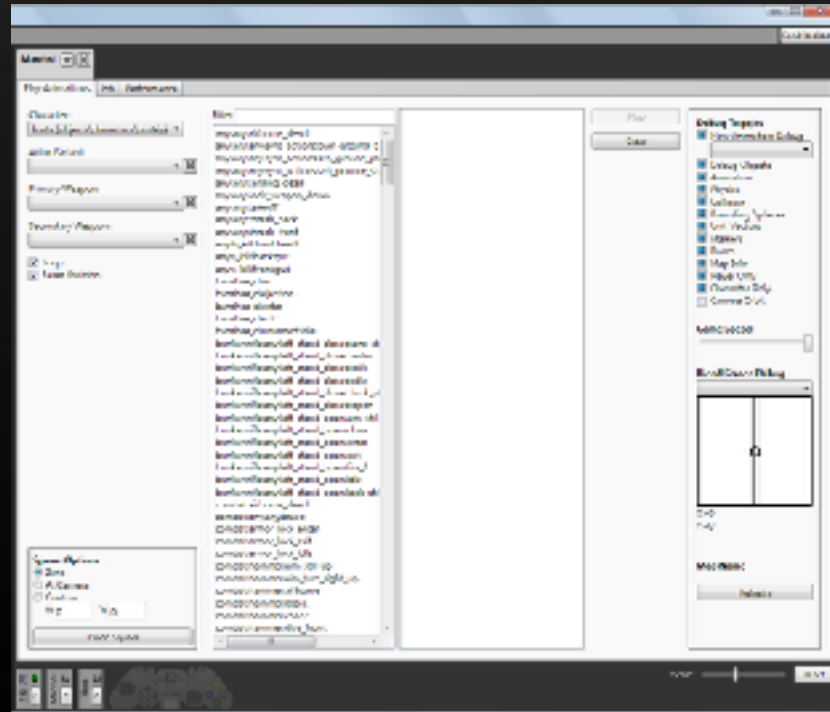
# Content explorer



# Quick preview



# Mantini



# Why this matters

- **It's not just about the code**
  - all content code has the same structure





# Content code structure

## Boost

```
template<class Archive>
void serialize(
    Archive & ar,
    gps_position & g,
    const unsigned int version)
{
    ar & g.degrees;
    ar & g.minutes;
    ar & g.seconds;
}
```

## Tag system (pseudocode)

```
TAG_GROUP(
    gps_position)
{
    {_field_long_integer,
    "degrees"},
    {_field_long_integer,
    "minutes"},
    {_field_real, "seconds"},
    {_field_terminator},
};
```



# Why this matters

- **It's not just about the code**
  - All content code has the same structure
  - More about what you can't do



# Why this matters

- **It's not just about the abstractions**
  - It's important to *have* them
  - Ideal if they map to a decent implementation
    - Even better if you don't have to ship them



# Why this matters

- It's about the culture Bungie built around the tag system
  - “You forgot the part where you need to stand behind the rest of the team with a baseball bat.” - Andy Firth
  - The bat- If it's not a tag, it's not in the game
  - But that's okay, because it's simple to use!



Per audacia ad astra!

**THANK YOU**



The Bungie logo is centered on a black background. It features the word "BUNGIE" in a white, sans-serif font. A grey arc curves over the letters "N" and "G". The letter "i" is lowercase and has a blue dot. A small trademark symbol (TM) is located to the upper right of the letter "E".

BUNGiE™