

Parallax OBEX Gold Standard



February 15, 2011

Talk on the forums

- Should they even be called “objects”?
 - They are technically classes? Objects need be instantiated.
 - Suggested “soft peripherals” or “virtual peripherals”
- “Raise the bar” for object submission
 - Define a standard?
- Peer review?
- Source-forge like structure?
- Parallax support of Gold Objects



PARALLAX 3
www.parallax.com

High Standards...

- Anyone can submit an object for Gold certification.
- Must fit a standard that we (Parallax) define.
- Object must be fully tested, certified, and supported.
- Interoperability
 - Any combination should work together out of the box.
 - Standardized interface to objects.
- Usable across all supported languages.
- Commission customers to bring their object up to standards.
- Stifle object submission?



PARALLAX 3
www.parallax.com

Parallax Semiconductor

- Professionals aren't going to want to wade through poor code
- Propose making ONLY gold objects available on Parallax Semi website
- Keep a clean, well indexed library
- AppNotes to support gold objects?



PARALLAX 
www.parallax.com

Support

- This is a big one!
- Gold objects need to be supported
 - Responsibility, customers will see it
 - Stand behind hardware and software
 - Imply limited FAE support for everyone?



PARALLAX
www.parallax.com

Might I suggest...?

- Partially changing structure of current OBEX to Sourceforge like structure.
 - Subversion-like controlled
 - We can implement mechanism to approve revision submissions (keep the riffraff out)
 - Customer rating of object version
 - Star rating
 - Customer comments
 - Periodic searches for gold certifiable submissions



PARALLAX 3D
www.parallax.com

See a need, fill a need

- Idea/Suggestion page
- Customers submit object ideas they want to see implemented
- Peers rate ideas
 - We can implement to gold standard
 - Customers can implement and submit for gold certification



PARALLAX 3D
www.parallax.com

Quote from Potatohead

- “Bottom line is, I want all the contributions, good, bad, well documented, code only, and everything in between. Worst case, somebody is faced with writing some code, or parsing through some that is written to learn how it works, and having done that, could very easily add that work back to the repository, same as the original contributor did. The [MIT] license permits that, so no worries. If it's needed, those that need it can organize to do it, and do the work to make it easier for others to do it too. No worries on that either.”
- I think Potatohead has the right idea – to keep users openly contributing to the open nature of the Propeller.



Gratuitous Explosions



USDOE