CosmoS

Cosmos-B

UI

*Cosmos-B is a managed code operation system designed to run on the Cosmos kit. This document describes the UI system. .*

UI

Ben KloosterMan

# 0.Overview

This document views user interface broadly as in User OS interaction though it will also cover the GUI.

Modern OS carry a lot of baggage, nowhere is this more visible than the user interface. It is also worth noting that an OS with a better and faster GUI have tended to be successful (with the exception of OS2) .

Also the increasing dependence on the internet has raised issues in how we use an OS.

Cosmos B will not have a Console or command line. This eliminates a whole class of security bugs and will force application designers to create consistent user friendly apps and configuration ( note this is one of the failings of Linux to this day). Note this does not mean all applications are GUIs there will be many services that run without a GUI but to configure or install them you will need a program or a GUI.

# 0.APPLICATIONS

Ss

Global logging.

Global configuration

How to configure

* Configure tab option
* Insert config settings in batch context

Logging global consistent via log service ( on debug builds it defaults to Synch else non critical logging is asynch)

# 0.GUI

Apple style 1 way of doing things ( not 20 like in MS) . Consistent layout.

Zooming desktop with Docking (Zooming code available using C# but needs GDI) . Docked items are docked and remain at the zoom level when docked. This will give a 3D effect but will be easier to read and simpler to use ( as well as being easier to code)

<http://arstechnica.com/software/news/2009/01/dock-and-windows-7-taskbar.ars/5>

Dock per application doesn’t work .. Doc window per app like win 7 good how to deal with folders ? Put folders and files on desktop but not shortcuts ?

# 0.Batch JOBS

Eliminates a whole class of bugs and security issues due to parameter injection . By using apps all parameters can be strongly typed. Instead of a filename the caller passes in a filename to which it already has rights etc.

No batch files or shell scripts. Python /F# /Perl/PHP can do what these can better and strongly typed.

Batch sequences are support using the ApplicationBatch context programs can add, read and remove strongly typed objects to the stream.

It is anticipitated that IDEs allow drag and drop of common functions and provide “Intellisense” and continuous compile to allow rapid and easy development of these scripts.

A large number of interfaces for standard objects

# 0.APPLICATION INSTALL

Centralized app delivery like Linux and iPhone even for commercial apps.

Basic DB .

# 0.REMOTE DESKTOP AND RECOVERY

Apple

User more connected to the internet/ Flash local is used more as storage and cache eg

Download assemblies from temp and then compile them to the cache. After compile original not needed.

No console what about remote access

* Silverlight Desktop!

In Size

# 0.APPLICATION FAILURE

Crash to desktop BAD failure and restart of service.

------------------------------------------------------------------------------