Main Page
Feature: Main page fetch summary data.
The external method: list<summary> fetchSummaryData(int startProcess, int numOfProcesses)

Summary:
ProcessName, LastMsg, Dateof Lastmsg, Stateofprocess, Howlongfromthe last update.

Feature:
Search by processName and sort (individual).

Propose API UI
list<summary> fetchSummaryData(int sortOption, string search, int page, int linesPerPage)

fetchSummaryData(ProcessName(sort), null, 6, 10)
=> database . first get

UI Team:
I have to go guys... sorry... Let me know if you need anything. I will check this document tonight and answer any questions you have on here... you can also set up another meeting on here if you want (whatever time too) I will check this doc and see what you guys want.

Hey Kyle.
Sorry my laptop was overheated and turned off by itself. Our meeting was good enough by that time I think. I see we have agreed on the basic idea of API calls. Chinh will write details of the calls (what we want from UI side) and send you soon. We can have further discussion after reading those details.

UI API:

Main page:
(1) Returntype GetProcessSumary(sortoption <enum>, searchpattern<string>, pagenum<int>&, numentriesperpage<int>);
List <Summary> GetProcessSumaryList (startid <iint>, numentries<int>, sortoption <enum>);
With:
Returntype = Struct { List<Process> entries, int totalpage };
Process = Struct { int ProcessId, string processname, string lastmsg, datetime lastmsgdate, processState <enum> };

Message Inquiry:
(2) Returntype GetMsgInquiry(processname<string>, priority<int>, sortoption<enum>,

category<enum>, pagenum<int>&, numentriesperpage<int>);

With:

Returntype = Struct { List<Msg> entries, int totalpage };

Msg = Struct { int ProcessId, string processname, string msgdetail, datetime msgdate, int priority };

category<enum> : can be either startup msg, shutdown msg

Individual process page:

(3) Returntype1 GetMsgsByProcessId(processId<int>, sortoption<enum>, pagenum<int>&, numentriesperpage<int>); // For msg table

(4) Returntype2 GetJobsByProcessId(processId<int>, sortoption<enum>, pagenum<int>&, numentriesperpage<int>); // For job table

With:

Returntype1 = Struct { List<Msg>, ProcessState <enum> };

Returntype2 = Struct { List<Job>, int totaljob, int totalfinjob, int totalprogressjob };

Job = Struct { string jobname, datetime jobstartdate, int plannedcount, int completecount, double progressing};

Note:

Possible issue and propose solution:

a/ GetMsgInquiry(“Falcon3”, 4, SORTBYNAME, NULL, 10, 5)

=> Filter List<orig_msg> using “Falcon3” as criteria = > List<msg_1>

=> Sort List<msg_1> using SORTBYNAME => List<msg_2>

=> Skip 45 items in List<msg_2> and get the next 5 items in List<msg_2>

=> But List<msg_2> has only 9 items

=> Return the first 5 items in List<msg_2> and modify pagenum to 1

b/ In external method (4),

** if (no plannedcount) return -1;

** if (no completecount) return -1;

** The range of job progressing is [0, 100];