Ver. 1

Unit Test Report for Network Printing System

- **Test Cases Specification**
- **Test Summary Report**

Project Team
T5 Team

Latest update on:
2015-11-10

Team Information

201411316 정진호

201411305 이찬규

201411296 이선명

201411294 이상혁

Table of Contents

Ver. 1

1 Introduction

1.1 Objectives

본 문서는 Network Printing System (이하 NPS) 의 Unit Test 를 수행한 결과에 대한 Report 문서이다. Test 해야할 요소들에 대한 Test Case 와 Test 수행 결과에 대한 내용을 담고 있다.

1.2 References

Software Requirements Analysis (Ver 3.0)

Software Design Specification (Ver 2.0)

Unit Test Plan (Ver 1.0)

2 Unit test case specification

2.1 Test case specification identifier

Identifier	Feature	Valid/Invalid Value	Expected Output
NPS UTC 1000_000	Wait Refill Ink Addition Process	Valid. current_ink_value = 0, wait_refill_ink_value = 0 process->trigger(0)	db->wait_refill_ink_value == 0
NPS UTC 1000_001	Wait Refill Ink Addition Process	Invalid. current_ink_value = 0, wait_refill_ink_value = 0 process->trigger(-10)	db->wait_refill_ink_value == 0
NPS UTC 1000_002	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = 0 process->trigger(MAX)	db->wait_refill_ink_value == MAX
NPS UTC	Wait Refill Ink Addition	Valid	db-

1000_003	Process	current_ink_value = 0, wait_refill_ink_value = 0 process->trigger(5000)	>wait_refill_ink_value == MAX
NPS UTC 1000_004	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = 0 process->trigger(10)	db->wait_refill_ink_value == 10
NPS UTC 1000_005	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = MAX / 2 process->trigger(0)	db->wait_refill_ink_value == MAX/2
NPS UTC 1000_006	Wait Refill Ink Addition Process	Invalid current_ink_value = 0, wait_refill_ink_value = MAX / 2 process->trigger(-10)	db->wait_refill_ink_value == MAX/2
NPS UTC 1000_007	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = MAX / 2 process->trigger(MAX)	db->wait_refill_ink_value == MAX
NPS UTC 1000_008	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = MAX / 2 process->trigger(5000)	db->wait_refill_ink_value == MAX
NPS UTC 1000_009	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = MAX / 2 process->trigger(10)	db->wait_refill_ink_value == MAX/2 + 10
NPS UTC 1000_010	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = MAX process->trigger(0)	db->wait_refill_ink_value == MAX

NPS UTC 1000_011	Wait Refill Ink Addition Process	Invalid current_ink_value = 0, wait_refill_ink_value = MAX process->trigger(-10)	db->wait_refill_ink_value == MAX
NPS UTC 1000_012	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = MAX process->trigger(MAX)	db->wait_refill_ink_value == MAX
NPS UTC 1000_013	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = MAX process->trigger(5000)	db->wait_refill_ink_value == MAX
NPS UTC 1000_014	Wait Refill Ink Addition Process	Valid current_ink_value = 0, wait_refill_ink_value = MAX process->trigger(10)	db->wait_refill_ink_value == MAX
NPS UTC 1000_015	Wait Refill Ink Addition Process	Valid. current_ink_value = MAX / 2, wait_refill_ink_value = 0 process->trigger(0)	db->wait_refill_ink_value == 0
NPS UTC 1000_016	Wait Refill Ink Addition Process	Invalid. current_ink_value = MAX / 2, wait_refill_ink_value = 0 process->trigger(-10)	db->wait_refill_ink_value == 0
NPS UTC 1000_017	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2 , wait_refill_ink_value = 0 process->trigger(MAX)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_018	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = 0	db->wait_refill_ink_value == MAX / 2

		process->trigger(5000)	
NPS UTC 1000_019	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = 0 process->trigger(10)	db->wait_refill_ink_value == 10
NPS UTC 1000_020	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = MAX / 2 process->trigger(0)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_021	Wait Refill Ink Addition Process	Invalid current_ink_value = MAX / 2, wait_refill_ink_value = MAX / 2 process->trigger(-10)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_022	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = MAX / 2 process->trigger(MAX)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_023	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = MAX / 2 process->trigger(5000)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_024	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = MAX / 2 process->trigger(10)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_025	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = MAX process->trigger(0)	db->wait_refill_ink_value == MAX / 2
NPS UTC	Wait Refill Ink Addition	Invalid	db-

1000_026	Process	current_ink_value = MAX / 2, wait_refill_ink_value = MAX process->trigger(-10)	>wait_refill_ink_value == MAX / 2
NPS UTC 1000_027	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = MAX process->trigger(MAX)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_028	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = MAX process->trigger(5000)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_029	Wait Refill Ink Addition Process	Valid current_ink_value = MAX / 2, wait_refill_ink_value = MAX process->trigger(10)	db->wait_refill_ink_value == MAX / 2
NPS UTC 1000_030	Wait Refill Ink Addition Process	Valid. current_ink_value = MAX, wait_refill_ink_value = 0 process->trigger(0)	db->wait_refill_ink_value == 0
NPS UTC 1000_031	Wait Refill Ink Addition Process	Invalid. current_ink_value = MAX, wait_refill_ink_value = 0 process->trigger(-10)	db->wait_refill_ink_value == 0
NPS UTC 1000_032	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = 0 process->trigger(MAX)	db->wait_refill_ink_value == 0
NPS UTC 1000_033	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = 0 process->trigger(5000)	db->wait_refill_ink_value == 0

NPS UTC 1000_034	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = 0 process->trigger(10)	db->wait_refill_ink_value == 0
NPS UTC 1000_035	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = MAX / 2 process->trigger(0)	db->wait_refill_ink_value == 0
NPS UTC 1000_036	Wait Refill Ink Addition Process	Invalid current_ink_value = MAX, wait_refill_ink_value = MAX / 2 process->trigger(-10)	db->wait_refill_ink_value == 0
NPS UTC 1000_037	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = MAX / 2 process->trigger(MAX);	db->wait_refill_ink_value == 0
NPS UTC 1000_038	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = MAX / 2 process->trigger(5000)	db->wait_refill_ink_value == 0
NPS UTC 1000_039	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = MAX / 2 process->trigger(10)	db->wait_refill_ink_value == 0
NPS UTC 1000_040	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = MAX process->trigger(0)	db->wait_refill_ink_value == 0
NPS UTC 1000_041	Wait Refill Ink Addition Process	Invalid current_ink_value = MAX, wait_refill_ink_value = MAX	db->wait_refill_ink_value == 0

		process->trigger(-10)	
NPS UTC 1000_042	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = MAX process->trigger(MAX)	db->wait_refill_ink_value == 0
NPS UTC 1000_043	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = MAX process->trigger(5000)	db->wait_refill_ink_value == 0
NPS UTC 1000_044	Wait Refill Ink Addition Process	Valid current_ink_value = MAX, wait_refill_ink_value = MAX process->trigger(10)	db->wait_refill_ink_value == 0
NPS UTC 1000_045	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = 0 process->trigger(0)	db->wait_refill_paper_value == 0
NPS UTC 1000_046	Wait Refill Paper Addition Process	InValid current_paper_value = 0, wait_refill_paper_value = 0 process->trigger(-10)	db->wait_refill_paper_value == 0
NPS UTC 1000_047	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = 0 process->trigger(MAX)	db->wait_refill_paper_value == MAX
NPS UTC 1000_048	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = 0 process->trigger(5000)	db->wait_refill_paper_value == MAX
NPS UTC	Wait Refill Paper	Valid	db-

1000_049	Addition Process	current_paper_value = 0, wait_refill_paper_value = 0 process->trigger(10)	>wait_refill_paper_ value == 10
NPS UTC 1000_050	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = MAX/2 process->trigger(0)	db- >wait_refill_paper_ value == MAX / 2
NPS UTC 1000_051	Wait Refill Paper Addition Process	InValid current_paper_value = 0, wait_refill_paper_value = MAX/2 process->trigger(-10)	db- >wait_refill_paper_ value == MAX / 2
NPS UTC 1000_052	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = MAX/2 process->trigger(MAX)	db- >wait_refill_paper_ value == MAX
NPS UTC 1000_053	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = MAX/2 process->trigger(5000)	db- >wait_refill_paper_ value == MAX
NPS UTC 1000_054	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = MAX/2 process->trigger(10)	db- >wait_refill_paper_ value == MAX
NPS UTC 1000_055	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = MAX process->trigger(0)	db- >wait_refill_paper_ value == MAX
NPS UTC 1000_056	Wait Refill Paper Addition Process	InValid current_paper_value = 0, wait_refill_paper_value = MAX process->trigger(-10)	db- >wait_refill_paper_ value == MAX

NPS UTC 1000_057	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = MAX process->trigger(MAX)	db->wait_refill_paper_value == MAX
NPS UTC 1000_058	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = MAX process->trigger(5000)	db->wait_refill_paper_value == MAX
NPS UTC 1000_059	Wait Refill Paper Addition Process	Valid current_paper_value = 0, wait_refill_paper_value = MAX process->trigger(10)	db->wait_refill_paper_value == MAX
NPS UTC 1000_060	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = 0 process->trigger(0)	db->wait_refill_paper_value == 0
NPS UTC 1000_061	Wait Refill Paper Addition Process	InValid current_paper_value = MAX/2, wait_refill_paper_value = 0 process->trigger(-10)	db->wait_refill_paper_value == 0
NPS UTC 1000_062	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = 0 process->trigger(MAX)	db->wait_refill_paper_value == MAX / 2
NPS UTC 1000_063	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = 0 process->trigger(5000)	db->wait_refill_paper_value == MAX / 2
NPS UTC 1000_064	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = 0	db->wait_refill_paper_value == 10

		process->trigger(10)	
NPS UTC 1000_065	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = MAX/2 process->trigger(0)	db->wait_refill_paper_value == MAX / 2
NPS UTC 1000_066	Wait Refill Paper Addition Process	InValid current_paper_value = MAX/2, wait_refill_paper_value = MAX/2 process->trigger(-10)	db->wait_refill_paper_value == MAX /2
NPS UTC 1000_067	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = MAX/2 process->trigger(MAX)	db->wait_refill_paper_value == MAX / 2
NPS UTC 1000_068	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = MAX/2 process->trigger(5000)	db->wait_refill_paper_value == MAX / 2
NPS UTC 1000_069	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = MAX/2 process->trigger(10)	db->wait_refill_paper_value == MAX / 2
NPS UTC 1000_070	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = MAX process->trigger(0)	db->wait_refill_paper_value == MAX /2
NPS UTC 1000_071	Wait Refill Paper Addition Process	InValid current_paper_value = MAX/2, wait_refill_paper_value = MAX process->trigger(-10)	db->wait_refill_paper_value == MAX / 2
NPS UTC	Wait Refill Paper	Valid	db-

1000_072	Addition Process	current_paper_value = MAX/2, wait_refill_paper_value = MAX process->trigger(MAX)	>wait_refill_paper_ value == MAX / 2
NPS UTC 1000_073	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = MAX process->trigger(5000)	db- >wait_refill_paper_ value == MAX / 2
NPS UTC 1000_074	Wait Refill Paper Addition Process	Valid current_paper_value = MAX/2, wait_refill_paper_value = MAX process->trigger(10)	db- >wait_refill_paper_ value == MAX / 2
NPS UTC 1000_075	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = 0 process->trigger(0)	db- >wait_refill_paper_ value == 0
NPS UTC 1000_076	Wait Refill Paper Addition Process	InValid current_paper_value = MAX, wait_refill_paper_value = 0 process->trigger(-10)	db- >wait_refill_paper_ value == 0
NPS UTC 1000_077	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = 0 process->trigger(MAX)	db- >wait_refill_paper_ value == 0
NPS UTC 1000_078	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = 0 process->trigger(5000)	db- >wait_refill_paper_ value == 0
NPS UTC 1000_079	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = 0 process->trigger(10)	db- >wait_refill_paper_ value == 0

NPS UTC 1000_080	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = MAX/2 process->trigger(0)	db->wait_refill_paper_value == 0
NPS UTC 1000_081	Wait Refill Paper Addition Process	InValid current_paper_value = MAX, wait_refill_paper_value = MAX/2 process->trigger(-10)	db->wait_refill_paper_value == 0
NPS UTC 1000_082	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = MAX/2 process->trigger(MAX)	db->wait_refill_paper_value == 0
NPS UTC 1000_083	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = MAX/2 process->trigger(5000)	db->wait_refill_paper_value == 0
NPS UTC 1000_084	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = MAX/2 process->trigger(10)	db->wait_refill_paper_value == 0
NPS UTC 1000_085	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = MAX process->trigger(0)	db->wait_refill_paper_value == 0
NPS UTC 1000_086	Wait Refill Paper Addition Process	InValid current_paper_value = MAX, wait_refill_paper_value = MAX process->trigger(-10)	db->wait_refill_paper_value == 0
NPS UTC 1000_087	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = MAX	db->wait_refill_paper_value == 0

		process->trigger(MAX)	
NPS UTC 1000_088	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = MAX process->trigger(5000)	db->wait_refill_paper_value == 0
NPS UTC 1000_089	Wait Refill Paper Addition Process	Valid current_paper_value = MAX, wait_refill_paper_value = MAX process->trigger(10)	db->wait_refill_paper_value == 0
NPS UTC 1000_090	Reset Wait Refill Ink Process	Valid wait_refill_ink_value = 100 process->trigger()	db->wait_refill_ink_value == 0
NPS UTC 1000_091	Reset Wait Refill Ink Process	Valid wait_refill_ink_value = 0 process->trigger()	db->wait_refill_ink_value == 0
NPS UTC 1000_092	Reset Wait Refill Paper Process	Valid wait_refill_paper_value = 100 process->trigger()	db->wait_refill_paper_value == 0
NPS UTC 1000_093	Reset Wait Refill Paper Process	Valid wait_refill_paper_value = 0 process->trigger()	db->wait_refill_paper_value == 0
NPS UTC 1000_094	Refill Ink Process	Valid current_ink_value = 0 wait_refill_ink_value = 0 process->trigger()	db->current_ink_value == 0
NPS UTC 1000_095	Refill Ink Process	Invalid current_ink_value = 0 wait_refill_ink_value = -50 process->trigger()	db->current_ink_value == 0

NPS UTC 1000_096	Refill Ink Process	Valid current_ink_value = 0 wait_refill_ink_value = 100 process->trigger()	db->current_ink_value == 100
NPS UTC 1000_097	Refill Ink Process	Valid current_ink_value = 0 wait_refill_ink_value = 1000 process->trigger()	db->current_ink_value == 100
NPS UTC 1000_098	Refill Ink Process	Valid current_ink_value = 0 wait_refill_ink_value = 50 process->trigger()	db->current_ink_value == 50
NPS UTC 1000_099	Refill Ink Process	Valid current_ink_value = MAX/2 wait_refill_ink_value = 0 process->trigger()	db->current_ink_value == MAX/2
NPS UTC 1000_100	Refill Ink Process	Invalid current_ink_value = MAX/2 wait_refill_ink_value = -50 process->trigger()	db->current_ink_value == MAX/2
NPS UTC 1000_101	Refill Ink Process	Valid current_ink_value = MAX/2 wait_refill_ink_value = 100 process->trigger()	db->current_ink_value == MAX/2 + 100
NPS UTC 1000_102	Refill Ink Process	Valid current_ink_value = MAX/2 wait_refill_ink_value = 1000 process->trigger()	db->current_ink_value == MAX/2 + 100
NPS UTC 1000_103	Refill Ink Process	Valid current_ink_value = MAX/2 wait_refill_ink_value = 50	db->current_ink_value == MAX/2 + 50

		process->trigger()	
NPS UTC 1000_104	Refill Ink Process	Valid current_ink_value = MAX wait_refill_ink_value = 0 process->trigger()	db->current_ink_value == MAX
NPS UTC 1000_105	Refill Ink Process	Invalid current_ink_value = MAX wait_refill_ink_value = -50 process->trigger()	db->current_ink_value == MAX
NPS UTC 1000_106	Refill Ink Process	Valid current_ink_value = MAX wait_refill_ink_value = 100 process->trigger()	db->current_ink_value == MAX
NPS UTC 1000_107	Refill Ink Process	Valid current_ink_value = MAX wait_refill_ink_value = 1000 process->trigger()	db->current_ink_value == MAX
NPS UTC 1000_108	Refill Ink Process	Valid current_ink_value = MAX wait_refill_ink_value = 50 process->trigger()	db->current_ink_value == MAX
NPS UTC 1000_109	Refill Paper Process	Valid current_paper_value = 0 wait_refill_paper_value = 0 process->trigger()	db->current_paper_value == 0
NPS UTC 1000_110	Refill Paper Process	Invalid current_paper_value = 0 wait_refill_paper_value = -5 process->trigger()	db->current_paper_value == 0
NPS UTC	Refill Paper Process	Valid	db-

1000_111		current_paper_value = 0 wait_refill_paper_value = 10 process->trigger()	>current_paper_value == 10
NPS UTC 1000_112	Refill Paper Process	Valid current_paper_value = 0 wait_refill_paper_value = 1000 process->trigger()	db->current_paper_value == 10
NPS UTC 1000_113	Refill Paper Process	Valid current_paper_value = 0 wait_refill_paper_value = 5 process->trigger()	db->current_paper_value == 5
NPS UTC 1000_114	Refill Paper Process	Valid current_paper_value = MAX/2 wait_refill_paper_value = 0 process->trigger()	db->current_paper_value == MAX/2
NPS UTC 1000_115	Refill Paper Process	Invalid current_paper_value = MAX/2 wait_refill_paper_value = -5 process->trigger()	db->current_paper_value == MAX/2
NPS UTC 1000_116	Refill Paper Process	Valid current_paper_value = MAX/2 wait_refill_paper_value = 10 process->trigger()	db->current_paper_value == MAX/2 + 10
NPS UTC 1000_117	Refill Paper Process	Valid current_paper_value = MAX/2 wait_refill_paper_value = 1000 process->trigger()	db->current_paper_value == MAX / 2 + 10
NPS UTC 1000_118	Refill Paper Process	Valid current_paper_value = MAX/2 wait_refill_paper_value = 5 process->trigger()	db->current_paper_value == MAX / 2 + 5

NPS UT C 1000_119	Refill Paper Process	Valid current_paper_value = MAX wait_refill_paper_value = 0 process->trigger()	db- >current_paper_val ue == MAX
NPS UTC 1000_120	Refill Paper Process	Invalid current_paper_value = MAX wait_refill_paper_value = -5 process->trigger()	db- >current_paper_val ue == MAX
NPS UTC 1000_121	Refill Paper Process	Valid current_paper_value = MAX wait_refill_paper_value = 10 process->trigger()	db- >current_paper_val ue == MAX
NPS UTC 1000_122	Refill Paper Process	Valid current_paper_value = MAX wait_refill_paper_value = 1000 process->trigger()	db- >current_paper_val ue == MAX
NPS UTC 1000_123	Refill Paper Process	Valid current_paper_value = MAX wait_refill_paper_value = 5 process->trigger()	db- >current_paper_val ue == MAX

NPS UTC
1000_124
	Virtual Refill Interface	/*정상 작동 테스트*/
		Valid
		struct RequestRefillInfo info;
		int client;
		test_buf[] = {0,0,0,0,10};
		
		struct sockaddr_in server;
		bzero(&server, sizeof(server));
		server.sin_addr.s_addr =
		inet_addr("127.0.0.1");
		s erver.sin_family = AF_INET;
		server.sin_port =
			info.type == 0 &&
			info.value == 10

		htons(6500);
		
		client = create_test_client();
		
		sendto(client, test_buf, 8, 0, (struct sockaddr *)&server, sizeof(server));
		
		info = interface->get_request_info();	

NPS UTC 1000_125	Virtual Refill Controller	//IDLE => Wait_Refill_Ink
		Valid
		current_ink_value = 0
		wait_ink_value = 0
		char test_buf[] = {0,0,0,10,10};
		sendto(client, test_buf, 8, 0, (struct sockaddr *)&server, sizeof(server));
		controller->tick();
			*(db->wait_refill_ink_value) == 2570

NPS UTC 1000_126	Virtual Refill Controller	//Wait_Refill_Ink => Reill_Ink => Wait_refill_Ink
		Valid
		
		for(i = 0; i < 21; i++)
		{
		controller->tick();
		}
			*(db->wait_refill_ink_value) == 2370 && *(db->current_ink_value) == 200

NPS UTC 1000_127	Virtual Refill Controller	//Wait_Refill_Ink => ... => IDLE
		Valid
		
		while(*(db->wait_refill_ink_value)){
			*(db->wait_refill_ink_value) == 0 && *(db->current_ink_value

		controller->tick(); }	) == 2570

NPS UTC
1000_128
	Virtual Refill Controller	//IDLE => Wait_Refill_Paper
		Valid
		
		current_paper_value = 0
		wait_paper_value = 0
		
		char test_buf[] = {1,0,0,0,60};
		sendto(client, test_buf, 8, 0,
		(struct sockaddr *)&server,
		sizeof(server));
		controller->tick();
			*(db-
			>wait_refill_paper_
			value) == 60)

NPS UTC
1000_129
	Virtual Refill Controller	//Wait_Refill_Ink => Reill_Ink
		=> Wait_refill_Ink
		Valid
		
		for(int i = 0; i < 21; i++)
		{
		controller->tick();
		}
			*(db-
			>wait_refill_paper_
			value) == 40 &&
			*(db-
			>current_paper_val
			ue) == 20

NPS UTC
1000_130
	Virtual Refill Controller	//Wait_Refill_Paper => ... =>
		IDLE
		Valid
		
		while(*(db-
		>wait_refill_paper_value))
		{
		controller->tick();
		}
			*(db-
			>wait_refill_paper_
			value) == 0 &&
			*(db-
			>current_paper_val
			ue) == 60

NPS UTC 2000_000	H/W Stop Process	*JobQueue 가 비어있는 경우 */ Valid /	// 프로세스가 작동하지 않아야 한다.

		JobQueue->jobs_info[0].job_id = -1 // job_id = -1 일 경우 Job_info 가 비어있음 JobQueue->jobs_info[0].state = 0 process->trigger()	test_jobs_info.state == 0

NPS UTC 2000_001	H/W Stop Process	/*인쇄중이거나 대기중인경우*/
		Valid
		JobQueue->jobs_info[0].job_id = 1;
		JobQueue->jobs_info[0].state = 0;
		process->trigger()
			//정상 작동
			test_jobs_info.state == 1

NPS UTC 2000_002	H/W Stop Process	/*현재 삭제중인 경우*/ Valid JobQueue->jobs_info[0].job_id = 1 JobQueue->jobs_info[0].state = 1 process->trigger()	test_jobs_info.state == 1
NPS UTC 2000_003	H/W Stop Interface	//hw_stop 함수 테스트 Valid test_buf[0] = 1 sendto(csock, test_buf, 1, 0, (struct sockaddr *)&sa, sizeof(sa)) output = interface->hw_stop()	output == 1
NPS UTC 2000_004	H/W stop Controller	Valid test_buf[1] = {1}	test_jobs_info.state == 1

		sendto(csock, test_buf, 1, 0, (struct sockaddr *)&server, sizeof(server)); process->tick()	

NPS UTC
3000_000
	auth_and_dispatch_contr
	oller
		// 현재 상태가 대기이며 인쇄
		요청 패킷이옴 (권한 실패)
		// 패킷 데이터는 프로토콜 참고
		current_state = ready
		packet->length = ~;
		packet->data = /* 0, 0, test,
		test, content:test 로 테스트 */
		
		current_state == fail_auth ?	

NPS UTC
3000_001
	auth_and_dispatch_contr
	oller
		// 관리자 명령이 온 경우
		// 그러나 일반사용자가 보냄
		test_packet_data[0] = 2;
		user_id = “user0”
		user_pw = “pass0”
			execute
			FailAuthDispatcher
			trigger

NPS UTC
3001_002
	auth_and_dispatch_contr
	oller
		test_packet_data[0] = 0;
		user_id = “admin”
		user_pw= “admin”
		// 이하 패킷 요청 유저는 동일
			execute
			PrintValidation
			Process

NPS UTC 3000_003	auth_and_dispatch_contr oller	test_packet_data[0] = 1;	execute RequestToCancelP rintingProcess trigger
NPS UTC 3000_004	auth_and_dispatch_contr oller	test_packet_data[0] = 2;	execute UserCreationProce ss trigger
NPS UTC 3000_005	auth_and_dispatch_contr oller	test_packet_data[0] = 3;	execute UserRemovalProce

			ss trigger
NPS UTC 3000_006	auth_and_dispatch_controller	test_packet_data[0] = 4;	execute UserListingProcess trigger
NPS UTC 3000_007	fail_auth_dispatcher	// send_packet 을 제대로 호출하는가? execute trigger	trigger sends packet
NPS UTC 3000_008	print_validation_process	test_jobs_info[JOB_MAX_SIZE-2].job_id = -1 _get_empty_job_index() == JOB_MAX_SIZE - 2	
NPS UTC 3000_009	print_validation_process	// test for get_require_ink char *buf = "Cowabunga Cowabunga Cowabunga ₩n";	_get_require_ink(buf, 27) == 27
NPS UTC 3000_010	print_validation_process	// test for get_require_paper char *buf = "a₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩na₩n";	_get_require_paper(buf, 40) == 2

NPS UTC 3000_011	print_validation_process	// Test for get_job_queue_require_ink
		res = get_job_queue_require_ink
			res == 0

NPS UTC 3000_012	print_validation_process	// Test for get_job_queue_require_paper
		res = get_job_queue_require_paper
			res == 0

NPS UTC 3000_013	print_validation_process	// Test for trigger
		trigger();
			res_type == 0

NPS UTC	print_validation_process	// Test for trigger	res_type == 2

3000_014		// 잉크 부족
		current_ink_value = 0;
		trigger();	

NPS UTC
3000_015
	print_validation_process	// Test for trigger
		// 용지 부족
		current_paper_value = 0;
		trigger();
			res_type == 3

NPS UTC
3000_016
	rquest_to_cancel_printing_process	// 첫번째 작업취소
		test_packet.data =
		"₩0₩0₩0₩0₩0user0₩0₩0₩0₩0₩0₩0pass0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩1",
		process->trigger(&test_packet, 0);
			test_jobs_info[0].state == REMOVING

NPS UTC
3000_017
	rquest_to_cancel_printing_process	// 두번째 작업 취소
		test_packet.data =
		"₩0₩0₩0₩0₩0user1₩0₩0₩0₩0₩0₩0pass0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩2",
		process->trigger(&test_packet, 0);
			test_jobs_info[0].job_id == 1 &&
			test_jobs_info[1].job_id == 3 &&
			test_jobs_info[2].job_id == 4 &&
			test_jobs_info[3].job_id == 5 &&
			test_jobs_info[4].job_id == -1

NPS UTC 3000_018	rquest_to_cancel_printing_process	// 세번째 작업 취소 test_packet.data = "₩0₩0₩0₩0₩0user2₩0₩0₩0₩0	test_jobs_info[0].job_id == 1 &&

		₩0₩0₩0pass2₩0₩0₩0₩0₩0₩0₩0₩0₩0₩3", process->trigger(&test_packet, 0);	test_jobs_info[1].job_id == 2 && test_jobs_info[2].job_id == 4 && test_jobs_info[3].job_id == 5 && test_jobs_info[4].job_id == -1

NPS UTC 3000_019	rquest_to_cancel_printing_process	// 네번째 작업 취소
		test_packet.data = "₩0₩0₩0₩0₩0user3₩0₩0₩0₩0₩0₩0₩0pass3₩0₩0₩0₩0₩0₩0₩0₩0₩0₩4",
		process->trigger(&test_packet, 0);
			test_jobs_info[0].job_id == 1 &&
			test_jobs_info[1].job_id == 2 &&
			test_jobs_info[2].job_id == 3 &&
			test_jobs_info[3].job_id == 5 &&
			test_jobs_info[4].job_id == -1

NPS UTC 3000_020	rquest_to_cancel_printing_process	// 다섯번째 작업 취소
		test_packet.data = "₩0₩0₩0₩0₩0user4₩0₩0₩0₩0₩0₩0₩0pass4₩0₩0₩0₩0₩0₩0₩0₩0₩0₩5",
			test_jobs_info[0].job_id == 1 &&
			test_jobs_info[1].jo

		process->trigger(&test_packet, 0);	b_id == 2 && test_jobs_info[2].job_id == 3 && test_jobs_info[3].job_id == 4 && test_jobs_info[4].job_id == -1

NPS UTC 3000_021	user_creation_process	// _exist_user 함수가 유저를 정상적으로 찾아내는가
		Valid
		
		test_user.user_id = "test₩0₩0₩0₩0₩0₩0₩0₩0"
		
		v_header.nextInfo = &test_user;
		test_user.prevInfo = &v_header;
		test_user.nextInfo = &v_tailer;
		v_tailer.prevInfo = &test_user;
		
		test_user_packet = {
		        .data = "₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0test₩0₩0₩0₩0₩0₩0₩0₩0scott₩0₩0₩0₩0₩0₩0₩0",
		        .length = TYPE_LENGTH + REQ_ID_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH +
			output == 1

		USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH, .sock = 0 }; process- >trigger(&test_user_packet); output = process- >_exist_user("test₩0₩0₩0₩0₩0 ₩0₩0₩0")	

NPS UTC
3000_022
	user_creation_process	//DB 에 유저 정보가 없을경우
		_exist_user 함수가 유저가
		없다고 제대로 체크하는가
		Valid
		
		v_header.prevInfo = NULL;
		v_header.nextInfo = &v_tailer;
		v_tailer.prevInfo = &v_header;
		v_tailer.nextInfo = NULL;
		
		test_user_packet = {
		.data =
		"₩0₩0₩0₩0₩0₩0₩0₩0₩0₩
		0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩
		0₩0₩0₩0₩0₩0₩0test₩0₩0
		₩0₩0₩0₩0₩0₩0scott₩0₩0₩0
		₩0₩0₩0₩0",
		.length = TYPE_LENGTH +
		REQ_ID_LENGTH +
		USER_ID_MAX_LENGTH +
		USER_PW_MAX_LENGTH +
		USER_ID_MAX_LENGTH +
		USER_PW_MAX_LENGTH,
			output != 1

		.sock = 0 }; process- >trigger(&test_user_packet); output = process- >_exist_user("test₩0₩0₩0₩0₩0 ₩0₩0₩0")	

NPS UTC
3000_023
	user_creation_process	//DB 에 유저 정보가 있을경우 잘
		찾는지(1)
		Valid
		
		test_user.user_id = "test"
		test_user2.user_id = "a"
		test_user3.user_id = "b"
		
		v_header.nextInfo = &test_user;
		test_user.prevInfo = &v_header;
		test_user.nextInfo =
		&test_user2;
		test_user2.prevInfo =
		&test_user;
		test_user2.nextInfo =
		&test_user3
		test_user3.prevInfo =
		&test_user2
		test_user3.nextInfo = &v_tailer;
		v_tailer.prevInfo = &test_user3;
		
		test_user_packet = {
		.data =
		"₩0₩0₩0₩0₩0₩0₩0₩0₩0₩
		0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩
			output == 1

		0₩0₩0₩0₩0₩0₩0₩0test₩0₩0₩0₩0₩0₩0₩0₩0scott₩0₩0₩0₩0₩0₩0₩0₩0",
		.length = TYPE_LENGTH + REQ_ID_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH,
		.sock = 0
		};
		process->trigger(&test_user_packet);
		
		output = process->_exist_user("test₩0₩0₩0₩0₩0₩0₩0₩0₩0")	

NPS UTC 3000_024	user_creation_process	//DB 에 유저 정보가 있을경우 잘 찾는지(2)
		Valid
		
		test_user.user_id = "test"
		test_user2.user_id = "a"
		test_user3.user_id = "b"
		
		v_header.nextInfo = &test_user;
		test_user2.prevInfo = &v_header;
		test_user2.nextInfo = &test_user;
		test_user.prevInfo = &test_user2;
		test_user.nextInfo = &test_user3;
			output == 1

		test_user3.prevInfo = &test_user;
		test_user3.nextInfo = &v_tailer;
		v_tailer.prevInfo = &test_user3;
		
		test_user_packet = {
		.data = "₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0test₩0₩0₩0₩0₩0₩0₩0₩0scott₩0₩0₩0₩0₩0₩0₩0",
		.length = TYPE_LENGTH + REQ_ID_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH,
		.sock = 0
		};
		process->trigger(&test_user_packet);
		
		output = process->_exist_user(“test₩0₩0₩0₩0₩0₩0₩0₩0”)	

NPS UTC 3000_025	user_creation_process	//DB 에 유저 정보가 있을경우 잘 찾는지(3) Valid test_user.user_id = "test" test_user2.user_id = "a" test_user3.user_id = "b"	output == 1

		v_header.nextInfo = &test_user2;
		test_user2.prevInfo = &v_header;
		test_user2.nextInfo = &test_user3;
		test_user3.prevInfo = &test_user2;
		test_user3.nextInfo = &test_user;
		test_user.prevInfo = &test_user3;
		test_user.nextInfo = &v_tailer;
		v_tailer.prevInfo = &test_user3;
		
		test_user_packet = {
		.data = "₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0test₩0₩0₩0₩0₩0₩0₩0₩0scott₩0₩0₩0₩0₩0₩0₩0",
		.length = TYPE_LENGTH + REQ_ID_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH,
		.sock = 0
		};
		process->trigger(&test_user_packet);
		
		output=_exist_user("test₩0₩0₩0₩0₩0₩0₩0₩0")) == 1	

NPS UTC 3000_026	user_creation_process	// DB 에 요청받은 유저 id 가 없는경우
		Valid
		
		v_header.prevInfo = NULL;
		v_header.nextInfo = &v_tailer;
		v_tailer.prevInfo = &v_header;
		v_tailer.nextInfo = NULL;
		
		test_user_packet = {
		.data = "₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0test₩0₩0₩0₩0₩0₩0₩0₩0scott₩0₩0₩0₩0₩0₩0₩0",
		.length = TYPE_LENGTH + REQ_ID_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH,
		.sock = 0
		};
		
		output = _exist_user(“test₩0₩0₩0₩0₩0₩0₩0₩0”)
			output == 1

NPS UTC 3000_027	user_creation_process	// DB 에 요청받은 유저 id 가 이미 있는경우
		test_user = {
		.user_id = "test₩0₩0₩0₩0₩0₩0₩0₩0"
		};
			call NetworkTxInterface->send_packet ?

		v_header.prevInfo = NULL;
		v_header.nextInfo = &test_user;
		test_user.prevInfo = &v_header;
		test_user.nextInfo = &v_tailer;
		v_tailer.prevInfo = &test_user;
		v_tailer.nextInfo = NULL;
		test_user_packet = {
		        .data =
		"₩0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩
		0₩0₩0₩0₩0₩0₩0₩0₩0₩0₩
		0₩0₩0₩0₩0₩0₩0test₩0₩0
		₩0₩0₩0₩0₩0₩0scott₩0₩0₩0
		₩0₩0₩0₩0",
		        .length = TYPE_LENGTH +
		REQ_ID_LENGTH +
		USER_ID_MAX_LENGTH +
		USER_PW_MAX_LENGTH +
		USER_ID_MAX_LENGTH +
		USER_PW_MAX_LENGTH,
		        .sock = 0
		};
		
		process-
		>trigger(&test_user_packet);	

NPS UTC 3000_028	user_listing_process	/* 정상 작동 테스트 */
		Valid
		v_header.nextInfo = &v_tailer;
		v_tailer.prevInfo = &v_header;
		// 테스트 유저 추가
		for (i = 0 ; i < 10 ; i++) {
		        // 구조체 초기화 (공간
		할당)
		        tmp_usr[i] = (struct
			call NetworkTxInterface->send_packet ?

		UserInfo *)malloc(sizeof(struct UserInfo));
		memset(tmp_usr[i], 0, sizeof(struct UserInfo));
		// ID
		memset(buf, 0, 12);
		sprintf(buf, "user%d", i);
		strcpy(tmp_usr[i]->user_id, buf);
		// PW
		memset(buf, 0, 12);
		sprintf(buf, "pass%d", i);
		strcpy(tmp_usr[i]->user_pw, buf);
		
		// 연결
		tmp_lnk = test_user_database.tailer->prevInfo;
		
		tmp_lnk->nextInfo = tmp_usr[i];
		tmp_usr[i]->prevInfo = tmp_lnk;
		tmp_usr[i]->nextInfo = test_user_database.tailer;
		test_user_database.tailer->prevInfo = tmp_usr[i];
		}
		test_user_packet = {
		.data = "₩0₩0₩0₩0₩0",
		.length = TYPE_LENGTH + REQ_ID_LENGTH,
		.sock = 0
		};	

		process->trigger(&test_user_packet);	

NPS UTC 3000_029	user_removal_process	// 존재하지 않는 유저의 삭제 Valid
		
		// 리스트 초기화 (연결 생성)
		v_header.prevInfo = NULL
		v_header.nextInfo = &v_tailer
		v_tailer.prevInfo = &v_header
		v_tailer.nextInfo = NULL
		
		// 테스트 유저 생성
		test_user = (struct UserInfo *)malloc(sizeof(struct UserInfo))
		strcpy(test_user->user_id, "bill_gates_")
		strcpy(test_user->user_pw, "bill_gates_")
		v_header.nextInfo = test_user
		test_user->prevInfo = &v_header
		test_user->nextInfo = &v_tailer
		v_tailer.prevInfo = test_user;
		
		res = 1;
		res &= !process->_remove_exist_user("steve_jobs_"); // 유저 이름이 존재하지 않는다.
		res &= process->_remove_user_id_cmp("steve_jobs_", test_user->user_id); // 테스트 유저와 비교해봐서 없는가?
			res == 1

NPS UTC 3000_030	user_removal_process	// 존재하는 유저를 삭제
		Valid
		start = TYPE_LENGTH + REQ_ID_LENGTH + USER_ID_MAX_LENGTH + USER_PW_MAX_LENGTH;
		
		p.data = (char*)malloc(start + 12);
		strcpy((char*)(p.data + start), "bill_gates_");
		p.length = start+12;
		process->trigger(&p);
			((v_header.nextInfo == &v_tailer) && (v_tailer.prevInfo == &v_header)) == 1

NPS UTC 4000_000	send_status_process	Valid job_info[0].state = 0 state = 0 (no wait_refill) char *msg = process->_transform();	*msg = "current_state: printing"
NPS UTC 4000_001	send_status_process	Valid job_info[0].state = 0 state = 1 (wait_refill) char *msg = process->_transform();	*msg = "current_state: waiting"
NPS UTC 4000_002	send_status_process	Valid job_info[0].state = 1 char *msg = process->_transform();	*msg = "currnt_state: removing"
NPS UTC 4000_003	send_status_process	Valid test_ink_value = 140 char *msg = process->_transform();	*msg = "ink: 140"

NPS UTC 4000_004	send_status_process	Valid test_paper_value = 5 char *msg = process->_transform();	*msg = "paper: 5"

NPS UTC 5000_000	decrease_amount_of_ink_paper	Valid
		db->current_ink_value = 150;
		db->current_paper_value = 2;
		trigger(100);
			db->current_ink_value ==50 &&
			db->current_paper_value == 1;

NPS UTC 5000_001	decrease_amount_of_ink_paper	Invalid db->current_ink_value = 150 db->current_paper_value = 0 trigger(151)	db->current_ink_value == 0 db->current_paper_value == 0
NPS UTC 5000_002	decrease_amount_of_ink_paper	Invalid db->current_ink_value = 0 trigger(-1)	db->current_ink_value == 0

NPS UTC 5000_003	job_queue_pop	//JobQueue 에 데이터가 0 개
		Valid
		
		for(i = 0; i < JOB_MAX_SIZE; i++){
		test_jobs_info[i].job_id = -1;
		test_jobs_info[i].fd = -1;
		}
		
		process->trigger();
		
		boolean = 1;
		for(i = 0; i < JOB_MAX_SIZE;
			boolean == 1

		i++){
		boolean = boolean &&
		(test_jobs_info[i].job_id == -1);
		}	

NPS UTC
5000_004
	job_queue_pop	//JobQueue 에 데이터가 1 개
		Valid
		for(i = 0; i < 1; i++){
		test_jobs_info[i].job_id = i;
		test_jobs_info[i].fd = -1;
		}
		for(i = 1; i < JOB_MAX_SIZE;
		i++){
		test_jobs_info[i].job_id = -
		1;
		test_jobs_info[i].fd = -1;
		}
		
		process->trigger();
		
		boolean = 1;
		for(i = 0; i < JOB_MAX_SIZE;
		i++){
		boolean &=
		(test_jobs_info[i].job_id == -1);
		}
			boolean == 1

NPS UTC
5000_005
	job_queue_pop	//JobQueue 에 데이터가 2 개
		Valid
		for(i = 0; i < 2; i++){
		test_jobs_info[i].job_id = i;
		test_jobs_info[i].fd = -1;
		}
		for(i = 2; i < JOB_MAX_SIZE;
		i++){
			boolean == 1

		test_jobs_info[i].job_id = -1;
		test_jobs_info[i].fd = -1;
		}
		
		process->trigger();
		
		boolean = 1;
		for(i = 0; i < 1; i++){
		boolean &= (test_jobs_info[i].job_id == i+1);
		}
		for(i = 1; i < JOB_MAX_SIZE; i++){
		boolean &= (test_jobs_info[i].job_id == -1);
		}	

NPS UTC 5000_006	job_queue_pop	//JobQueue 에 데이터가 3 개 Valid
		for(i = 0; i < 3; i++){
		test_jobs_info[i].job_id = i;
		test_jobs_info[i].fd = -1;
		}
		for(i = 3; i < JOB_MAX_SIZE; i++){
		test_jobs_info[i].job_id = -1;
		test_jobs_info[i].fd = -1;
		}
		
		process->trigger();
			boolean == 1

		boolean = 1;
		for(i = 0; i < 2; i++){
		boolean &=
		(test_jobs_info[i].job_id == i+1);
		}
		for(i = 2; i < JOB_MAX_SIZE;
		i++){
		boolean &=
		(test_jobs_info[i].job_id == -1);
		}	

NPS UTC
5000_007
	job_queue_pop	//JobQueue 에 데이터가 4 개
		for(i = 0; i < 4; i++){
		test_jobs_info[i].job_id = i;
		test_jobs_info[i].fd = -1;
		}
		for(i = 4; i < JOB_MAX_SIZE;
		i++){
		test_jobs_info[i].job_id = -
		1;
		test_jobs_info[i].fd = -1;
		}
		
		process->trigger();
		
		boolean = 1;
		for(i = 0; i < 3; i++){
		boolean &=
		(test_jobs_info[i].job_id == i+1);
		}
		for(i = 3; i < JOB_MAX_SIZE;
		i++){
		boolean &=
		(test_jobs_info[i].job_id == -1);
		}
			boolean == 1

NPS UTC 5000_008	job_queue_pop	//JobQueue 에 데이터가 5 개
		for(i = 0; i < 5; i++){
		test_jobs_info[i].job_id = i;
		test_jobs_info[i].fd = -1;
		}
		
		process->trigger();
		
		boolean = 1;
		for(i = 0; i < 4; i++){
		boolean &= (test_jobs_info[i].job_id == i+1);
		}
		for(i = 4; i < JOB_MAX_SIZE; i++){
		boolean &= (test_jobs_info[i].job_id == -1);
		}
			boolean == 1

NPS UTC 5000_009	save_print_result	print_test("test, "test", 0);	buf == output
NPS UTC 5000_010	save_print_result	str = "testtest….." // 32 자 print_test(str, str[30 자], 0);	buf == output
NPS UTC 5000_011	save_print_result	str = "a₩na₩na…." // 12 줄 print_test(str, str[10 줄], 0);	buf == output
NPS UTC 5000_012	next_page	Valid test_jobs_info[0].current_page = 0 trigger()	test_jobs_info[0].current_page == 1

NPS UTC 5000_013	printing_controller	//IDLE there is no job
		//don't call any function ??
		Valid
			!(call any function) ??

		test_jobs_info[0].job_id = -1;
		
		controller->tick();
		for(i = 0; i < 20; i++)
		controller->tick();
		controller->tick();	

NPS UTC 5000_014	printing_controller	//IDLE -> job_ready -> Printing -> job_ready
		//call Decrease amount, Save Print result, Next Page ??
		Valid
		test_jobs_info[0].job_id = 0;
		controller->tick();
		for(i = 0; i < 20; i++)
		controller->tick();
		controller->tick();
			call decrease_amount_ink_paper, save_print_result, next_page ??

NPS UTC 5000_015	printing_controller	//job_ready -> System_Wait -> job_ready
		//!(call Decrease amount, Save Print result, Next Page) ??
		
		Valid
		
		test_wait_ink_value = 1
		controller->tick();
			!(call any function) ??

NPS UTC 5000_016	printing_controller	//job_ready -> end_job -> IDLE
		//call job_queue_pop ??
		
		Valid
			call job_queue_pop

		test_wait_ink_value = 0 test_jobs_info[0].current_page = test_jobs_info[0].total_page controller->tick(); controller->tick();	
NPS UTC 5000_017	printing_controller	//IDLE -> job_ready -> end_job -> IDLE Valid test_wait_ink_value = 0 cont	

2.2 Test items

2.3 Input specifications

2.4 Output specifications

3 Environmental needs

NPS 는 POSIX API 를 이용하므로 POSIX 호환 환경에서 실행되어야 한다 (ex. Cygwin, Linux).

4 Unit test summary report

4.1 Test summary report identifier

4.2 Evaluation

```
[TEST] NPS UTC 3000_000 AUTH_AND_DISPATCH_CONTROLLER - unit_test_1                    [SUCCESS]
[TEST] NPS UTC 3000_001 AUTH_AND_DISPATCH_CONTROLLER - unit_test_2                    [SUCCESS]
[TEST] NPS UTC 3000_002 AUTH_AND_DISPATCH_CONTROLLER - unit_test_3                    [SUCCESS]
[TEST] NPS UTC 3000_003 AUTH_AND_DISPATCH_CONTROLLER - unit_test_4                    [SUCCESS]
[TEST] NPS UTC 3000_004 AUTH_AND_DISPATCH_CONTROLLER - unit_test_5                    [SUCCESS]
[TEST] NPS UTC 3000_005 AUTH_AND_DISPATCH_CONTROLLER - unit_test_6                    [SUCCESS]
[TEST] NPS UTC 3000_006 AUTH_AND_DISPATCH_CONTROLLER - unit_test_7                    [SUCCESS]
[TEST] NPS UTC 3000_011 REQUEST_TO_CANCEL_PRINTING_PROCESS - unit_test_1                      [SUCCESS]
[TEST] NPS UTC 3000_012 REQUEST_TO_CANCEL_PRINTING_PROCESS - unit_test_2                      [SUCCESS]
[TEST] NPS UTC 3000_013 REQUEST_TO_CANCEL_PRINTING_PROCESS - unit_test_3                      [SUCCESS]
[TEST] NPS UTC 3000_014 REQUEST_TO_CANCEL_PRINTING_PROCESS - unit_test_4                      [SUCCESS]
[TEST] NPS UTC 3000_015 REQUEST_TO_CANCEL_PRINTING_PROCESS - unit_test_5                      [SUCCESS]
[TEST] NPS UTC 3000_003 FAIL_AUTH_DISPATCHER - unit_test_1                  [SUCCESS]
[TEST] NPS UTC 3000_004 PRINT_VALIDATION_PROCESS - unit_test_1              [SUCCESS]
[TEST] NPS UTC 3000_005 PRINT_VALIDATION_PROCESS - unit_test_2              [SUCCESS]
[TEST] NPS UTC 3000_006 PRINT_VALIDATION_PROCESS - unit_test_3              [SUCCESS]
[TEST] NPS UTC 3000_007 PRINT_VALIDATION_PROCESS - unit_test_4              [SUCCESS]
[TEST] NPS UTC 3000_008 PRINT_VALIDATION_PROCESS - unit_test_5              [SUCCESS]
Print Length Error 1 - 290
[TEST] NPS UTC 3000_009 PRINT_VALIDATION_PROCESS - unit_test_6              [SUCCESS]
file_size:1
0 1 ink paper1 1 asdfasdf[TEST] NPS UTC 3000_010 PRINT_VALIDATION_PROCESS - unit_test_7                [SUCCESS]
file_size:1
1 0 ink paper1 0 asdfasdf[TEST] NPS UTC 3000_011 PRINT_VALIDATION_PROCESS - unit_test_8                [SUCCESS]
[TEST] NPS UTC 3000_016 USER_CREATION_PROCESS - unit_test__exist_user_1             [SUCCESS]
[TEST] NPS UTC 3000_017 USER_CREATION_PROCESS - unit_test__exist_user_2             [SUCCESS]
[TEST] NPS UTC 3000_018 USER_CREATION_PROCESS - unit_test__exist_user_3             [SUCCESS]
[TEST] NPS UTC 3000_019 USER_CREATION_PROCESS - unit_test__exist_user_4             [SUCCESS]
[TEST] NPS UTC 3000_021 USER_CREATION_PROCESS - unit_test_1                 [SUCCESS]
[TEST] NPS UTC 3000_022 USER_CREATION_PROCESS - unit_test_2                 [SUCCESS]
[TEST] NPS UTC 3000_023 USER_LISTING_PROCESS - unit_test_1                  [SUCCESS]
[TEST] NPS UTC 3000_024 USER_REMOVAL_PROCESS - unit_test_1                  [SUCCESS]
[TEST] NPS UTC 3000_025 USER_REMOVAL_PROCESS - unit_test_2                  [SUCCESS]
[TEST] NPS UTC 2000_003 HW_STOP_INTERFACE - unit_test_1             [SUCCESS]
[TEST] NPS UTC 2000_004 HW_STOP_CONTROLLER - unit_test_1                    [SUCCESS]
[TEST] NPS UTC 2000_000 HW_STOP_PROCESS - unit_test_1               [SUCCESS]
[TEST] NPS UTC 2000_001 HW_STOP_PROCESS - unit_test_2               [SUCCESS]
[TEST] NPS UTC 2000_002 HW_STOP_PROCESS - unit_test_3               [SUCCESS]
[TEST] NPS UTC 1000_090 RESET_WAIT_REFILL_INK_PROCESS - unit_test_1                 [SUCCESS]
[TEST] NPS UTC 1000_091 RESET_WAIT_REFILL_INK_PROCESS - unit_test_2                 [SUCCESS]
[TEST] NPS UTC 1000_092 RESET_WAIT_REFILL_PAPER_PROCESS - unit_test_1               [SUCCESS]
[TEST] NPS UTC 1000_093 RESET_WAIT_REFILL_PAPER_PROCESS - unit_test_1               [SUCCESS]
[TEST] NPS UTC 1000_000 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_1              [SUCCESS]
[TEST] NPS UTC 1000_001 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_2              [SUCCESS]
[TEST] NPS UTC 1000_002 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_3              [SUCCESS]
```

```
[TEST] NPS UTC 1000_003 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_4                 [SUCCESS]
[TEST] NPS UTC 1000_004 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_5                 [SUCCESS]
[TEST] NPS UTC 1000_005 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_6                 [SUCCESS]
[TEST] NPS UTC 1000_006 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_7                 [SUCCESS]
[TEST] NPS UTC 1000_007 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_8                 [SUCCESS]
[TEST] NPS UTC 1000_008 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_9                 [SUCCESS]
[TEST] NPS UTC 1000_009 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_10                [SUCCESS]
[TEST] NPS UTC 1000_010 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_11                [SUCCESS]
[TEST] NPS UTC 1000_011 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_12                [SUCCESS]
[TEST] NPS UTC 1000_012 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_13                [SUCCESS]
[TEST] NPS UTC 1000_013 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_14                [SUCCESS]
[TEST] NPS UTC 1000_014 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_15                [SUCCESS]
[TEST] NPS UTC 1000_015 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_16                [SUCCESS]
[TEST] NPS UTC 1000_016 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_17                [SUCCESS]
[TEST] NPS UTC 1000_017 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_18                [SUCCESS]
[TEST] NPS UTC 1000_018 NPS UTC 1000_018 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_19                      [SUCCESS]
[TEST] NPS UTC 1000_019 NPS UTC 1000_019 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_20                      [SUCCESS]
[TEST] NPS UTC 1000_020 NPS UTC 1000_020 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_21                      [SUCCESS]
[TEST] NPS UTC 1000_021 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_22                [SUCCESS]
[TEST] NPS UTC 1000_022 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_23                [SUCCESS]
[TEST] NPS UTC 1000_023 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_24                [SUCCESS]
[TEST] NPS UTC 1000_024 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_25                [SUCCESS]
[TEST] NPS UTC 1000_025 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_26                [SUCCESS]
[TEST] NPS UTC 1000_026 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_27                [SUCCESS]
[TEST] NPS UTC 1000_027 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_28                [SUCCESS]
[TEST] NPS UTC 1000_028 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_29                [SUCCESS]
[TEST] NPS UTC 1000_029 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_30                [SUCCESS]
[TEST] NPS UTC 1000_030 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_31                [SUCCESS]
[TEST] NPS UTC 1000_031 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_32                [SUCCESS]
[TEST] NPS UTC 1000_032 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_33                [SUCCESS]
[TEST] NPS UTC 1000_033 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_34                [SUCCESS]
[TEST] NPS UTC 1000_034 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_35                [SUCCESS]
[TEST] NPS UTC 1000_035 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_36                [SUCCESS]
[TEST] NPS UTC 1000_036 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_37                [SUCCESS]
[TEST] NPS UTC 1000_037 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_38                [SUCCESS]
[TEST] NPS UTC 1000_038 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_39                [SUCCESS]
[TEST] NPS UTC 1000_039 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_40                [SUCCESS]
[TEST] NPS UTC 1000_040 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_41                [SUCCESS]
[TEST] NPS UTC 1000_041 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_42                [SUCCESS]
[TEST] NPS UTC 1000_042 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_43                [SUCCESS]
[TEST] NPS UTC 1000_043 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_44                [SUCCESS]
[TEST] NPS UTC 1000_044 WAIT_REFILL_INK_ADDITION_PROCESS - unit_test_45                [SUCCESS]
[TEST] NPS UTC 1000_045 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_1                       [SUCCESS]
[TEST] NPS UTC 1000_046 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_2                       [SUCCESS]
[TEST] NPS UTC 1000_047 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_3                       [SUCCESS]
[TEST] NPS UTC 1000_048 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_4                       [SUCCESS]
[TEST] NPS UTC 1000_049 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_5                       [SUCCESS]
[TEST] NPS UTC 1000_050 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_6                       [SUCCESS]
[TEST] NPS UTC 1000_051 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_7                       [SUCCESS]
[TEST] NPS UTC 1000_052 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_8                       [SUCCESS]
[TEST] NPS UTC 1000_053 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_9                       [SUCCESS]
[TEST] NPS UTC 1000_054 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_10                      [SUCCESS]
```

```
[TEST] NPS UTC 1000_054 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_10                    [SUCCESS]
[TEST] NPS UTC 1000_055 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_11                    [SUCCESS]
[TEST] NPS UTC 1000_056 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_12                    [SUCCESS]
[TEST] NPS UTC 1000_057 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_13                    [SUCCESS]
[TEST] NPS UTC 1000_058 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_14                    [SUCCESS]
[TEST] NPS UTC 1000_059 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_15                    [SUCCESS]
[TEST] NPS UTC 1000_060 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_16                    [SUCCESS]
[TEST] NPS UTC 1000_061 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_17                    [SUCCESS]
[TEST] NPS UTC 1000_062 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_18                    [SUCCESS]
[TEST] NPS UTC 1000_063 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_19                    [SUCCESS]
[TEST] NPS UTC 1000_064 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_20                    [SUCCESS]
[TEST] NPS UTC 1000_065 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_21                    [SUCCESS]
[TEST] NPS UTC 1000_066 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_22                    [SUCCESS]
[TEST] NPS UTC 1000_067 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_23                    [SUCCESS]
[TEST] NPS UTC 1000_068 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_24                    [SUCCESS]
[TEST] NPS UTC 1000_069 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_25                    [SUCCESS]
[TEST] NPS UTC 1000_070 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_26                    [SUCCESS]
[TEST] NPS UTC 1000_071 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_27                    [SUCCESS]
[TEST] NPS UTC 1000_072 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_28                    [SUCCESS]
[TEST] NPS UTC 1000_073 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_29                    [SUCCESS]
[TEST] NPS UTC 1000_074 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_30                    [SUCCESS]
[TEST] NPS UTC 1000_075 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_31                    [SUCCESS]
[TEST] NPS UTC 1000_076 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_32                    [SUCCESS]
[TEST] NPS UTC 1000_077 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_33                    [SUCCESS]
[TEST] NPS UTC 1000_078 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_34                    [SUCCESS]
[TEST] NPS UTC 1000_079 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_35                    [SUCCESS]
[TEST] NPS UTC 1000_080 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_36                    [SUCCESS]
[TEST] NPS UTC 1000_081 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_37                    [SUCCESS]
[TEST] NPS UTC 1000_082 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_38                    [SUCCESS]
[TEST] NPS UTC 1000_083 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_39                    [SUCCESS]
[TEST] NPS UTC 1000_084 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_40                    [SUCCESS]
[TEST] NPS UTC 1000_085 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_41                    [SUCCESS]
[TEST] NPS UTC 1000_086 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_42                    [SUCCESS]
[TEST] NPS UTC 1000_087 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_43                    [SUCCESS]
[TEST] NPS UTC 1000_088 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_44                    [SUCCESS]
[TEST] NPS UTC 1000_089 WAIT_REFILL_PAPER_ADDITION_PROCESS - unit_test_45                    [SUCCESS]
[TEST] NPS UTC 1000_094 REFILL_INK_PROCESS - unit_test_1                        [SUCCESS]
[TEST] NPS UTC 1000_095 REFILL_INK_PROCESS - unit_test_2                        [SUCCESS]
[TEST] NPS UTC 1000_096 REFILL_INK_PROCESS - unit_test_3                        [SUCCESS]
[TEST] NPS UTC 1000_097 REFILL_INK_PROCESS - unit_test_4                        [SUCCESS]
[TEST] NPS UTC 1000_098 REFILL_INK_PROCESS - unit_test_5                        [SUCCESS]
[TEST] NPS UTC 1000_099 REFILL_INK_PROCESS - unit_test_6                        [SUCCESS]
[TEST] NPS UTC 1000_100 REFILL_INK_PROCESS - unit_test_7                        [SUCCESS]
[TEST] NPS UTC 1000_101 REFILL_INK_PROCESS - unit_test_8                        [SUCCESS]
[TEST] NPS UTC 1000_102 REFILL_INK_PROCESS - unit_test_9                        [SUCCESS]
[TEST] NPS UTC 1000_103 REFILL_INK_PROCESS - unit_test_10                       [SUCCESS]
[TEST] NPS UTC 1000_104 REFILL_INK_PROCESS - unit_test_11                       [SUCCESS]
[TEST] NPS UTC 1000_105 REFILL_INK_PROCESS - unit_test_12                       [SUCCESS]
[TEST] NPS UTC 1000_106 REFILL_INK_PROCESS - unit_test_13                       [SUCCESS]
[TEST] NPS UTC 1000_107 REFILL_INK_PROCESS - unit_test_14                       [SUCCESS]
[TEST] NPS UTC 1000_108 REFILL_INK_PROCESS - unit_test_15                       [SUCCESS]
[TEST] NPS UTC 1000_109 REFILL_PAPER_PROCESS - unit_test_1                      [SUCCESS]
[TEST] NPS UTC 1000_110 REFILL_PAPER_PROCESS - unit_test_2                      [SUCCESS]
```

```
[TEST] NPS UTC 1000_111 REFILL_PAPER_PROCESS - unit_test_3                    [SUCCESS]
[TEST] NPS UTC 1000_112 REFILL_PAPER_PROCESS - unit_test_4                    [SUCCESS]
[TEST] NPS UTC 1000_113 REFILL_PAPER_PROCESS - unit_test_5                    [SUCCESS]
[TEST] NPS UTC 1000_114 REFILL_PAPER_PROCESS - unit_test_6                    [SUCCESS]
[TEST] NPS UTC 1000_115 REFILL_PAPER_PROCESS - unit_test_7                    [SUCCESS]
[TEST] NPS UTC 1000_116 REFILL_PAPER_PROCESS - unit_test_8                    [SUCCESS]
[TEST] NPS UTC 1000_117 REFILL_PAPER_PROCESS - unit_test_9                    [SUCCESS]
[TEST] NPS UTC 1000_118 REFILL_PAPER_PROCESS - unit_test_10                   [SUCCESS]
[TEST] NPS UTC 1000_119 REFILL_PAPER_PROCESS - unit_test_11                   [SUCCESS]
[TEST] NPS UTC 1000_120 REFILL_PAPER_PROCESS - unit_test_12                   [SUCCESS]
[TEST] NPS UTC 1000_121 REFILL_PAPER_PROCESS - unit_test_13                   [SUCCESS]
[TEST] NPS UTC 1000_122 REFILL_PAPER_PROCESS - unit_test_14                   [SUCCESS]
[TEST] NPS UTC 1000_123 REFILL_PAPER_PROCESS - unit_test_15                   [SUCCESS]
[TEST] NPS UTC 1000_124 VIRTUAL_REFILL_INTERFACE - unit_test_1                [SUCCESS]
[TEST] NPS UTC 1000_125 VIRTUAL_REFILL_CONTROLLER - unit_test_1               [SUCCESS]
[TEST] NPS UTC 1000_126 VIRTUAL_REFILL_CONTROLLER - unit_test_2               [SUCCESS]
[TEST] NPS UTC 1000_127 VIRTUAL_REFILL_CONTROLLER - unit_test_3               [SUCCESS]
[TEST] NPS UTC 1000_128 VIRTUAL_REFILL_CONTROLLER - unit_test_4               [SUCCESS]
[TEST] NPS UTC 1000_129 VIRTUAL_REFILL_CONTROLLER - unit_test_5               [SUCCESS]
[TEST] NPS UTC 1000_130 VIRTUAL_REFILL_CONTROLLER - unit_test_6               [SUCCESS]
[TEST] NPS UTC 5000_014 PRINTING_CONTROLLER - unit_test_1                     [SUCCESS]
[TEST] NPS UTC 5000_015 PRINTING_CONTROLLER - unit_test_2                     [SUCCESS]
[TEST] NPS UTC 5000_016 PRINTING_CONTROLLER - unit_test_3                     [SUCCESS]
[TEST] NPS UTC 5000_017 PRINTING_CONTROLLER - unit_test_4                     [SUCCESS]
[TEST] NPS UTC 5000_017 PRINTING_CONTROLLER - unit_test_5                     [SUCCESS]
[TEST] NPS UTC 5000_003 JOB_QUEUE_POP - unit_test_1                [SUCCESS]
[TEST] NPS UTC 5000_004 JOB_QUEUE_POP - unit_test_2                [SUCCESS]
[TEST] NPS UTC 5000_005 JOB_QUEUE_POP - unit_test_3                [SUCCESS]
[TEST] NPS UTC 5000_006 JOB_QUEUE_POP - unit_test_4                [SUCCESS]
[TEST] NPS UTC 5000_007 JOB_QUEUE_POP - unit_test_5                [SUCCESS]
[TEST] NPS UTC 5000_008 JOB_QUEUE_POP - unit_test_6                [SUCCESS]
[TEST] NPS UTC 5000_012 NEXT_PAGE - unit_test_1           [SUCCESS]
[TEST] NPS UTC 5000_000 DECREASE_AMOUNT_OF_INK_PAPER - unit_test_1                  [SUCCESS]
[TEST] NPS UTC 5000_001 DECREASE_AMOUNT_OF_INK_PAPER - unit_test_2                  [SUCCESS]
[TEST] NPS UTC 5000_002 DECREASE_AMOUNT_OF_INK_PAPER - unit_test_3                  [SUCCESS]
[TEST] NPS UTC 5000_009 SAVE_PRINT_RESULT - unit_test_1            [SUCCESS]
[TEST] NPS UTC 5000_010 SAVE_PRINT_RESULT - unit_test_2            [SUCCESS]
[TEST] NPS UTC 5000_011 SAVE_PRINT_RESULT - unit_test_3            [SUCCESS]
[TEST] NPS UTC 4000_000 SEND_STATUS_PROCESS - unit_test_1                     [SUCCESS]
[TEST] NPS UTC 4000_001 SEND_STATUS_PROCESS - unit_test_2                     [SUCCESS]
[TEST] NPS UTC 4000_002 SEND_STATUS_PROCESS - unit_test_3                     [SUCCESS]
[TEST] NPS UTC 4000_003 SEND_STATUS_PROCESS - unit_test_4                     [SUCCESS]
[TEST] NPS UTC 4000_004 SEND_STATUS_PROCESS - unit_test_5                     [SUCCESS]
New connection from 127.0.0.1 on socket 14
[TEST] NETWORK_RX_INTERFACE - unit_test_1                  [SUCCESS]
[TEST] NETWORK_TX_INTERFACE - unit_test_1                  [SUCCESS]


ALL:191 SUCCESS:191 FAIL:0
```