

Automated Test Results - Old Requirements

<u>Test ID</u> (Prepend AO)	<u>Requirement ID</u>	<u>Description</u>	<u>Unit Test</u>	<u>Input Data/Action</u>	<u>Expected Result</u>	<u>Actual Result</u>
2.0	FR_MOVEMENT	Verify that the user can move the chef up.	ChefTests.moveUp()	<pre>chefTest.move(false, false, false, true, 1);</pre>	<pre>newVelocityInY>initialVelocity</pre>	As expected
2.1	FR_MOVEMENT	Verify that the user can move the chef down.	ChefTests.moveDown()	<pre>chefTest.move(false, false, true, false, 1);</pre>	<pre>newVelocityInY<initialVelocity</pre>	As expected
2.2	FR_MOVEMENT	Verify that the user can move the chef right.	ChefTests.moveRight()	<pre>chefTest.move(false, true, false, false, 1);</pre>	<pre>newVelocityInX>initialVelocity</pre>	As expected
2.3	FR_MOVEMENT	Verify that the user can move the chef left.	ChefTests.moveLeft()	<pre>chefTest.move(true, false, false, false, 1);</pre>	<pre>newVelocityInX<initialVelocity</pre>	As expected
3.0	FR_SWITCH	Verify that the system allows the user to switch chefs.	ChefTests.chefSwitch()	<pre>(chefTest.getUserControlChef()==true) && (chefTest2.getUserControlChef()==false)</pre>	<pre>true</pre>	As expected
4.0	FR_INTERACT	Verify that the system allows the user to interact with the kitchen.	ChefTests.chefCollision()	Sets test chef to colliding and verifies user cannot control it.	No user control	As expected
4.1	FR_INTERACT	Verify that the system allows the user to interact with the kitchen.	ChefTests.chefCollision()	Verifies user control is restored after a short interval.	User control	As expected

<u>Test ID (Prepend AO)</u>	<u>Requirement ID</u>	<u>Description</u>	<u>Unit Test</u>	<u>Input Data/Action</u>	<u>Expected Result</u>	<u>Actual Result</u>
4.2	FR_INTERACT	Verify that the system allows the user to interact with the kitchen.	ChefTests.chefCollision()	Verifies chefs move in opposite directions on collision.	Opposing movement	As expected
4.3	FR_INTERACT	Verify that the system allows the user to interact with the kitchen.	StationTests	Verifies if each station returns the correct item	Each station returns the correct ingredient	As expected
4.3	FR_INTERACT	Verify that the system allows the user to interact with the kitchen.	StationTests.recipeRecognisedByPlateStation()	Verifies if the platestation returns the correct recipe with the correct ingredients place on it	Return correct recipe from ingredients on the plate station	As expected
7.0	FR_COLLISION	Verify that the system does not allow chefs to parse through each other.	ChefTests.chefCollision()	Verifies if the chefs collide the user cannot control the chef for the cooldown. After cooldown it verifies if the user gets control after.	Chefs don't parse through each other	As expected

Manual Test Results - New Requirements

<u>Test ID</u> (Prepend MN)	<u>Requirement ID</u>	<u>Description & Category</u> (E2E/Integration)	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
1.0	FR_MAIN_MENU	Verify that the mode (scenario/endless) selected is recorded by the system and therefore the mode played.	Graphical dependency, hence cannot be tested using unit tests	Load game Select scenario mode a. Select difficulty b. Check that if five orders are completed then game ends Select endless mode c. Game only ends when rep=0	Scenario mode is played when selected and endless mode is played when selected	As expected
1.1	FR_MAIN_MENU	Verify that the difficulty selected is recorded and therefore the difficulty played.	Graphical dependency, hence cannot be tested using unit tests	1. Load game 2. Select scenario mode 3. Select easy difficulty a. Game ends are 5 orders 4. Select normal difficulty a. Game ends after 8 order 5. Select hard difficulty a. Game ends after 11 orders	difficulty selected the difficulty played in scenario mode	As expected
2.0	FR_LOAD_GAME	Verify that the last game played is	Graphical dependency,	1. Load game 2. Select any mode	After saving you load back	As expected

<u>Test ID (Prepend MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
		loaded when returning to the game from the main menu.	hence cannot be tested using unit tests	3. Complete 2 orders 4. Note down time (not order time), money, dishes of current order, reputation points and order count 5. Open pause menu 6. Press on save and quit 7. Press on resume button in the main menu 8. Check if all what was noted down is the same after you load in the game	to the same game	
3.0	FR_ORDER_WAVES	Verify that 5 tickets are generated per wave for easy difficulty.	Graphical dependency, hence cannot be tested using unit tests	1. Load game 2. Select scenario mode <ul style="list-style-type: none"> d. Select easy difficulty e. Check that if five orders are completed then game ends 	5 orders per easy game in scenario mode	As expected
3.1	FR_ORDER_WAVES	Verify that 8 tickets are generated per wave for normal difficulty.	Graphical dependency, hence cannot be tested using unit tests	1. Load game 2. Select scenario mode <ul style="list-style-type: none"> f. Select normal difficulty g. Check that if 8 orders are completed then game ends 	8 orders per normal game in scenario mode	As expected
3.2	FR_ORDER_WAVES	Verify that 11 tickets are generated per	Graphical dependency, hence cannot	1. Load game 2. Select scenario mode <ul style="list-style-type: none"> a. Select hard difficulty 	11 orders per hard game in scenario mode	As expected

<u>Test ID</u> (Prepend MN)	<u>Requirement ID</u>	<u>Description & Category</u> (E2E/Integration)	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
		wave for hard difficulty.	be tested using unit tests	b. Check that if 11 orders are completed then game ends		
3.3	FR_ORDER_WAVES	Verify that new tickets only come in once all tickets of the previous wave have been completed.	Graphical dependency, hence cannot be tested using unit tests	<ol style="list-style-type: none"> 1. Load game 2. Select any mode 3. Finish 3 orders 4. Note down dishes in fourth order 5. Complete one of the dishes 6. Note down if the dish graphic is removed or not 7. Note down if another graphic is added or not 	No new dishes are added until the user finishes the order	As expected
4.0	FR_ORDER_TIME_LIMIT	Verify that recipes have different times to complete.	Graphical dependency, hence cannot be tested using unit tests	<ol style="list-style-type: none"> 1. Load game 2. Select any mode 3. Verify that Burgers, Salads and Jacket Potatoes have a 40 second counter 4. Verify that Pizza has a 60 second counter 	Each recipe's timer is correct	As expected
5.0	FR_EXCEEDED_TIME_LIMIT	Verify that the system removes an order ticket once its time limit has elapsed and the user has failed at that order.	Graphical dependency, hence cannot be tested using unit tests	<ol style="list-style-type: none"> 1. Load game <ol style="list-style-type: none"> a. Repeat steps below for all modes and difficulty 2. Wait for order time to complete and do not complete order 3. Repeat for all orders 	When order is not completed the ticket is removed	As expected

<u>Test ID (Prepend MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
6.0	FR_GAME_OVER	Verify that when the user is in scenario mode and has completed all orders with at least 1 rep point remaining, the game is won.	Graphical dependency, hence cannot be tested using unit tests	<ol style="list-style-type: none"> 1. Load game 2. Select scenario mode <ol style="list-style-type: none"> a. Repeat for all difficulties b. Do not complete order in time c. Do not complete order in time 3. Complete 5 orders in time 4. Game is won 	Game is won as long as user has at least 1 rep and completes five orders	As expected
6.2	FR_GAME_OVER	Verify that when the user is in endless mode and has lost all reputation points, the system recognises the game as having ended.	Graphical dependency, hence cannot be tested using unit tests	<ol style="list-style-type: none"> 1. Load game 2. Select endless mode 3. Do not complete order 4. Rep count decrease by 1 5. Do not complete order 6. Rep count decrease by 1 7. Do not complete order' 8. rep count decrease by 1 9. Game over screen loads 	When a total of three orders are not completed in time then rep = 0 and game ends	As expected
8.0	FR_UNLOCKABLES	Verify that content can be unlocked.	Graphical dependency, hence cannot be tested using unit tests	<ol style="list-style-type: none"> 4. Load game <ol style="list-style-type: none"> a. Repeat steps below for all modes and difficulty 5. Use mouse to click shop button 6. Select chef to buy <ol style="list-style-type: none"> a. Exit shop by clicking exit button with mouse b. One extra chef on 	When an item is bought from the shop it is then available to use in game. The padlock graphic is removed from bought item or	As expected

<u>Test ID</u> (Prepend MN)	<u>Requirement ID</u>	<u>Description & Category</u> (E2E/Integration)	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
				<p>screen</p> <p>7. Select frying station to buy</p> <p>a. Exit shop by clicking exit button with mouse</p> <p>b. One extra frying station in kitchen unlocked and can be used</p> <p>8. Select chopping station to buy</p> <p>a. Exit shop by clicking exit button with mouse</p> <p>b. One extra chopping station in kitchen unlocked and can be used</p> <p>9. Select oven station to buy</p> <p>a. Exit shop by clicking exit button with mouse</p> <p>b. One extra oven station in kitchen unlocked and can be used</p>	new chef is on screen	
9.0	FR_SPEND_MON EY	Verify that the correct price is subtracted from total earnings on spend.	Graphical dependency, hence cannot be tested using unit tests	<p>1. Load game</p> <p>a. Repeat steps below for all modes and difficulty</p> <p>2. Use mouse to click shop button</p> <p>3. Select available items to buy</p>	Total earnings count should decrease by price of item bought	As expected

<u>Test ID</u> <u>(Prepend</u> <u>MN)</u>	<u>Requirement ID</u>	<u>Description &</u> <u>Category</u> <u>(E2E/Integration)</u>	<u>Reason for</u> <u>Manual</u> <u>Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected</u> <u>Result</u>	<u>Actual</u> <u>Result</u>
				4. Total earnings decrease by item price 5. Use mouse to press exit button to exit shop		

Manual Test Results - Old Requirements

<u>Test ID</u> (Prepend MO)	<u>Requirement ID</u>	<u>Description & Category</u> (E2E/Integration)	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
1.0	FR_FINISH	Verify that the system recognises when all customers have been served.	Cannot instantiate into testing environment	<ol style="list-style-type: none"> 1. Load game 2. Select scenario mode 3. Select easy difficulty <ol style="list-style-type: none"> a. Game ends when 5 customers served 4. Select normal difficulty <ol style="list-style-type: none"> a. Game ends when 8 customers served 5. Select hard difficulty <ol style="list-style-type: none"> a. Game ends when 11 customers served 	Game ends when all customers for that difficulty have been served	As expected
1.0	FR_CONTROLS	Verify that the controls used in the system are easily understood by people new to it	Requirement is subjective	<ol style="list-style-type: none"> 1. Load game 2. Select any mode 3. Press on pause button 4. Press on the i button in the pause menu 5. Note down if the controls are displayed 6. Press on ESC to exit and continue playing 	The controls show up	As expected

<u>Test ID (Prepend MO)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
2.0	FR_SWITCH_INF ORM_USER	Verify that the user is notified when they have changed chef.	Regards user experience (being notified).	<ol style="list-style-type: none"> 1. Load game 2. Select mode <ol style="list-style-type: none"> a. Select difficulty if scenario mode 3. Press 1 <ol style="list-style-type: none"> a. 'C' is on back of chef 1 4. Press 2 <ol style="list-style-type: none"> a. 'C' is on back of chef 2 5. Press 3 <ol style="list-style-type: none"> a. 'C'; is on back of chef 6. Repeat for all modes and difficulties 	C is on back of selected chef so user knows when they change chef	As expected
3.0	FR_RP_LOSS_IN FORM_USER	Verify that the user is notified when they have lost a reputation point.	Regards user experience (being notified).	<ol style="list-style-type: none"> 1. Load game 2. Select mode <ol style="list-style-type: none"> a. Select difficulty if scenario mode 3. Fail order by not serving customer in time frame 4. Count for rep in top left of screen decreases by one for every failed order 5. Repeat for all modes and difficulties 	For every failed order rep decreases by 1 and count is updated	As expected
4.0	FR_RECIPE	Verify that the user is able to see and	Regards user experience (being	<ol style="list-style-type: none"> 1. Load game 2. Select mode 	Users know what	As expected

<u>Test ID (Prepend MO)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
		understand the recipe they are currently making.	notified) and is subjective.	<ul style="list-style-type: none"> a. Select difficulty if in scenario mode 3. When order comes in the ticket in view of the player and shown fully on screen 4. Repeat for all modes and difficulties 	recipe they are making by being able to see the recipe on top right corner of the screen	
5.0	FR_DEMO	[Not implemented].	N/A	N/A	N/A	N/A
6.0	NFR_RESPONSIVE	Verify that the system is responsive to mouse and keyboard input.	Non-functional requirement regards user experience and is subjective.	<ul style="list-style-type: none"> 1. Load game 2. Select button with mouse (select scenario button) 3. Appropriate response displayed (difficulty selection screen displayed) 	When buttons and keys are clicked the correct response is loaded	As expected
7.0	NFR_GAME_LENGTH	Verify that recipes can be prepared in a reasonable length of time.	Non-functional requirement regards user experience and is subjective.	<ul style="list-style-type: none"> 1. Load game 2. Selected mode <ul style="list-style-type: none"> a. Select difficulty if in scenario mode 3. Prepare order and time how long it takes 4. Repeat for all orders in game 		

<u>Test ID</u> (Prepend MO)	<u>Requirement ID</u>	<u>Description & Category</u> (E2E/Integration)	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
				5. Repeat for all modes and difficulties		

Manual Test Results - New Requirements

<u>Test ID</u> (Prep end MN)	<u>Requirement ID</u>	<u>Description & Category</u> (E2E/Integration)	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
1.0	FR_MAIN_MENU	Verify that the main menu is displayed and buttons are accessible. (Accompanied by automated tests that verify buttons do as they should).	Verification of user interface and user's experience.	1. Load game	Main menu and buttons are clearly shown when game is loaded	As expected
2.0	FR_DIFFICULTY	Verify that when playing in easy mode, tickets predominantly come in waves of 1 and 2, and that these tickets are only for salads and burgers.	Requirement is loosely defined (e.g., 'predominantly').	1. Load game 2. Select scenario mode 3. Select easy mode 4. Play round of easy mode by completing 5 orders	Tickets predominantly come in waves of 1 and 2, and that these tickets are only for salads and burgers	As expected
2.1	FR_DIFFICULTY	Verify that when playing in normal mode, tickets quickly progress to 2 and then have a few waves with 3.	Requirement is loosely defined (e.g., 'quickly', 'a few').	1. Load game 2. Select scenario mode 3. Select normal mode 4. After completing 4 orders the tickets come in waves of 3	tickets quickly progress to 2 and then have a few waves with 3	As expected
2.2	FR_DIFFICULTY	Verify that when playing in hard mode, tickets mainly come in	Requirement is loosely defined	1. Load game 2. Select scenario mode	tickets mainly come in waves	As expected

<u>Test ID (Prep end MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
		waves of 3 but some of 2.	(e.g., 'mainly', 'but some').	3. Select hard mode 4. After completing 4 orders the tickets come in waves of 3 till the end of the scenario	of 3 but some of 2	
2.3	FR_DIFFICULTY	Verify that when playing in endless mode, the time limit to serve all dishes decreases as the game progresses.	Requirement regards the gameplay as a whole over some time.	[not implemented]	N/A	N/A
3.0	FR_ORDER_TIME_LIMIT	Verify that a time limit is shown and updated beneath tickets.	Requirement regards UI/UX.	1. Load game 2. Select game mode <ol style="list-style-type: none"> Select difficulty if in scenario mode 3. When ticket comes in check in hud that timer is updating while completing the order 4. Repeat for all modes and difficulties	Timer for ticket decreases as user plays games	As expected
4.0	FR_GAME_OVER	Verify that when the user is in scenario mode and has lost (automatically tested), he/she is notified that they have lost.	Requirement regards UI/UX.	[not implemented]	N/A	N/A

<u>Test ID (Prep end MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
4.1	FR_GAME_OVER	Verify that when the user is in scenario mode and has run out of reputation points, he/she is shown the number of customers served, his/her score and his/her earnings from that particular game.	Requirement regards UI/UX.	<ol style="list-style-type: none"> 1. Load game 2. Select scenario mode 3. Lose game by not completing orders in time 4. Check what is shown on the screen and if it is correct for the game played 	when the user is in scenario mode and has run out of reputation points, he/she is shown the number of customers served, his/her score and his/her earnings from that particular game	As expected
5.0	FR_CUSTOMERS_SERVED	Verify that in endless mode, there is a counter - that increments - of the number of customers served.	Requirement regards the user experience and the logic is trivial ($x \leftarrow x+1$).	[Not implemented].	N/A	N/A
6.0	FR_MONEY	Verify that total earnings are displayed during gameplay and at the end.	Requirement regards UI/UX.	<ol style="list-style-type: none"> 1. Load game 2. Select Scenario mode 3. Select Easy mode 4. Complete orders and note down the earning after each order 5. After finishing 5 orders 	500 if 5 orders were completed	As expected

<u>Test ID (Prep end MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
				check if what you noted is the same as what is displayed at the end		
7.0	FR_UNLOCKABLES	Verify that unlockable content is shown on the main game map, is greyed out when locked and shows the respective price below.	Requirement regards UI/UX.	<ol style="list-style-type: none"> 1. Load game 2. Select endless mode 3. Note down how many Pans, Chopping boards and Ovens are locked (lock graphic is displayed on them). 	2 Pans locked, 2 Chopping boards locked and 1 Oven locked	As expected
8.0	FR_SPEND_MONEY	Verify that unlockable content is in colour and accessible once unlocked.	Requirement regards UI/UX.	<ol style="list-style-type: none"> 1. Load game 2. Select endless mode 3. Complete 5 orders and note down how much money you earned 4. Open the shop menu 5. Note down price of a station/chef 6. Buy said station/chef 7. Check if money = oldmoney - price 	After buying a chef/station your money should decrease by the price of the item bought	As expected
9.0	FR_ACTION_TIME_LIMIT	Verify that a progress bar is shown and animates above a chefs head when performing an action (e.g., cooking, preparing).	Requirement regards UI/UX.	<ol style="list-style-type: none"> 1. Load game 2. Select game mode <ol style="list-style-type: none"> a. Select difficulty if in scenario 	progress bar is shown and animates above a chefs	As expected

<u>Test ID (Prep end MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
				mode 3. Move chef to pick up ingredient 4. Press e to pick up ingredient 5. Move chef to station to [perform action 6. Press e to perform action 7. Check for progress bar above chef <ol style="list-style-type: none"> Repeat for all modes and difficulties Repeat for all ingredients and stations 	head when performing an action	
9.1	FR_ACTION_TIME_LIMIT	Verify that a chef performing an action can be selected but cannot move.	Requirement regards UI/UX.			
10.0	FR_QTE (Quick Time Event)	Verify that when a chef is at a baking/frying, a small progress bar is shown.	Requirement regards UI/UX.	1. Load game 2. Select any mode 3. Pick up tomato using chef 4. Move chef to chopping board and press E	Progress bar is created after chef starts interacting with a station	As expected

<u>Test ID (Prep end MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
				5. Note down if progress bar is created or not		
10.1	FR_QTE	Verify that the system starts a QTE when the correct chef (carrying apt ingredient) arrives at a station and the action key is pressed.	Requirement regards UI/UX.	<ol style="list-style-type: none"> 1. Load game 2. Select any mode 3. Pick up tomato using chef 4. Move chef to chopping board and press E 5. After the green progress bar, note down if a red progress bar is shown. Press on Q and note down if the tomato was prepared properly. 	Red progress bar appears after the green one, if you press Q before the red progress bar is over the ingredient would be prepared	The ingredient is always prepared if you press Q within 3 seconds after the green bar. However the red progress bar doesn't always appear.
11.0	FR_PREP_FAILED	Verify that if the QTE isn't completed in time, the user is notified of this and told that why must bin the item and start over.	Requirement regards UI/UX.	[Not implemented]	N/A	N/A
12.0	FR_INGRED	Verify that the system provides	Requirement	[Not implemented]	N/A	N/A

<u>Test ID (Prep end MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
	IENTS	selectable ingredients for salads, burgers, pizzas and jacket potatoes.	regards UI/UX.			
12.1	FR_INGREDIENTS	Verify that the system does not display orders for pizzas/jacket potatoes until the baking station has been unlocked.	Requirement regards UI/UX.	[Not implemented]	N/A	N/A
17.0	FR_POWERUPS	Verify that powerups are displayed in the HUD according to the requirement and that they are effective.	Requirement regards UI/UX	<ol style="list-style-type: none"> 1. Load game 2. Select any mode 3. Complete 2 orders 4. Note down if a powerup appeared in the HUD 5. Note down if the powerup is acting appropriately 	After 2 orders are complete a power up is generated and its functioning properly	As expected
18.0	FR_PAUSE_MENU	Verify that the game can be paused and that controls, resume and save & exit buttons are accessible and functional.	Requirement regards UI/UX	<ol style="list-style-type: none"> 1. Load game 2. Select any mode 3. Noted down time, order dishes and order count 4. Press on pause button 5. Unpause using the resume button or X button 6. Note down if the time, 	The game stays the same before and after pausing	As expected

<u>Test ID (Prepend MN)</u>	<u>Requirement ID</u>	<u>Description & Category (E2E/Integration)</u>	<u>Reason for Manual Testing</u>	<u>Input Data/Steps to be Followed</u>	<u>Expected Result</u>	<u>Actual Result</u>
				<p>order and order time is the same as before pausing</p> <p>7. Pause again and press on the save and quit</p>		
19.0	FR_SAVE_STATE	[Verified in AN2.0].	N/A	N/A	N/A	N/A
20.0	FR_AUTO_SAVE	[Not implemented].	N/A	N/A	N/A	N/A
21.0	FR_DEMO	[Not implemented].	N/A	N/A	N/A	N/A

