

## PSP NTP TIME SYNC APIs

### CFE\_PSP\_TIME\_Init

<b>Syntax</b>	int32 CFE_PSP_TIME_Init(void)
<b>Description</b>	Initialize the CFE PSP Time Task synchronizing with the NTP server  This function intializes the cFE PSP Time sync task with the NTP server.
<b>Parameters</b>	None
<b>Returns</b>	CFE_PSP_SUCCESS CFE_PSP_ERROR
<b>Notes</b>	None

### CFE\_PSP\_Sync\_From\_OS\_Enable

<b>Syntax</b>	int32 CFE_PSP_Sync_From_OS_Enable(bool enable)
<b>Description</b>	Enable/disable time sync  This function sets the enabling/disabling of time sync.
<b>Parameters</b>	[in] enable - Boolean flag for sync or not sync
<b>Returns</b>	True - If synchronized False - If not synchronized
<b>Notes</b>	When the flag is true, the NTP Sync task actively tries to sync clocks. When the flag is false, the NTP Sync task will remain active without sync.

### CFE\_PSP\_NTP\_Daemon\_Get\_Status

<b>Syntax</b>	bool CFE_PSP_NTP_Daemon_Get_Status(void)
<b>Description</b>	<p>Get the NTP daemon status</p> <p>This function checks if the VxWorks NTP client task is running. It does not check if the task has successfully synchronized with an NTP server.</p>
<b>Parameters</b>	None
<b>Returns</b>	<p>True - If NTP client task is running</p> <p>False - If NTP client task is not running</p>
<b>Notes</b>	The task name for the VxWorks NTP client is the default "ipntpd".

### **net\_clock\_vxworks\_Destroy**

<b>Syntax</b>	int32 net_clock_vxworks_Destroy(void)
<b>Description</b>	<p>Gracefully shutdown NTP Sync Module</p> <p>Function will attempt to delete the task. Usually this function will be called when exiting cFS.</p>
<b>Parameters</b>	None
<b>Returns</b>	<p>CFE_PSP_SUCCESS</p> <p>CFE_PSP_ERROR</p>
<b>Notes</b>	The task name for the VxWorks NTP client is the default "ipntpd".

### **CFE\_PSP\_Sync\_From\_OS\_GetFreq**

<b>Syntax</b>	uint16 CFE_PSP_Sync_From_OS_GetFreq(void)
<b>Description</b>	Get the currently set sync frequency

	This function returns the NTP time synchronization frequency, in seconds.
<b>Parameters</b>	None
<b>Returns</b>	Current frequency
<b>Notes</b>	None

### CFE\_PSP\_Sync\_From\_OS\_SetFreq

<b>Syntax</b>	int32 CFE_PSP_Sync_From_OS_SetFreq(uint16 new_frequency_sec)
<b>Description</b>	Change the sync frequency  This function updates the NTP time synchronization frequency, in seconds.
<b>Parameters</b>	[in] new_frequency_sec - The new frequency, in seconds
<b>Returns</b>	CFE_PSP_SUCCESS - If successfully changed CFE_PSP_ERROR
<b>Notes</b>	None

### CFE\_PSP\_Set\_OS\_Time

<b>Syntax</b>	int32 CFE_PSP_Set_OS_Time(const uint32 ts_sec, const uint32 ts_nsec)
<b>Description</b>	Set the OS time  This function sets the VxWorks OS time.
<b>Parameters</b>	[in] ts_sec - Time in seconds [in] ts_nsec - Time in nanoseconds
<b>Returns</b>	CFE_PSP_SUCCESS

	CFE_PSP_ERROR
<b>Notes</b>	The changes do not occur if the NTP client is setup to synchronize with an NTP server. Set the OS CLOCK_REALTIME to a specified timestamp. Parameters are in UNIX time format, since Epoch 1/1/1970.

### CFE\_PSP\_Get\_OS\_Time

<b>Syntax</b>	int32 CFE_PSP_Get_OS_Time(CFE_TIME_SysTime_t *myT)
<b>Description</b>	Gets the current time from VxWorks OS  This function gets the current VxWorks OS time.
<b>Parameters</b>	[out] myT - Pointer to the variable that stores the returned time value None
<b>Returns</b>	CFE_PSP_SUCCESS CFE_PSP_ERROR true - CFE Time Service is ready false - CFE Time Service is not ready
<b>Notes</b>	This function is used by the NTP Sync task to grab the current OS time. It uses CLOCK_REALTIME. NTP Sync will not occur if NTP time is less than CFE_MISSION_TIME_EPOCH_UNIX_DIFF None

### CFE\_PSP\_Update\_OS\_Time

<b>Syntax</b>	void CFE_PSP_Update_OS_Time(void)
<b>Description</b>	Update cFE time  This function updates the time used by the cFE Time service.
<b>Parameters</b>	None

<b>Returns</b>	None
<b>Notes</b>	This method is run on an independent thread and will continue to run until the thread is deleted using net_clock_linux_destroy

### CFE\_PSP\_StartNTPDaemon

<b>Syntax</b>	int32 CFE_PSP_StartNTPDaemon(void)
<b>Description</b>	Start the NTP client  This function starts the NTP client task, ipntpd, on VxWorks.
<b>Parameters</b>	None
<b>Returns</b>	NTP client Task ID CFE_PSP_ERROR
<b>Notes</b>	None

### CFE\_PSP\_StopNTPDaemon

<b>Syntax</b>	int32 CFE_PSP_StopNTPDaemon(void)
<b>Description</b>	Stop the NTP client  This function stops the NTP client task, ipntpd, on VxWorks.
<b>Parameters</b>	None
<b>Returns</b>	CFE_PSP_SUCCESS CFE_PSP_ERROR
<b>Notes</b>	None

## CFE\_PSP\_NTP\_Daemon\_Enable

<b><i>Syntax</i></b>	int32 CFE_PSP_NTP_Daemon_Enable(bool enable)
<b><i>Description</i></b>	<p>Enable/disable the NTP client</p> <p>This function enables/disables the NTP client task, ipntpd, on VxWorks.</p>
<b><i>Parameters</i></b>	[in] enable - Boolean flag for enable or disable
<b><i>Returns</i></b>	<p>NTP client task ID - If successfully starts the NTP clien task</p> <p>CFE_PSP_SUCCESS - If successfully stops the NTP client task</p> <p>CFE_PSP_ERROR</p>
<b><i>Notes</i></b>	None