

Pay Periods Time Tracker Implementation Guide

Version 1.1.1 | Created: August 12, 2025 PST

Author: Philippe Addelia | **Company:** NAKUPUNA CONSULTING

Overview

This guide helps you implement the enhanced time tracker with pay period management and persistence features.

Files Included

Core Files

1. `time_tracker_pay_periods.html` - Enhanced employee time tracker with pay period support
2. `pay-periods-config.json` - Configuration file containing pay period definitions
3. `Generate-PayPeriodsConfig.ps1` - PowerShell script to create initial config
4. `Update-PayPeriodsConfig.ps1` - PowerShell script to maintain config file

File Locations

```
TimeTracker/  
├── time_tracker_pay_periods.html  
├── pay-periods-config.json  
├── Scripts/  
│   ├── Generate-PayPeriodsConfig.ps1  
│   └── Update-PayPeriodsConfig.ps1  
└── Documentation/  
    └── Pay Periods Implementation Guide.md
```

Step-by-Step Implementation

Step 1: Create Pay Periods Configuration

Option A: Use the Provided Configuration

- Use the sample `pay-periods-config.json` file
- Modify dates to match your actual pay periods
- Place in the same folder as the HTML file

Option B: Generate Custom Configuration

```
powershell
```

```
# Navigate to the Scripts folder
```

```
cd "C:\TimeTracker\Scripts"
```

```
# Generate config for 26 pay periods starting August 2, 2025
```

```
.\Generate-PayPeriodsConfig.ps1 -StartDate "2025-08-02" -NumberOfPeriods 26 -OutputPath "../pay-periods-config.js"
```

Step 2: Deploy the Enhanced Time Tracker

1. Backup Current Implementation

```
powershell
```

```
# Create backup of existing files
```

```
Copy-Item "time_tracker.html" "time_tracker_backup_$(Get-Date -Format 'yyyyMMdd').html"
```

2. Deploy New Version

- Replace existing time tracker with `time_tracker_pay_periods.html`
- Ensure `pay-periods-config.json` is in the same directory
- Test configuration loading

Step 3: Verify Configuration Loading

1. Open the Enhanced Time Tracker

- Load `time_tracker_pay_periods.html` in browser
- Check the "Pay Period Selection" section
- Verify green status message: "Pay periods loaded successfully"

2. Test Pay Period Selection

- Dropdown should show all configured pay periods
- Current pay period should auto-select
- Period start/end dates should populate automatically

Step 4: Configure Persistence

The enhanced tracker automatically persists:

- Selected pay period across browser restarts
- Employee name and daily target
- All time tracking data

No additional configuration required - persistence works out of the box.

Key Features Added

1. Pay Period Management

- **Automatic Detection:** Current pay period auto-selected based on today's date
- **Dropdown Selection:** Easy switching between pay periods
- **Period Information:** Shows timesheet due date, pay day, working days
- **Persistence:** Selected pay period remembered across sessions

2. Enhanced User Interface

- **Pay Period Info Panel:** Displays due dates and working days remaining
- **Configuration Status:** Visual indicator of config file loading status
- **Manual Fallback:** Option to use manual date selection if config fails

3. Improved Data Export

- **Pay Period Context:** Exports include pay period information
- **Better Filenames:** Uses pay period ID in export filenames
- **Enhanced Summary:** Additional metadata in CSV exports

Configuration File Format

```
json
{
  "payPeriods": [
    {
      "id": "2025-16",
      "periodStart": "2025-08-02",
      "periodEnd": "2025-08-15",
      "timesheetDue": "2025-08-15",
      "payDay": "2025-08-22",
      "description": "Pay Period 16 - Aug 2-15, 2025"
    }
  ],
  "configVersion": "1.1.1",
  "lastUpdated": "2025-08-12",
  "defaultPeriodType": "current"
}
```

Field Definitions

- **id**: Unique identifier (format: YYYY-NN)
- **periodStart**: First day of pay period
- **periodEnd**: Last day of pay period
- **timesheetDue**: When timesheet must be submitted
- **payDay**: When employees receive payment
- **description**: Human-readable period description

Maintenance Operations

Adding New Pay Periods

```
powershell

# Add 4 new pay periods to existing config
.\Update-PayPeriodsConfig.ps1 -Action Add -AddPeriods 4
```

Updating Pay Dates

```
powershell

# Change pay day for specific period
.\Update-PayPeriodsConfig.ps1 -Action Update -PeriodId "2025-16" -NewDate "2025-08-23" -DateType payDay
```

Viewing Configuration

```
powershell

# Display current config summary
.\Update-PayPeriodsConfig.ps1 -Action Show
```

Troubleshooting

Configuration Not Loading

Symptoms:

- Red error message in Pay Period Selection
- Dropdown shows "Loading pay periods..."
- Manual date selection is enabled

Solutions:

1. Check File Location

Ensure pay-periods-config.json is in same folder as HTML file

2. Validate JSON Format

```
powershell  
  
# Test JSON validity  
Get-Content "pay-periods-config.json" | ConvertFrom-Json
```

3. Check Browser Console

F12 > Console tab > Look for fetch errors

Period Not Auto-Selecting

Symptoms:

- Config loads but no period is selected
- Today's date doesn't fall in any period

Solutions:

1. Check Date Ranges

```
powershell  
  
# Verify period coverage  
.\Update-PayPeriodsConfig.ps1 -Action Show
```

2. Add Missing Periods

```
powershell  
  
# Add periods to cover current date  
.\Update-PayPeriodsConfig.ps1 -Action Add -AddPeriods 2
```

Data Not Persisting

Symptoms:

- Pay period selection resets on restart
- Employee info gets cleared

Solutions:

1. Check Browser Storage

F12 > Application > Local Storage > Verify timeTrackerData exists

2. Clear Browser Cache

Try hard refresh (Ctrl+Shift+R) or incognito mode

Migration from Existing Tracker

Data Preservation

The enhanced tracker is compatible with existing time tracking data. No data migration is required.

Settings Transfer

1. **Employee Information:** Automatically preserved
2. **Time Entries:** All existing entries remain accessible
3. **Preferences:** Daily targets and other settings maintained

Backward Compatibility

- Original time tracker files remain functional
- Can run both versions simultaneously for testing
- Export formats remain compatible

Best Practices

1. Regular Config Updates

- Update pay periods quarterly
- Backup config before major changes
- Test changes with `-WhatIf` parameter

2. File Management

- Keep config file in same directory as HTML
- Use version control for config changes
- Document customizations

3. Employee Training

- Show employees new pay period dropdown
- Explain automatic period selection
- Demonstrate improved export features

Advanced Customization

Custom Pay Day Calculations

Modify the PowerShell scripts to handle:

- Different pay day offsets per period
- Holiday pay day adjustments
- Month-end pay schedules

Additional Metadata

Extend the config format to include:

- Payroll processing deadlines
- Holiday calendars
- Project-specific billing periods

Support and Maintenance

Regular Tasks

- **Monthly:** Add new pay periods for upcoming months
- **Quarterly:** Review and update pay schedules
- **Annually:** Archive old periods and start new year

Monitoring

- Check config loading status in employee trackers
- Monitor export file naming for accuracy
- Verify period auto-selection behavior

Version History

v1.1.1 - August 12, 2025

- **Added:** Pay period configuration system
- **Enhanced:** Automatic period detection and selection

- **Improved:** Data persistence across browser sessions
- **Fixed:** Period start/end confusion on app restart

Next Planned Features

- **Admin Console Integration:** Centralized pay period management
 - **Holiday Calendar:** Automatic working days calculation
 - **Notification System:** Alerts for timesheet due dates
-

Technical Notes

Browser Compatibility

- **Minimum:** Chrome 60+, Firefox 55+, Safari 12+, Edge 79+
- **Features:** Uses Fetch API, localStorage, modern JavaScript
- **Fallback:** Manual date selection for config loading failures

Performance Considerations

- Config file size should remain under 100KB
- Recommended maximum: 104 pay periods (2 years)
- Browser localStorage limit: ~5-10MB total

Security Notes

- Configuration file contains no sensitive data
 - All processing happens client-side
 - No external API dependencies
-

For additional support or customization requests, contact the development team.