

# AAR System Installation Guide

## Quick Fix for the submit\_task Error

### Step 1: Create Required Directories

Run this in your `aar_system` directory:

```
bash

mkdir -p engines/activity
mkdir -p engines/equipment
mkdir -p engines/environmental

# Create __init__.py files
echo "# Analysis engine module" > engines/activity/__init__.py
echo "# Analysis engine module" > engines/equipment/__init__.py
echo "# Analysis engine module" > engines/environmental/__init__.py
```

### Step 2: Copy Analysis Engine Files

Copy these artifact contents to the specified files:

#### 1. Soldier Activity Engine

- Source: `soldier_activity_engine` artifact
- Target: `engines/activity/soldier_activity_engine.py`

#### 2. Equipment Management Engine

- Source: `equipment_management_engine` artifact
- Target: `engines/equipment/equipment_management_engine.py`

#### 3. Environmental Monitoring Engine

- Source: `environmental_monitoring_engine` artifact
- Target: `engines/environmental/environmental_monitoring_engine.py`

### Step 3: Update Core Files

#### 4. Analysis Orchestrator (FIXES THE ERROR)

- Source: `updated_analysis_orchestrator` artifact
- Target: `services/orchestration/analysis_orchestrator.py`

#### 5. Main Application





- Source: `updated_main_application` artifact
- Target: `ui/main_application.py`

## Step 4: Test the Installation

```
bash
```

```
python main.py
```

## What This Fixes

-  **Fixes** `submit_task` **error** - New orchestrator has all required methods
-  **Adds 3 new analysis engines** - Activity, Equipment, Environmental
-  **Complete requirements coverage** - All REQ-\* specifications implemented
-  **Enhanced UI** - Analysis control tab and better integration

## File Structure After Installation

```
aar_system/
├── core/
│   ├── event_bus.py
│   └── models.py
├── engines/
│   ├── safety/
│   │   └── soldier_safety_engine.py
│   ├── network/
│   │   └── network_performance_engine.py
│   ├── activity/
│   │   └── soldier_activity_engine.py    # NEW
│   ├── equipment/
│   │   └── equipment_management_engine.py # NEW
│   └── environmental/
│       └── environmental_monitoring_engine.py # NEW
├── services/
│   ├── orchestration/
│   │   └── analysis_orchestrator.py    # UPDATED
│   └── reporting/
│       └── report_service.py
└── ui/
    ├── main_application.py            # UPDATED
    └── components/
        ├── data_management_tab.py
        ├── analysis_results_tab.py
        └── reports_tab.py
```

## Expected Output After Fix

When you run `python main.py`, you should see:

```
[SUCCESS] Activity Analysis Engine registered
[SUCCESS] Equipment Management Engine registered
[SUCCESS] Environmental Monitoring Engine registered
Total registered engines: 5
- safety
- network
- activity
- equipment
- environmental
Starting AAR System...
```

# Troubleshooting

## If you still get import errors:

1. Make sure all `__init__.py` files exist in the engine directories
2. Check that the file paths match exactly
3. Verify the Python files have proper indentation

## If you get Unicode errors:

- Use the fixed setup script (without emojis)
- Make sure your editor saves files in UTF-8 encoding

## If analysis fails:

- Check the Analysis Control tab for engine status
- Look at the console output for detailed error messages
- Use the test script to verify individual engines work