

# AI Dependency Management in Update Script

---

## Overview

---

The `update_from_github.sh` script now includes integrated AI dependency management, making it seamless to keep your AI Code Assistant up-to-date alongside the main application.

## What's Integrated

---

### Automatic AI Checks

When you run `./update_from_github.sh`, the script now automatically:

**1. Checks Ollama Installation**

- Verifies if Ollama is installed
- Offers to install if missing
- Shows current version if installed

**2. Verifies Ollama Service**

- Checks if Ollama service is running
- Attempts to start it if stopped
- Uses systemd or background process as appropriate

**3. Checks AI Models**

- Verifies availability of required models:
  - `llama3.2` - Primary model for style analysis
  - `llama2` - Backup model for device diagnostics
  - `mistral` - Fast model for quick queries
  - `deepseek-coder:6.7b` - AI Code Assistant model
  - Prompts to download missing models (with 10-second timeout)
  - Shows list of all available models

**4. Runs Dependency Verification**

- Executes `ai-assistant/check-dependencies.js`
- Verifies Node.js version
- Checks npm dependencies
- Tests model generation capability
- Validates directory structure

## Usage

---

### Standard Update (with AI checks)

```
./update_from_github.sh
```

This runs the full update process including AI dependency checks.

## Fast Update (skip AI checks)

```
./update_from_github.sh --skip-ai
```

Use this flag when:

- You don't use AI features
- You want a faster update
- You've recently verified AI dependencies
- You're troubleshooting non-AI issues

## Help

```
./update_from_github.sh --help
```

Shows usage information and available options.

## Features

### Non-Breaking Design

- **AI setup failures don't stop updates** - The application will still update even if AI setup encounters issues
- **Clear warnings** - You'll see warnings about AI limitations but the update continues
- **Graceful degradation** - Application works without AI features if setup fails

### User-Friendly Prompts

When a model is missing, you'll see:

```
📦 Checking AI models...
⚠️ llama3.2 - not found
Would you like to download llama3.2? (y/N) [10s timeout]
```

- **10-second timeout** - Automatically skips if no response
- **Manual download option** - Shows command to download later
- **No interruption** - Update continues regardless of choice

## Comprehensive Logging




All AI setup activities are logged to `update.log` :

- Installation attempts
- Service status changes
- Model downloads
- Verification results

## What Gets Checked

### System Requirements

- ☒ Ollama binary installed
- ☒ Ollama service running
- ☒ Required AI models available

-  Node.js version (18+)
-  npm dependencies installed
-  Directory structure correct





AI Models

The script checks for these models:

Model	Purpose	Size
llama3.2	Style analysis, AI features	~2GB
llama2	Device diagnostics backup	~3.8GB
mistral	Fast queries	~4.1GB
deepseek-coder:6.7b	AI Code Assistant	~3.8GB




Integration with Existing Workflow

The AI checks are seamlessly integrated into the existing update workflow:

```
1. Passwordless sudo check
2. PM2 installation check
3. Backup configuration  database
4. Git status check
5. Stop server
6. Pull latest changes
7. Restore channel presets
8. Initialize data files
9. Check local configuration
10. Smart dependency installation
11. libCEC installation check
12.  AI dependencies setup  NEW 
13. Environment variables check
14. Database update
15. Build application
16. AI style analysis (background)
17. PM2 startup configuration
18. Restart server
```

Status Messages

Success Messages

```
 Ollama already installed (version: 0.1.17)
 Ollama service is running
 llama3.2 - available
 AI dependency check passed
 AI dependencies verified successfully
```

## Warning Messages

- ⚠️ Ollama installation failed - AI features will be limited
- ⚠️ Could not start Ollama service - AI features will be limited
- ⚠️ AI dependency check reported issues
- ⚠️ AI setup completed with warnings

## Info Messages

- ℹ️ AI dependency check script not found (ai-assistant/check-dependencies.js)
- 🔧 Skipping AI dependency checks (--skip-ai flag set)

## Troubleshooting

### AI Setup Failed

If AI setup fails during update:

1. **Check the logs:**

```
bash
tail -100 update.log | grep -A 5 -B 5 "AI"
```

2. **Run manual verification:**

```
bash
node ai-assistant/check-dependencies.js
```

3. **Install Ollama manually:**

```
bash
curl -fsSL https://ollama.com/install.sh | sh
```

4. **Download models manually:**

```
bash
ollama pull llama3.2
ollama pull deepseek-coder:6.7b
```

### Ollama Service Won't Start

Try these steps:

1. **Check if already running:**

```
bash
pgrep -x ollama
```

2. **Start manually:**

```
bash
ollama serve &
```

3. **Check systemd status** (if available):

```
bash
systemctl status ollama
sudo systemctl start ollama
```

## Models Won't Download

If model downloads fail:

1. **Check internet connection**
2. **Check disk space:** Models are 2-4GB each
3. **Try downloading individually:**

```
bash
```

```
ollama pull llama3.2
```

## Performance Impact

---

### With AI Checks (default)

- **First run:** +5-10 minutes (model downloads)
- **Subsequent runs:** +10-30 seconds (verification only)
- **With all models cached:** +5-10 seconds

### With `-skip-ai` Flag

- **Any run:** +0 seconds (no AI checks)

## Best Practices

---

### When to Use `-skip-ai`

#### ✅ Use `-skip-ai` when:

- Doing quick bug fixes
- Testing non-AI features
- Running multiple updates in succession
- Troubleshooting application issues
- Working in CI/CD pipelines

#### ❌ Don't use `-skip-ai` when:

- First time setup
- After long periods without updates
- When AI features aren't working
- After Ollama updates
- When models might be outdated

## Recommended Workflow

1. **Regular updates** (weekly):

```
bash
```

```
./update_from_github.sh
```

2. **Quick updates** (multiple per day):

```
bash
```

```
./update_from_github.sh --skip-ai
```

3. **Periodic AI verification** (monthly):

```
bash
```

```
node ai-assistant/check-dependencies.js
```

## Future Enhancements

---

Planned improvements:

- [ ] Automatic model updates when new versions available
- [ ] Model size optimization recommendations
- [ ] GPU acceleration detection and setup
- [ ] Model performance benchmarking
- [ ] Automatic cleanup of old model versions
- [ ] Integration with model registry for version tracking

## Related Documentation

---

- [AI Code Assistant README](#) (./ai-assistant/README.md)
- [AI Code Assistant Deployment Guide](#) (./ai-assistant/DEPLOYMENT.md)
- [Update Process Documentation](#) (./UPDATE\_PROCESS.md)
- [Update Script Guide](#) (./UPDATE\_SCRIPT\_GUIDE.md)

## Support

---

If you encounter issues with AI integration:

1. Check this documentation
2. Review `update.log` for errors
3. Run `node ai-assistant/check-dependencies.js`
4. Open an issue on GitHub with logs

---

**Last Updated:** October 6, 2025

**Version:** 1.0.0

**Related PR:** #91