

# Teeth-Management-System Installer Documentation (install.sh)

**Audience:** Developers deploying or maintaining the script on Linux servers

**Shell:** `bash` with `set -euo pipefail`

**Purpose:** Provide a one-shot setup for the *Teeth-Management-System* development environment, installing Python, venv, Node (apt systems), Oracle XE prerequisites, and the `astart` launcher.

---

## 1. Overview

The installer script (`install.sh`) is a **Bash-based automation tool** intended to prepare a Linux development environment for the Teeth-Management-System project.

### Supported environments

- Linux distributions with one of these package managers:
  - `apt-get`
  - `dnf`
  - `yum`
  - `pacman`
  - `zypper`
- Requires Bash (`#!/usr/bin/env bash`)
- Designed for **development setups**, not hardened production installation.

### High-level goals

- Ensure Python 3, pip, and project venv.
  - Install Python dependencies from `requirements.txt`.
  - (APT systems only) Optionally install Node.js via NodeSource.
  - Detect or install **Oracle XE** plus prerequisite swap/kernel/limits configs.
  - Deploy a global executable (`/bin/astart`) for starting the app.
- 

## 2. Current Behavior (Step-by-Step Execution)

The script's main flow currently runs in this order:

### **\*\*1. \*\*** `ensure_oracle_prereqs`

- Detects Oracle by:
- Checking for `oracle` system user

- Or presence of `/etc/oratab`
- If not detected:
- Downloads **Oracle XE RPM**
- On `apt` systems: converts RPM → DEB using `alien`
- Installs XE package
- Ensures **4GB swap**
- Installs OS dependencies: `libaio`, `bc`, `sysstat`, `unixodbc`, etc.
- Appends Oracle-required:
- `/etc/security/limits.conf` entries
- `/etc/sysctl.conf` kernel parameters
- Applies kernel settings via `sysctl -p`

## **\*\*2.\*\*** `ensure_python_pip`

- Installs `python3` and `python3-pip` via detected package manager.
- If pip missing but Python exists:
- Attempts `python3 -m ensurepip` fallback.

## **\*\*3.\*\*** `create_venv`

- Creates virtual environment under `Ai-chatbot/venv` using `python3 -m venv`.

## **\*\*4.\*\*** `install_requirements`

- Activates venv if it exists.
- Installs from `Ai-chatbot/requirements.txt`.
- If venv missing:
- Attempts to recreate venv.
- Fallback: installs packages globally.

## **\*\*5.\*\*** `install_node_system_wide` **(APT only)**

- NodeSource setup for Node 18/20:
- Adds NodeSource repo
- Installs `nodejs`
- Performs conflict cleanup:
- Removes `/usr/include/node/common.gypi`
- Removes `libnode-dev`, `nodejs`, `npm`

## **\*\*6.\*\*** `install_astart`

- Copies launcher from `$SCRIPT_DIR/astart` → `/bin/astart`.
- Overwrites existing file.
- Sets `chmod +x`.
- Warns if missing.

### 3. Functions Summary

Function	Purpose
<code>detect_pkg_mgr</code>	Detects system package manager.
<code>install_pkgs</code>	Installs packages using detected package manager.
<code>ensure_python_pip</code>	Ensures Python 3 + pip installation.
<code>create_venv</code>	Creates venv at <code>Ai-chatbot/venv</code> .
<code>install_requirements</code>	Installs project dependencies using venv.
<code>install_node_system_wide</code>	NodeSource install (APT only) + conflict cleanup.
<code>ensure_oracle_prereqs</code>	Detects/installs Oracle XE and configs.
<code>install_astart</code>	Installs launcher to <code>/bin/astart</code> .
<code>main</code>	Full script execution flow.

### 4. Variables and Paths

Variable	Description
<code>SCRIPT_DIR</code>	Directory containing <code>install.sh</code> .
<code>AI_DIR</code>	<code>\$SCRIPT_DIR/Ai-chatbot</code> .
<code>REQ_FILE</code>	<code>\$AI_DIR/requirements.txt</code> .
<code>VENV_DIR</code>	<code>\$AI_DIR/venv</code> .
<code>ASTART</code>	<code>\$SCRIPT_DIR/astart</code> .
<b>Launcher Destination</b>	<code>/bin/astart</code> .

### 5. Node Install Details (APT Systems)

#### NodeSource flow

1. Removes conflicting files/packages:
2. `/usr/include/node/common.gypi`
3. `libnode-dev`, `nodejs`, `npm`
4. Adds NodeSource repo for **Node 18/20**.

5. Installs `nodejs` via `apt-get install`.

### Notes

- Only runs on systems using `apt-get`.
  - No equivalent steps exist for `dnf`, `yum`, `pacman`, or `zypper`.
- 

## 6. Oracle Prerequisites & Install Logic

### Detection

Oracle is considered installed if:

- User `oracle` exists **or**
- File `/etc/oratab` exists

### If not detected

1. Downloads Oracle XE RPM.
2. Converts to DEB using `alien` on apt-based systems.
3. Installs the resulting package.
4. Installs Oracle-required dependencies.
5. Configures limits + kernel params.
6. Applies settings with `sysctl -p`.

### Swap

- Ensures presence of a **4GB swapfile**.
- 

## 7. Launcher Install (\*\*`astart`\*\*)

### Current behavior

- Copies from `$SCRIPT_DIR/astart` → `/bin/astart`.
- Overwrites existing file.
- Sets executable bit.
- Warns if missing.

### Notes

Earlier script versions:

- Had fallbacks using `Ai-chatbot/astart`.
- At times generated launcher code dynamically.

Current version: **copy-only**.

---

## 8. Evolution Timeline

### v1 – Initial minimal installer

- Ensured Python3, pip, venv.
  - Installed requirements.
  - Copied launcher.
- 

### Node Option B added

- Introduced apt-only NodeSource installation.
  - Added dpkg conflict fixes.
- 

### Refactors & logging improvements

- Added color helpers ( `msg` , `ok` , `warn` , `err` ).
  - Expanded inline comments.
- 

### Venv-first install logic

- Prioritizes using venv before falling back to system pip.
- 

### Launcher logic evolution

1. Basic copy
  2. Generated launcher when missing
  3. Fallback to Ai-chatbot/astart
  4. Current: copy-only from project root
- 

### Oracle handling evolution

1. Only configured prerequisites when Oracle was already present
  2. Avoided auto-install in later version
  3. Current aggressive approach: auto-install via RPM/alien + configure system
-

## Stability fixes

- Removed duplicated blocks that caused `unexpected end of file`.
- 

# 9. Caveats & Recommendations

## Caveats

- Oracle XE via RPM/alien may break on unsupported distros.
- Node installation only works on apt systems.
- Script assumes correct project structure.
- Launcher requires a maintained `astart` file.

## Recommendations

- Add optional flags (e.g., `--with-oracle`).
  - Prefer installing launcher to `/usr/local/bin`.
  - Add non-interactive mode.
  - Pin Node and Python versions.
  - Add checksum validation for Oracle RPM.
-