**Installer**

**for**

**Ignition-based Applications**

**Document Version:**

**Dec 23, 2015**

Table of Contents

Table of Contents

1. Change History 2

2. Introduction 3

3. Use Cases 4

3.1 Fresh Installation 4

3.2 Product Update 4

3.3 Multi-project Update 4

3.4 User-created Install Package 4

4. Typical Workflow 5

4.1 Master Builder 5

4.2 Builder for a Specific Project 5

4.3 Install-project Creation 5

4.4 Installation 5

5. Install Panels 7

5.1 Overview 7

# Change History

| Version | Date | Author | Changes |
| --- | --- | --- | --- |
| 1.0 | Dec 23, 2015 | C. Coughlin (ILS) | Initial concept, use cases |

# Introduction

An Ignition-based product typically requires more than just a project file for complete installation. The product may require ancillary components that may involve a variety of installation techniques. These may be confusing to the end user. Absense of these components may result in an incomplete or inconsistent installation.

Additionally, updates of previously installed versions tend to be manual operations that are tedious and error-prone.

This document describes an ILS-Automation product that handles application releases in the familiar paradigm of an installer. Using the ILS Application Installer, the end user is presented with a wizard-style sequence of screens that handle installation of the various components that make up an application. These components may be, among other things: full or partial projects, UDT definitions, icons, internal and/or external python packages, SQL update scripts, and Java-modules.

The release bundle or installer for a particular delivery is delivered in a single file, an Ignition module file. Embedded within the module is an Ignition project that is available only when the install module is in the Gateway. The project contains the end-user wizard-style interface.

# Use Cases

The following major use cases that have driven design of this product.

## Fresh Installation

Description:

In a fresh installation, the entire product is installed. No attention is given to existing instances. There may be options that cover feature-selection or custom configurations.

Target products:

Batch Expert +

## Product Update

Description:

In a fresh installation, the entire product is installed. No attention is given to existing instances. There may be configuration options

Target products:

EM Chemicals Applications, Batch Expert +

## Multi-project Update

Description:

In a fresh installation, the entire product is installed. No attention is given to existing instances. There may be configuration options

Target products:

AED-RTA.

## User-created Install Package

Description:

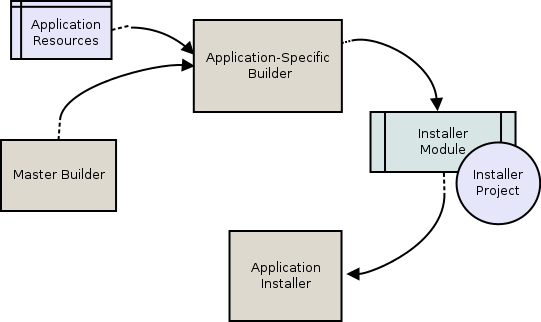
In a fresh installation, the entire product is installed. No attention is given to existing instances. There may be configuration options

Target products:

AED-RTA.

# Typical Workflow

This section describes the general order of operations involved with preparation and use of an installation bundle.



Workflow

## Master Builder

The master builder is a project supplied by ILS-Automation. It contains construction templates for each type of resource handled by the installer. In addition it supplies the general-purpose scripts needed to create the final installer project.

The master builder project code is accompanied by an "Application Installer" Ignition module. This module is necessary for both installer-construction and end-user phases.

## Application Resources

The application resources are the components to be installed. They exist in the file system of the development machine. Potential resources are:

* icons (.png files in a jar)
* UDT definitions (.xml export from Ignition)
* modules (signed .modl file)
* partial projects (Vision resources to be merged)

In addition to the above, a "starter jar" file is required. It is supplied by ILS-Automation along with the master builder. It is a primordial version of the Ignition module that is ultimately constructed.

## Application-Specific Builder

The engineer or developer preparing an installer for a specific product and version copies the master builder project and configures an install sequence.

The install builder is designed to produce an installer that follows the very common paradigm of a series of overlaid panels, each guiding an installation steps In the builder, the panels refer to file-system locations of the various resources, but do not load them until the ultimate insteller-project is created.

The amount of work required to create a project for an update is likely to be minimal if starting from the previous builder project. The panels are likely to remain unchanged, and, if the files paths point to locations in a build system (i.e. source control repository), file path configurations are likely to be unchanged as well. Any upgrades in the master builder may simply be merged.

## Installer Module/Project

Once the product builder project is complete and configured (including a loonk to the starter jar), the engineer/developer selects a location in the file system and launches the build. On completion of the build, an Ignition module will have been created. It will contain, embedded, all the resources that comprise the product. Most importantly it will contain the install-wizard for use by the end user.

The install wizard is an Ignition project that appears when the Gateway is re-started with the Application Installer module in place. The wizard project disappears when the Application Installer module is removed.

## Application Installer

The end-user loads the release-project into the target Gateway, starts the Designer and follows the installer instructions. The ILS “Application Installation” module must have been previously installed. On completion of the install, the user’s system is configured with a new project containing the subject product and version. Any existing user-data in the previous project is preserved. The newly-created-project is placed in a disabled state to prevent it from interfering with the existing system.

# Install Panels

Each panel described below is a template in the Master Builder project. Each panel handles a specific type of application resource.

## Overview

When complete, the panels are overlain to form the familiar wizard layout familiar to many users. A sample is shown below:

Wizard