Compass Agile Enterprise

Upgrading Compass Agile Enterprise

<div align="right">**Chapter 1**</div>

Background and Overview

Compass Agile Enterprise is a patterns-based Rapid Web Development and Enterprise Resource Planning (ERP) Framework.

Because it is intended to be configured and extended by developers and customers, care must be taken during the customization process as well as during upgrades.

This document contains summary comments on the considerations for creating extensions, focusing on the impact to the upgrade process. It is not intended to be an exhaustive reference on CompassAE-based development.

CompassAE is based on the Ruby on Rails (RoR) framework, and as such most guidelines and best practices for upgrading RoR applications can be applied to upgrading CompassAE.

Upgrade Process At-a-Glance

The steps below outline the basic process for upgrading CompassAE at a summary level.

1. Upgrade Preparation and Environment Updates
    a. Check gem versions and dependencies
    b. Install or update any new required gems, including native dependencies
2. Create Back-ups
    a. Back up Compass Database (including custom data) and static configuration files
    b. Back up SOLR indexes if required
    c. Back up CompassAE codebase
    d. Back up static content files
3. Prep customizations for CompassAE upgrade scripts
    a. Run any custom scripts required by user extensions
4. Upgrade CompassAE
    a. Run any pre-upgrade rake tasks provided

        b. Update CompassAE code

        c. Run upgrade schema migrations

        d. Run upgrade data migrations

        e. Run any post-upgrade rake tasks provided

5. Upgrade custom add-ons

        a. Run any custom scripts to upgrade data/code extensions

Upgrade Steps in Detail

Chapter 2

Step 1: Upgrade Preparation and Environment Updates

      b. Check gem versions and dependencies
      c. Install or update any new required gems, including native dependencies

Step 2: Create Backups

2a. CompassAE Database Backups

CompassAE relational databases should be backed up using the RDBMS tools available for the relevant DBMS engine. CompassAE does not add any requirements to common database backup procedures.

2b. SOLR Index Backups

If possible, it is safest to regerate SOLR indexes after the CompassAE and custom data and code are updated. This is because changes to the database can have an impact on SOLR indexes. TrueNorth Solutions will include release notes identifying any such impacts with the product upgrade. If a CompassAE product upgrade does impact index data, TrueNorth will include an update script to be executed in Step 3e (run Post-update rake tasks).

It is the customer's responsibility to account for any impacts of custom code and data.

2c. Back up CompassAE Codebase

Generally, the running CompassAE codebase should have been archived following the release of the current running version of the software. However if there is any question, the CompassAE code should be updated here.

CompassAE code is distributed as source and can be backed up with any archive utility, such as TAR or zip.

2d. Back up Static Content Files

During operation, CompassAE synchronizes relational data, indexes and file system (static) data. In a default install, everything but the relational data is stored in operating system files in the application directory tree, so archiving the application directory tree will cover 2a-2c. However it is common and even advisable for production environments to locate index and content-type files in special locations in the file system. In this case, it is common practice for system administrators to create an archival script based on the file layout and run this script at this step in the process.

Another common practice is to use Capistrano or similar utility, which uses symlinks to point from the code to the "durable" assets which must survive an upgrade. In this case, the archive step can be added to the Capistrano task to protect against the possibility that the upgraded code has any unintended impact on the durable data/assets.

Step 3: Prep customizations for CompassAE upgrade scripts

3a. Run any custom scripts required by user extensions

This step is a placeholder for any implementation-dependent steps that must be completed prior to upgrading CompassAE itself. Examples of activities that might be performed during this step include:

- Custom extraction of relational data, perhaps data that will be transformed prior to being reloaded
- Custom archival of any non-relational data including SOLR indexes. For example if there are large SOLR indexes that need to be transformed in some custom way offline, they can be processed here.

Step 4: Upgrade CompassAE

This is the actual CompassAE upgrade that is distributed by TrueNorth Solutions, Inc.

Capistrano Users

If you are using Capistrano, use the supplied Capistrano task to perform all the steps below.

Non-Capistrano Users

4a. Run any pre-upgrade rake tasks provided

Pre-upgrade rake tasks are batch scripts that make any necessary preparations for the schema,

data and code to change. Examples of this kind of task would be staging data for a schema change, or creating a work area for transformations that will occur during the upgrade.

```
rake compass:upgrade:prepare
```

4b. Update CompassAE code

Update the code itself

```
rake compass:update_version [INSTALL_PACKAGE_LOCATION='path/to/
install']
```

If the package file location is omitted, CompassAE will search for the install package in the directory from which the command is executed.

Warning	This step does not itself back up the previous version of the code. That is done by convention for Capistrano-managed installs, but for manually-managed installs, must be accomplished in step 2c.

4c. Run upgrade schema migrations

These are standard RoR schema migrations and as such are incremental in nature. The command follows:

```
rake db:migrate
```

Information	In order to maintain a manageable number of migrations, TrueNorth collects and merges migrations after data and functions are considered stable for a period of time.

4d. Run upgrade data migrations

Data migrations function similarly to schema migrations, except that their intent is to provide an incremental way to manage data unique to a release. The command syntax is:

```
rake db:migrate_data
```

4e. Run any post-upgrade rake tasks provided

This task will include data reloading, directory cleanup and other upgrade completion tasks:

```
rake compass:upgrade:cleanup
```

Step 5: Upgrade custom add-ons

5a. Run any custom scripts to upgrade data/code extensions

These scripts can now assume that the CompassAE schema, code and data are in sync and up-to-date, and can proceed with any custom code extension updates or data management tasks.