

# Getting Started

## Software Versions

This version of Habitat uses the following versions of this software:

Software	Version
Sitecore	9.1.0
Solr	7.2.1
Sitecore Installation Framework	2.0.0

## Locations and settings

This project assumes the following settings:

Setting	Value	Change in <sup>1</sup> <i>see below</i>
Source location	C:\projects\Habitat\	Habitat.Dev.config
Website location	C:\inetpub\wwwroot\habitat.dev.local\	gulp-config.js, settings.ps1, xconnect-XP0.json, sitecore-XP0.json
Website URL	<a href="https://habitat.dev.local/">https://habitat.dev.local/</a>	publishsettings.targets, settings.ps1, Habitat.Website.config
SQL Server	.	settings.ps1
SQL Server Admin	sa	settings.ps1
SQL Server Password	12345	settings.ps1
SOLR Location	c:\solr\	settings.ps1
SOLR URL	<a href="https://localhost:8983/solr">https://localhost:8983/solr</a> ( <i>Note https</i> )	settings.ps1
SOLR Windows Service Name	Solr	settings.ps1

<sup>1</sup> Files referred are:

- .\src\Project\Habitat\code\App\_Config\Environment\Project\Habitat.Dev.config

- `.\src\Project\Habitat\code\App_Config\Include\Project\Habitat.Website.config`
- `.\gulp-config.js`
- `.\publishsettings.targets`
- `.\settings.ps1`
- `.\build\assets\sitecore-XP0.json`
- `.\build\assets\xconnect-XP0.json`

## Prerequisites

---

**Important!: Check the prerequisites before starting the installation.**

- **Do check** the prerequisites of Sitecore Experience Platform in the release notes available on [dev.sitecore.net](https://dev.sitecore.net)
- **Do check** the [Resources](#) page for the tools needed
- **Always** run your Visual Studio or PowerShell Command Line with elevated privileges *or As Administrator*

The Sitecore install script will check some prerequisites, including running the `SIFPrerequisites.json` configuration.

### Solr

The installation requires the Apache Solr search engine. Solr must be running as a Windows Service. This can be accomplished through running [NSSM](#). Furthermore, Sitecore is secure by default and therefore Solr must be running as https. If your Solr environment is not currently running with HTTPS, you can create an SSL certificate for Solr by following these steps (make sure your Solr settings are correctly configured in `settings.ps1`):

1. Open an elevated PowerShell command line.
2. Run `.\build\GenerateSolrCertificate.ps1` to generate the certificate file in the correct location.
3. Add the following lines to the `bin\solr.in.cmd` file in Solr:
  - `set SOLR_SSL_KEY_STORE=etc/solr-ssl.keystore.jks`
  - `set SOLR_SSL_KEY_STORE_PASSWORD=secret`
  - `set SOLR_SSL_TRUST_STORE=etc/solr-ssl.keystore.jks`
  - `set SOLR_SSL_TRUST_STORE_PASSWORD=secret`

## Installation

---

### 1. Installing Sitecore

1. **Clone** the [Habitat repository](#) to your local file system.
2. Download the correct version of Sitecore from [dev.sitecore.net](https://dev.sitecore.net) and place it in the `.\build\assets` folder.
  - Habitat will install by default on an *Sitecore XP Single*, i.e. a standalone version of Sitecore CMS including xConnect.
  - The currently supported version is defined in the `.\settings.ps1` file

- The installation requires the following files:
  - Sitecore package: `.\build\assets\Sitecore X.X.X rev. XXXXXX (OnPrem)_single.scwdp.zip`
  - Sitecore configuration: `.\build\assets\sitecore-XP0.json`
  - Sitecore SOLR configuration: `.\build\assets\sitecore-solr.json`
  - xConnect package: `.\build\assets\Sitecore X.X.X rev. XXXXXX (OnPrem)_xp0xconnect.scwdp.zip`
  - xConnect configuration: `.\build\assets\xconnect-XP0.json`
  - xConnect SOLR configuration: `.\build\assets\xconnect-solr.json`
  - Identity Server package: `.\build\assets\Sitecore.IdentityServer X.X.X rev. XXXXXX (OnPrem)_identityserver.scwdp.zip`
  - Identity Server configuration: `.\build\assets\IdentityServer.json`
  - Certificate configuration: `.\build\assets\createcert.json`
  - Prerequisites configuration: `.\build\assets\Prerequisites.json`
  - Single Developer configuration: `.\build\assets\XP0-SingleDeveloper.json`
  - Sitecore license: `.\build\assets\license.xml`
- 3. Are you using system settings other than the defaults specified at the top of this page?
  - If yes, you need to update the files accordingly.
  - **Include or omit trailing slashes as per the default setting in each file!**
- 4. Open an elevated privileges PowerShell command prompt (started with **Run as administrator**)
- 5. Run `.\install-xp0.ps1`
  - On subsequent runs of the install, you may use the `-SkipPrerequisites` flag to skip prerequisite checks.

## 2. Build and Deploy Habitat

1. Restore Node.js modules
  - Open an elevated privileges command prompt (started with **Run as administrator**)
  - Run `npm install` in the root of repository.
2. If gulp has not been installed globally, you can do so by running `npm install -g gulp`
3. Build and publish the solution using either:
  - Open an command prompt with elevated privileges and run `gulp` in the root of repository.
  - Use Visual Studio:
    - Open **Visual Studio 2017** in administrator mode by right-clicking on its icon and selecting **Run as administrator**.
    - Open the Habitat solution.
    - Open the **Visual Studio 2017** Task Runner Explorer pane (**View | Other Windows | Task Runner Explorer**).
    - Switch to "Solution 'Habitat'"
    - Run the "default" task

## 3. Rebuild search indexes

1. **Rebuild** the `sitecore_master_index` and `sitecore_web_index` to deploy the updated schema to Solr and

index the deployed content.

- This is required for Habitat search-based features to work properly.

## Additional Information

---

### Gulp

The project is configured to run Gulp through the command line or using the Task Runner Explorer pane in Visual Studio.

In the initial installation running the **default** task will execute all the configuration and building tasks for the solution. If for some reason setup fails, it is possible to run the install tasks one by one:

- **Publish-All-Projects** builds and publishes all the Visual Studio projects to the Sitecore website in the right order
- **Apply-Xml-Transform** makes the needed changes to the web.config and other existing configs in the Sitecore website
- **Sync-Unicorn** runs a complete synchronization of Unicorn for all projects in the right order

### Helper tasks

- The **Auto-Publish-...** tasks help by automatically publishing files when they are changed.
  - The **Auto-Publish-Css** automatically publishes .css files when changed (Configure Sass compilation in Visual Studio)
  - The **Auto-Publish-Assemblies** task publishes assemblies as they are built using the standard Visual Studio build process
  - The **Auto-Publish-Views** task publishes .cshtml files when they are changed.
- The **Publish-...** tasks helps you by manually publishing different types of files or project types to your website.

### SMTP Settings

Habitat project uses the default Sitecore helpers to send emails. For this to work, you need to set the SMTP settings in Sitecore.config.

**NOTE:** If you are planning to use secure connections with your SMTP server you need to add following section to your web.config.

```
<system.net>
  <mailSettings>
    <smtp deliveryMethod="Network">
      <network enableSsl="true" />
    </smtp>
  </mailSettings>
</system.net>
```

### Facebook Login

Habitat demonstrates utilizing the Federated Authentication features of Sitecore to allow login via Facebook account. To enable this functionality:

1. [Register a Facebook App](#) using a Facebook Developer Account.
2. Add the domain under which you are running Habitat to the *App Domains* setting of your Facebook App. By default this is `habitat.dev.local`.
3. Use the *App ID* and *App Secret* from your app settings to populate the `Sitecore.Feature.Accounts.Facebook.AppId` and `Sitecore.Feature.Accounts.Facebook.AppSecret` settings, by default found in `Feature\Accounts\code\App_Config\Include\Feature\Feature.Accounts.config`.