# Sugar Developer Quick Start

version 1

**Trademarks**
Atonce, Neova, NeovaCoreEngine, Sugar, Sugar System, Jam are either registered trademarks or trademarks of Atonce Technologies and may be registered in the United States or in other jurisdictions including internationally. Other product names, logos, designs, titles, words, or phrases mentioned within this publication may be trademarks, service marks, or trade names of Atonce Technologies or other entities and may be registered in certain jurisdictions including internationally. No right or license is granted to any Atonce trademark.

**Third-Party Information**
This guide contains links to third-party websites that are not under the control of Atonce Technologies, and Atonce Technologies is not responsible for the content on any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. Atonce Technologies provides these links only as a convenience, and the inclusion of the link does not imply that Atonce Technologies endorses or accepts any responsibility for the content on those third-party sites. No right, license or interest is granted in any third party technology referenced in this guide.

**NOTICE: THIS PUBLICATION AND THE INFORMATION HEREIN IS FURNISHED “AS IS”, IS SUBJECT TO CHANGE WITHOUT NOTICE, AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY ATONCE TECHNOLOGIES. ATONCE TECHNOLOGIES ASSUMES NO RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR INACCURACIES, MAKES NO WARRANTY OF ANY KIND (EXPRESS, IMPLIED, OR STATUTORY) WITH RESPECT TO THIS PUBLICATION, AND EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSES, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.**

Atonce Technologies

Published November 2009

## Content

section 1

# Introduction

1

## The concepts behind

This document aims to quickly introduce developers to the SUGAR framework, concepts, tools and services. Sugar is a generic framework, intended to be used as a software base for a wide range of applications, from embedded video-game runtimes to serious real time authoring tools.

To achieve this goal while remaining scalable, the Sugar framework is conceptually based upon the SOP paradigm (Service-Oriented Programming). There are some significant benefits to this approach :

- Capabilities of the system to significantly reduce the complexity of integration and customization ;
- Ability of programmers to rewrite all or parts of any component, including sensitive ones ;
- Provide an efficient way to modularize collaborative developments shared by people working remotely ;
- Enhance the simpleness of a horizontal semantic schema rather than an inheritance tree (mixing up polymorphism and re-usability): finally, it remains a well defined set of user-services ;
- Ensure the embedding of any technical constraints, like a platform specific environment, a specific native programming language or some implementation tips.

## Overview

The Sugar project is composed of several sub-projects:

- The framework's foundation unit ;
- The framework's optional components, gathering semantically several services ;
- Some code-generation and software-construction tools, aiming to ease the project expansion, to reduce the maintenance work, ensure the global quality ;
- Some proprietary authoring editors and others asset management tools.

## Requirements

At the moment, the Sugar framework is available for the following configurations:

| Targeted platform | Hosting platform | Hardware | Software |
|---|---|---|---|
| Windows XP/ Vista/ 7 | Windows XP/ Vista/ 7 | NVidia/ ATI/ GC | MS Visual Studio<br>Subversion<br>Acrobat Reader<br>DirectX 9<br>OpenGL 2.2 |
| Mac OS X | Mac OS X | any | Xcode<br>Subversion<br>Acrobat Reader<br>OpenGL 2.2 |
| iPhone 3G | Mac OS X | any | Xcode<br>iPhone SDK 3.1<br>Subversion<br>Acrobat Reader<br>OpenGL 2.2 |

This list will change frequently. Please visit the official sugar website at http://www.atonce-technologies.com/sugar to obtain an up-to-date version of this document.

## Installation

Firstly, depending on your targeted platform, all the third-party tools need to be installed in your computer and you need to be familiar with them. For a complete list of these requirements, please read the previous section.

The whole Sugar project (and sub-projects) is hosted on a web-based Subversion server. In order to checkout the Sugar's repositories to your computer filesystem, a valid Sugar registration is required. Otherwise, contact the support team at `support@atonce-technologies.com`.

If you are a valid registered user, you should have your personal account ID (username and password) mailed back by the support team at the time of the registration. Now proceed with the following steps:

- Create a new directory, anywhere, named for instance `atoncetech` ;
- Inside this root directory, create two new folders, named `devcenter` and `sugar` ;
- Using you favorite svn client, *checkout* theses two working directories, with the web url `www.atoncetech.com/svn/devcenter/trunk` and `www.atoncetech.com/svn/sugar/trunk` respectively.
- Depending on the OS :
  - On Windows: Double-click on the script file `install.wsf`, located in the `devcenter` directory.
  - On Mac: From a bash prompt, run the script file `install.sh`, located in the `devcenter` directory.
- Check the installation by typing the `atmake` command from a shell console. If successful, the atmake tool should display a message listing all the available modules.

## Files organization

Once the installation process is done, the project's files are organized following the map below :

- atoncetech
  Root folder of Atonce's projects (This name can be arbitrary changed).
  - devcenter
    Folder gathering several generic tools, implemented by the Sugar system
    - atmake
      Source code folder of the atmake programming tool
    - deprecated
      Folder containing old stuffs
    - python_*
      Python customized distribution, one for each supported OS
    - atmake.*
      Startup script files of the atmake programming tool
    - install.*
      Script files to install the Sugar system
    - uninstall.*
      Script files to uninstall the Sugar system
  - sugar
    Source code files of the Sugar project
    - misc
      Deprecated or experimental files
    - doc
      Technical reports, user manual and others documents about tools and APIs
    - proj
      IDE project files for MSVisualStudio, Xcode, Eclipse and makefiles
    - sidl
      Conventionally, this folder holds all the SIDL files for the whole sugar projects
    - suapp
      Source-code files of the Sugar-based applications, including unit tests.
    - subin
      Binaries of the Sugar project, including libraries, add-ins or executables
    - subuild
      Third party packages required to successfully build all parts of the Sugar system
    - sufw
      C++ Source code of the Sugar framework, foundations and components
      - core
        Foundations and some system services
      - directdriver
        Graphics, sound, inputs and feedbacks HAL
      - flash
        Shockwave Flash support

## Generating

Parts of the Sugar framework are program-generated from a collection of SIDL files, conventionally located in the `atoncetech/sugar/sidl`. This is especially the case for some C++ source-code files and projects. Firstly, to initiate the generation stage, enter the atmake command:

```
$ atmake sugar gencode
```

at the terminal prompt. As a result, SIDL interfaces, skeletons and unit tests have been written back. Secondly, enter the next command:

```
$ atmake sugar genproj
```

to generate the various project's files for the native IDE of the current hosting OS. In the moment, Microsoft Visual Studio and Xcode IDEs are supported. For your information, the default SIDL entry point is the file `atoncetech/sugar/sidl/fw.sidl`, considering itself several others SIDL files.

## Compiling

Once the project files have been generated, targeted programs and libraries can be constructed following the common usage of your favorite programing IDE. Project file are located in the folder `atoncetech/sugar/proj`.

## Executing

Resulting binaries will be produced in the folder `atoncetech/sugar/subin`. Among theses files we find especially:

- `np_suapp.xpi` plugin for Mozilla Firefox 3.5.4 or upper. This plugin allows to display any .suapp package (Sugar downloadable content) through the Mozilla web browser ;
- `su_fw[_DBG].[lib|ar]` static archive of code, corresponding to the whole Sugar framework ;
- `su_fw_test[_DBG][.exe]` executable, representing a compilation of unit tests.

## Cleaning

To clean you file-system of all theses generated files, the following command shall be used:

```
$ atmake sugar clean
```

Several others commands and options are available. To have a self-documented overview of the functionalities of the sugar module for atmake, the command `atmake sugar` could be used alone to display a reminder as a help page.

## On the SIDL side

(todo ...)

## On the Stub side

(todo ...)

## SIDL side

(todo ...)

## Stub side

(todo ...)

appendix 1

# Sugar's API quick tour

A1

## The core foundations

(todo ...)

## Some C++ namespaces

(todo ...)

appendix 2

# SIDL grammar through examples

A2

## A basic SIDL file

(todo ...)

## Some interfaces

(todo …)

## Some implementations

(todo …)

## Collect project's source-code

(todo …)

## Ensure project's unit tests

(todo …)