

# SaaS CUSTOMIZATION AND PERSONALIZATION





## TABLE OF CONTENTS

> 1.0 Introduction	1
> 2.0 Look and Feel	2
2.1 Accessibility	2
2.2 Corporate Standards	2
2.3 Personal Preferences	2
2.4 White Label	3
> 3.0 Internationalization (I18N) and Localization (L10N)	3
3.1 Architecture Implications	4
> 4.0 Display Options	4
> 5.0 Print Options	5
> 6.0 Security Policies	5
> 7.0 Workflow and Approval	5
> 8.0 Customer Defaults	6
> 9.0 Summary	6
> 10.0 References	7
> 11.0 Glossary of Terms	8



## 1.0 INTRODUCTION


This paper is one of a series of papers that explore and discuss the technical architectural components of the Software-as-a-Service (SaaS) model.

SaaS shares the distinction of being both a business model and an application delivery model. SaaS enables customers to utilize an application on a pay-as-you-go basis and eliminates the need to install and run the application on the customer's own hardware. Customers generally access the application via a Web browser or thin client over the Internet. SaaS is most often subscription based and all ongoing support, maintenance, and upgrades are provided by the software vendor as part of the service. Application customization capabilities, if available at all, are generally provided to all customers in a consistent manner. From the perspective of the software vendor, the SaaS model provides stronger protection of its intellectual property, operational control of the environment running the software, and generally a repeatable revenue stream from the service subscription fees. Software vendors have varying capabilities and applications can come in varying flavors but SaaS applications most typically support many unique customers using a single instance of that application — also known as multi-tenancy.

This paper covers the topic of customization and personalization as it applies to SaaS architecture. Customers tend to have different perspectives and preferences on how they would like things to work, and even different users within a given customer may differ in what they want. In the world of SaaS, the ideal is to deliver, manage and maintain a single application for all customers to use rather than customized instances for each customer. So the goal is to provide as much flexibility as possible to customers and their users to give them a sense of control over the application. This is accomplished by providing as much customization and personalization functionality as possible.

The terms customization and personalization are sometimes used distinctively and sometimes interchangeably. Sometimes personalization is used to describe presenting content to individual users based on knowledge of who they are. For instance, if a corporate user is a member of a particular department, a SaaS application may have logic to present this user with information that is relevant to that department. Another example might be to present a logged in user with their name at the top of each screen as a link to their individual employee record, thus making it easy to access their specific data. Other times personalization refers to giving a user the ability to define what parts of the application they prefer to see (i.e., My Yahoo or iGoogle). Customization is sometimes thought of as defining certain preferences that affect how an application behaves (i.e., the various Options settings in the Microsoft Office products).

For the purposes of this paper we will consider the terms to be interchangeable and in order to avoid confusion we will only refer to this as customization. Rather than worry about



which is which, we will instead present a number of different aspects of an application where users can be provided functionality to customize. This list will inherently be limited as there are virtually unlimited possibilities in this area, but the ideas presented here can be applied to many different types of applications.

## 2.0 LOOK AND FEEL

The term “look and feel” generally refers to the visual aspects of a user interface such as the font type and size, colors, backgrounds, etc. With the emergence of cascading style sheets (CSS) it has become a lot easier to control the look and feel of a Web application and even to give control of many attributes to individual users. There are many reasons why users might want to do this besides individual tastes or why a SaaS vendor may want to provide look and feel customization.

### 2.1 Accessibility


One reason for wanting to control the look and feel of a SaaS application has to do with physical limitations of individuals. People with weak eyesight might prefer looking at larger font sizes and people who are color blind maybe able to distinguish certain colors whereas others look the same. In fact, there is a U.S. government regulation referred to as Section 508<sup>1</sup> that requires U.S. government agencies to provide equal access to disabled employees and compliance maybe necessary for a SaaS vendor to become an approved government provider.

### 2.2 Corporate Standards

Customers may want to control the look and feel of the SaaS application in order to comply with approved corporate standards and so that it can “fit in” with other tools provided to employees. Some companies are very concerned with their brand and want their employees’ desktops to reflect that brand rather than those of a bunch of different SaaS vendors. Others at least would like to be able to insert their corporate logo for display on the application pages. So, allowing these customers flexibility in this area can ease acceptance of the outside SaaS vendor.

### 2.3 Personal Preferences

In addition to corporate level standards, individual users also like control over the look and feel. Aside from defining individual preferences for internationalization and localization (see section 3) such as language and time zone, users often like to control their own work environment within the application. This can include structural changes to the UI such as customizing dashboards to only show data relevant to the individual, and can even go as far as allowing users to alter font styles and colors to suit their



individual preferences. Addressing the needs of individual users in this way can also serve to ease acceptance of the SaaS solution.

## 2.4 White Label

A SaaS vendor may want to white label the service so others can use their application as part of a more comprehensive branded offering. A white label service would be one created by the SaaS vendor but then other companies re-brand to make it look like their own. An example might be a tool used by a consultant to deliver a broader branded service offering.

## 3.0 INTERNATIONALIZATION (I18N) AND LOCALIZATION (L10N)

Internationalization and localization are means of adapting an application to different languages and regional differences. Because these terms are long, they are often referred to by their respective acronyms i18n and L10n — these conventions use the first and last letters of the words and then the number of letters in between, with the capital L in localization helping to distinguish it from the i in i18n. There is a subtle but important distinction between the terms. According to Wikipedia:

*Internationalization is the adaptation of products for potential use virtually everywhere, while localization is the addition of special features for use in a specific locale. Internationalization is done once per product, while localization is done once for each combination of product and locale. The processes are complementary, and must be combined to lead to the objective of a system that works globally.*

Aspects commonly covered by i18n and L10n include language of screen prompts, time zone, date/time format, telephone numbers, addresses, currencies and number formatting. Other more subtle aspects might include cultural differences regarding symbols and colors. A color that might be deemed fashionable in one country might be considered offensive in another.

There are many other important business aspects of i18n and L10n including data privacy laws, import/export laws, regulatory compliance (i.e., Basel Accords — Europe, Sarbanes-Oxley — U.S., etc.), accounting standards (i.e., taxes, accruals, financial reporting, etc.), rounding and measurement units. Covering the details of all of these is beyond the scope of this document, but they are listed to emphasize the depth of this topic and provide the SaaS vendor with a starting point for consideration.



### 3.1 Architecture Implications

From an architectural standpoint, many modern programming languages are designed to easily handle this type of customization if the built-in functionality is leveraged. If done properly, a simple user-defined setting can inherently change date/time handling, text processing and character encoding. As a general rule, it is also good idea to store screen prompts separately in what are called resource files or bundles rather than hard coding them into pages. This way they can easily be provided in different languages and with the user specifying preference, the application can display the appropriate set.


Another consideration regarding i18n is providing for translations of user-provided content in different languages. In a large global enterprise it is common for a user in one country to create content in their native language and then have other users review or otherwise utilize that content in another country.

This brings up the issue of translation. The technologies available today to automatically translate content to different languages are very limited in their capabilities. Especially if content is laced with technical terminology or jargon specific to an individual organization or industry, it is virtually impossible to get accurate translations in an automated fashion. There are a couple of alternatives that can be provided involving workflows (the topic of workflows will be covered later in this paper). One approach involves routing content off to a third party translator that then returns the content in the language requested for a fee. Aside from the inherent cost of this approach, the results can be inaccurate due to the terminology issue referred to previously. A better approach might be to route the content internally within the organization to a person fluent in both source and destination languages, as well as any technical or corporate jargon that it may contain. This approach generally would provide the most accurate results.

Just as there are many other business aspects of i18n and L10n, some of which were listed in the previous section, there are corresponding implications to be considered from a SaaS architecture standpoint. As the SaaS vendor decides what aspects of i18n and L10n will be supported by their solution, it is important to consider how the implementation will impact the overall architecture.

## 4.0 DISPLAY OPTIONS

Certain display options may be useful for some clients but not work at all for others. A simple example might be a nice hierarchical representation of a departmental structure that would display well for a smaller customer. A larger enterprise with hundreds of departments might find the same display of data virtually unreadable. Allowing customers to customize the SaaS application to utilize or avoid features such as this can provide more universal usefulness.



Another example might include allowing users to define what components of an application they see when they login. With a full featured application, it is not unreasonable to expect certain users to only utilize a specific subset of this functionality on a regular basis. In order to make their user experience more productive it may be helpful to allow them to hide or display certain components from view as appropriate.

## 5.0 PRINT OPTIONS

As much as we continue to strive toward a paperless workplace, corporate users are still addicted to printed output! Providing customers with the ability to customize what their output will look like can be critical. This can include providing corporate logos, defining what page headers and footers will look like, and providing copyright, privacy and confidentiality notices.


Another useful customization option for printed output is to give users the ability to define styles for structured content that include options such as font style and size, color or options such as bold, italics or underline. This way they can easily change the look and feel of their output with a few simple clicks of the mouse.

## 6.0 SECURITY POLICIES

Security is a major concern for many corporate customers and allowing these customers to customize a SaaS application to comply with their individual security policies can be helpful. Some of the settings that might be appropriate may include password formats (i.e., minimum and maximum lengths, requirements to include capital letters or numeric and special characters, etc.), number of failed login attempts before locking out a user ID and password expirations. SaaS solutions may also want to provide the customer with the ability to define a customized login message with a corporate standard security warning. Another potential security setting might allow a customer to limit login access to the SaaS application from certain IP addresses or ranges (i.e., only allowing users to login from the corporate network). These are just some of the many security related settings that can differ from customer to customer and should be considered for inclusion in the customizable options provided by a SaaS application.

## 7.0 WORKFLOW AND APPROVAL

Workflow is a term used to describe defined steps in a process. Each step generally has some defined inputs, some type of processing or transformation that is then to be done on the input, and finally some defined outputs. A simple example might be defining the steps needed in an approval process. A worker edits some data and then needs to submit it for approval. The approval process may involve separate individuals such as a manager



and someone from the legal department. The process generally provides for some way of passing the work around, often via a simple email notification, and it has order with a clear beginning and end.

Providing this type of functionality in a SaaS application can be very useful to customers. It allows them to apply controls to their work and this is something auditors and regulators like to see. In fact, some entire SaaS applications can represent one step in a larger corporate workflow, accepting inputs from one or more systems, processing the data, and then passing it on to another system. An example of this type of workflow might be a CRM system where customers and prospects are managed through a sales process. Once the sale has been completed, the information is passed on to operations for fulfillment.


## 8.0 CUSTOMER DEFAULTS

Customers often want to configure an application so their users all experience some default behaviors. This configurability is generally limited to some set of administrative users that are authorized to set such defaults. There are many defaults that maybe set, some of which have already been covered. One that we have not yet discussed and is often overlooked is the definition of default permission settings. If a SaaS application provides granular control over access to different data, it can be very useful for the administrators of the application to define default permissions for these different types of data. For example, corporate policy may provide open access to data across departments. In this case administrators can set default permissions to allow view access to all users. Other companies may follow a more restrictive policy whereby only the people who are members of a given department are authorized to view data related to that department. Administrators can then set the defaults to reflect these restrictions.

Providing this level of configuration in a SaaS application provides the customer with the flexibility to apply their specific controls and policies so their users can uniformly follow them. If this functionality is not provided, users are forced to authorize access to data explicitly and this makes it more likely that policies are overlooked either willingly or inadvertently.

## 9.0 SUMMARY

Providing functionality in a SaaS application to allow customers and their users to customize the solution according to their specific needs is critical to achieving the goal of providing a single application shared by everyone. This requires careful thought, advance planning and design in the overall application.



In this paper, we have touched upon various topics among the vast list of possibilities that can be considered for customization. Where appropriate, issues for consideration were raised and examples of alternatives presented. The topics covered here can have general applicability in many different types of applications.

One thing to remember about customization is that virtually every aspect of an application can be a candidate for customization options. The difficulty is in deciding which customization functionality will provide the greatest number of users with the most benefit, thereby creating a user experience that is both productive and pleasant. There is clearly a trade-off involved here, because the more customization options provided the more unwieldy the customization process becomes. So, the trick is to find the balance between giving users what they want without overwhelming them with options.

## 10.0 REFERENCES

<sup>1</sup>Section 508 Amendment to the Rehabilitation Act of 1973, see <http://www.section508.gov/> as a starting point for information.

## 11.0 GLOSSARY OF TERMS

Term	Description
CRM	<b>Customer Relationship Management (CRM)</b> generally refers to an application that helps with the interaction with customers and prospects as they move through the sales process.
CSS	<b>Cascading Style Sheets (CSS)</b> is a style sheet language used to describe the presentation of a document written in a markup language.
Customer/Tenant	<b>Customer/Tenant</b> is a term in this document that refers to the company or business that subscribes to the SaaS offering.
i18n	<b>i18n</b> stands for <b>Internationalization</b> , the adaptation of products for <i>potential</i> use virtually everywhere, which is done once per product.
L10n	<b>L10n</b> stands for <b>Localization</b> , the addition of special features for use in a <i>specific</i> locale, which is done once for each combination of product and locale.
CRM	<b>Customer Relationship Management (CRM)</b> generally refers to an application that helps with the interaction with customers and prospects as they move through the sales process.
CSS	<b>Cascading Style Sheets (CSS)</b> is a style sheet language used to describe the presentation of a document written in a markup language.
Customer/Tenant	<b>Customer/Tenant</b> is a term in this document that refers to the company or business that subscribes to the SaaS offering.
i18n	<b>i18n</b> stands for <b>Internationalization</b> , the adaptation of products for <i>potential</i> use virtually everywhere, which is done once per product.
L10n	<b>L10n</b> stands for <b>Localization</b> , the addition of special features for use in a <i>specific</i> locale, which is done once for each combination of product and locale.
SaaS Offering	A <b>SaaS Offering</b> in this document is defined as a turnkey service offering, which includes the application license, maintenance, application support, subscription pricing, and hosting and associated delivery infrastructure.
SaaS-Enabled Application	A <b>SaaS-Enabled Application</b> is defined as an application that has been designed and built to be deployed and consumed in a service model, and incorporates many of the attributes previously defined above.
Users	<b>User</b> in this document is defined as an employee of the customer, as is the ultimate user of the service.



**Worldwide Headquarters**

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA  
Tel: +1 781 280-4000 Fax: +1 781 280-4095  
[www.progress.com](http://www.progress.com)

**For regional international office locations and contact information, please refer to [www.progress.com/worldwide](http://www.progress.com/worldwide)**

© Copyright 2008 Progress Software Corporation. All rights reserved. Progress is a registered trademark of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. or other countries. Any other trademarks contained herein are the property of their respective owners.