



CaptainCasa Enterprise Client

Release 2.5

Table of Contents

Overview.....	3
Some more Details.....	4
Java 1.6 Update 10 Adaptations.....	4
Component Library Extensions.....	4
Spinner.....	4
Animation of Image Captures.....	4
Client Side Value Synchronization.....	5
Sound.....	5
Shadowed Pane.....	5
Improved Background Painting.....	5
Scrollbar Coloring.....	5
Touch Pad Layout.....	5
Dynamic Integration of XML Layout.....	6
PDF Output / PDF Printing.....	6
Subsystem Integration.....	7
Embedded Usage Mode.....	7
Tool Improvements.....	8
Improved Preview Options.....	8
Code Generation Improvements.....	8
Direct WYSIWYG Positioning in PAINTAREA.....	8
Eclipse Plug-in.....	9
Tools run as stand alone Application.....	9
Layout Editor Extension Concept.....	10
Extension Concept.....	10
Extension Kits.....	10

Overview

A lot of improvements were added to the CaptainCasa Enterprise Client environment. This document only describes the “big points”.

- **Java 1.6 Update 10 Adaptations** - especially in the area of applets there is a major improvement due to 1.6 Update 10. Applets can now be started in isolated virtual machines with individual memory configuration.
- **Component Library Extensions** - a couple of new components were added and new attributes/ features were made available to existing ones.
- **PDF Output/Printing** - PDF can now directly be output on client side by embedding a corresponding component. The component is a 100% Java component and renders static PDF that represents Acrobat Reader 5.0 standards. The output includes client side printing.
- **Subsystem Integration** - a couple of client side components allow to talk to other client programs, either by http or by UDP protocol. As result the integration of client side subsystems (e.g. scanners, card readers, ...) can be done in a simple way.
- **Embedded Usage Mode** - the client and the server (if running in Tomcat) can be bundled to form a stand alone program. In this case Tomcat runs in so called embedded mode, the client directly talks to Tomcat - it's not going through the http stack.
- **Tool Improvements** - a couple of improvements: improved coloring, improvements in the WYSIWYG-functions, an Eclipse plug-in for synchronizing with Eclipse projects, new functions for generating the managed bean Java code.
- **Layout Editor Extension Concept** - the Layout Editor can now be extended to integrate other pages and functions.
- **Web Service Integration Kit** - This is an add-on kit providing the integration of Web Services using Axis2.
- **ABAP Integration Kit** - This is an add-on kit providing the direct integration of ABAP function modules to process screens. Dedicated usage scenario: touch screen interfaces that are implemented directly in front of ABAP logic.

Some more Details...

Java 1.6 Update 10 Adaptations

Java 1.6 Update 10 is a significant release for the client side processing of Java applications. The most important feature: applets now can be started within the same isolation level than web start applications.

This means: applets can be started so that each applet is opened in an isolated Java virtual machine. Before all applets were run in one virtual machine, competing for the right version and for memory. Now each virtual machine can be configured with its own memory profile - before it was restricted to 64 MBytes for all client side Java programs.

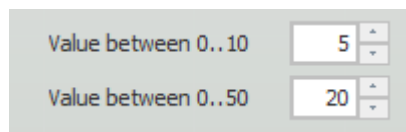
The CaptainCasa Enterprise Client generation of html-applet pages was updated so that by default each applet is started in an isolated virtual machine, receiving 256MBytes as maximum. Of course you may individually update the corresponding parameters.

Component Library Extensions

Some new components were added:

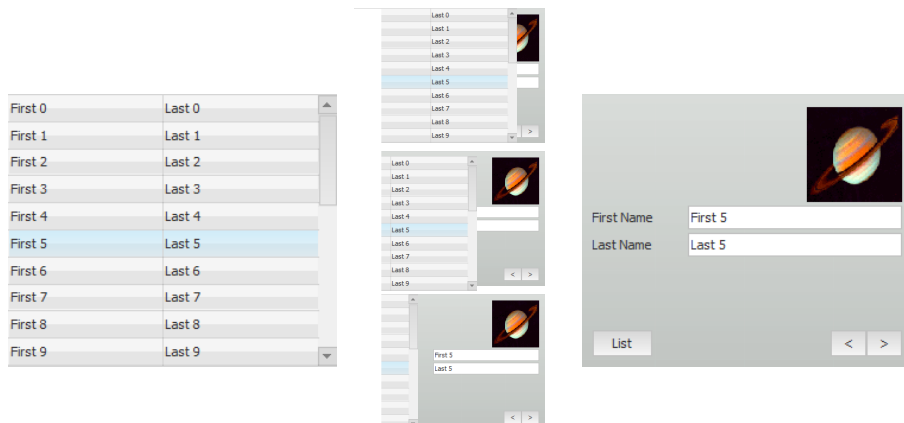
Spinner

The spinner allows to enter an integer value within a certain range either by keyboard input or by pressing the up/down buttons.



Animation of Image Captures

The image-capture component is able to capture a certain pane's content and to hide/move it in an animated way. It is typically used to clearly show the user that a certain area of the screen is updated.



In the example the area is first filled with a grid. On grid selection the grid moves to the

left and shows the new detail content.

Client Side Value Synchronization

In case where one and the same content is rendered twice within a page there is now the possibility to synchronize the corresponding components without roundtrip to the server. Components now can bind to certain value-reference. Components binding to the same value reference are automatically synchronized on client side.

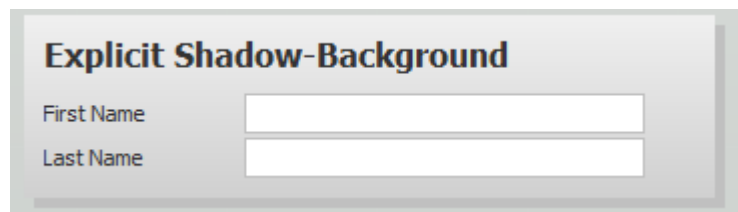
Typical usage scenario: data that is both entered within the row of a grid and within a detail area.

Sound

A sound component allows to output .wav-files on client side. The component is included within the status-bar so that error messages come with a certain sound.

Shadowed Pane

The shadowed pane is a just normal container with shadowed rendering. Colors and distances can be individually set.



Improved Background Painting

The background painting (“BGPAIN command”) was extended and now provides a set of new commands. These include a flexible text output - and more.

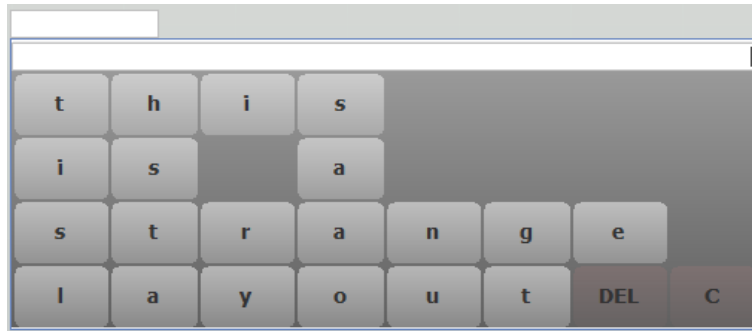


Scrollbar Coloring

The coloring of scroll bars always was an issue “under discussion”... Now the default coloring was updated so that they much better fit into non-gray scenarios. In addition scrollbars are available in a shaded-variant - in this case they do not provide colors on their owns, but just draw a dark or light shading.

Touch Pad Layout

Fields can be configured to open a touch pad. This touch pad now can be easily configured on server side to adapt to your individual layout.



Dynamic Integration of XML Layout

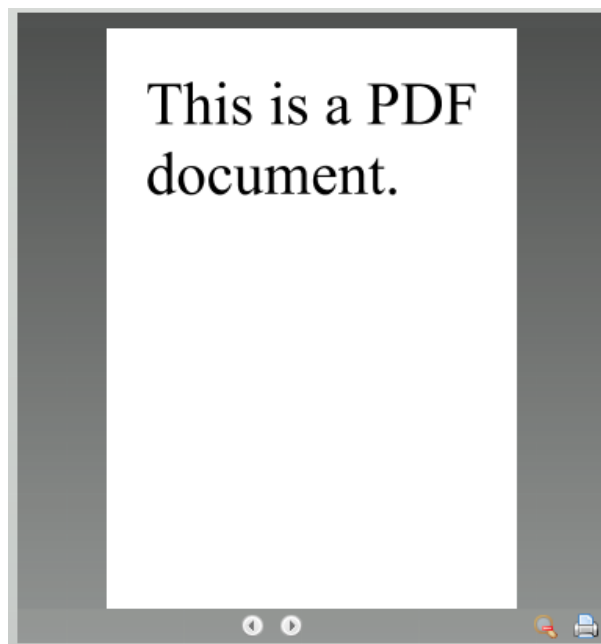
The creation of dynamic areas within pages is much simpler now: you do not have to work with the JSF component tree but can just embed XML dynamically. The XML is parsed at runtime and transferred into corresponding components. - Of course you can still directly manipulate the JSF component tree at run time as well.

This makes it easy as well, to embed XML layouts coming from different resources than the file system.

PDF Output / PDF Printing

Reporting output that is produced as PDF document can now be directly output on client side - without the need of having installed Adobe Acrobat Reader. The PDF rendering is done by a Java Swing component (pdf-rendered) - that is able to render PDF documents that can be rendered with Adobe Acrobat Reader 5.0. The component was originally developed by Sun/Javasoft as part of the Swing project and then was open sourced. The exact definition of capabilities is available at <https://pdf-renderer.dev.java.net/>.

From our usage experience it renders all of our PDF documents in a reliable and fast way - so we can recommend to use it.



The component also allows direct client-side printing. In case the server already knows

the name of the client-side printer, the typical print-dialog does not need to show up.

The direct rendering of PDF is a big improvement in the area of reporting data output. With release 2.1 we provided the direct output of Jasper-Reports documents on client side. PDF on the one hand is independent from the reporting tool's format, on the other side its format is very compact compared to the XML produced by Jasper-Reports.

Together with the client side component we added a concept to provide dynamic content on server side - so that it can be picked by a URL and so that it does not use the server side file system at all.

Subsystem Integration

Some new components enable the CaptainCasa client to talk to other software running on the client machine. This includes:

- Capability to receive http-get-messages on client side. The content of the message is sent to server side as result.
- Capability to receive UDP messages on client side. Again the content is sent to server once a message is received.
- Capability to send http-messages from the client side and to send back the response of this message directly to the server side. By this the client can call another program on client side, receive back a response (e.g. data queried from the other program) and direct load the data to the server side processing.

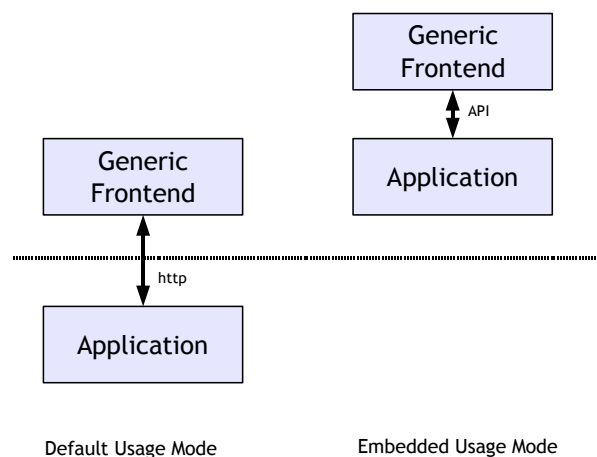
In combination with our touch screen components this subsystem integration completes the possibilities to use CaptainCasa Enterprise Client in operational environments, such as Manufacturing Execution.

Embedded Usage Mode

The typical usage mode is: a rich client is running in the frontend, the application is running in the backend - both being coupled by a http communication.

“Embedded Usage Mode” now means, that it is possible to directly couple the frontend-part and the backend-part so that they run in one virtual machine. In addition the communication between both is not done through a network protocol (http) anymore, but the client directly talks to the server by using a Java API.

Result: a fat client, stand alone usage concept for CaptainCasa based applications.



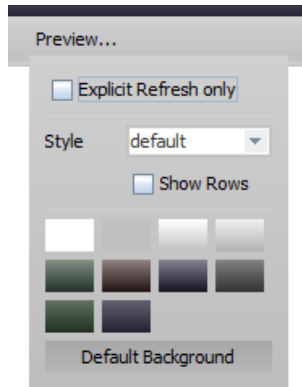
Usage scenarios: stand alone delivery of applications (“CD delivery”), demo applications

that need to run in any case - without personal firewalls blocking the communication.

Tool Improvements

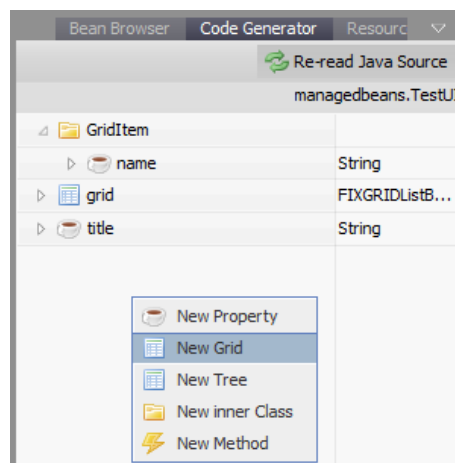
Improved Preview Options

It is now possible to explicitly set the style and background of the preview area. In addition it is possible to also highlight components which normally are only structure-components, without own rendering. Most prominent one: the ROW component.



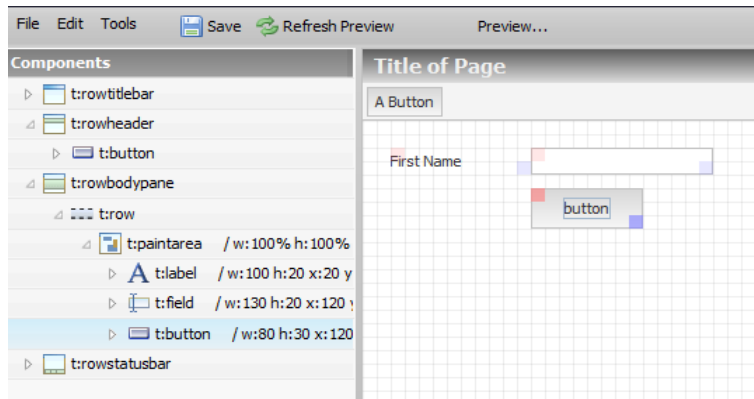
Code Generation Improvements

In the code generator you now can directly create the Java source that is required for managing grids and/or trees on server side.



Direct WYSIWYG Positioning in PAINTAREA

A specific container allows the direct arrangement of components by their x,y-position. Inside this container components can be arranged within the Layout Editor by directly positioning and sizing them via drag & drop.



x,y-positions are defined as pixel positions. Pixel positions and sizes in general are scaled at frontend side dependent from the client's settings.

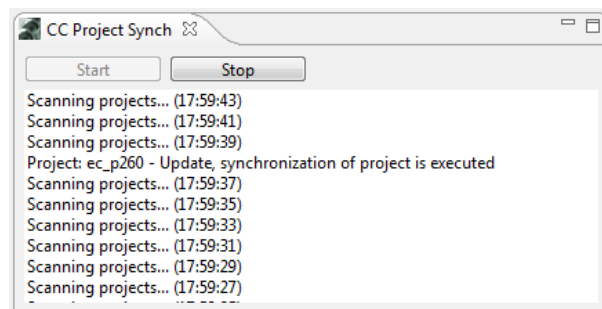
Eclipse Plug-in

The Layout Editor is a toolset that is used for:

- editing pages
- generating code
- managing resources

It contains these tools that are required for “layout work” - while all the tooling that is required for “coding work” is left to the development environment. The Layout Editor tools set and the development environment (e.g. Eclipse) share the same file directory in which sources are kept.

For a simplified usage together with Eclipse there is a plug-in available that automatically synchronizes file updates that are done by the Layout Editor toolset to the corresponding Eclipse project. As consequence developers do not have to constantly refresh their Eclipse project view against the file system.



Tools run as stand alone Application

The first user of the “Embedded Usage Mode” (see above) is the Layout Editor tool. All the logic of the tool up to now ran as web application of an explicit Tomcat instances. Now the Tomcat runs implicitly, as part of the editor client that you start.

Consequence: the tooling runs in its own, embedded Tomcat environment - the application that you develop is running in a separate instance that runs completely decoupled.

Layout Editor Extension Concept

Extension Concept

It is now possible to add own tools to the Layout Editor toolset. The tools are notified about certain events of the toolset (e.g. project loaded) and can access the project's file directory and the design time classloader that is built up for the project's classes.

The tool's that are added are shown in the tool area "on the right" of the Layout Editor. By following a certain directory concept each tools is living within its own directory of the Layout Editor toolset application.

We do not plan to build up an own tooling environment, in parallel to the normal development environment (e.g. Eclipse). The purpose of the extension concept is to allow tools that are required for layout purposes to be positioned next to the Layout Editor. A general tooling that is independent from layout management should be added directly to the development environment, e.g. as Eclipse plug-in.

Extension Kits

Based on the extension concept there are now first implementations that you can add to the CaptainCasa toolset.

- **Web Service Integration Kit** - This kit allows to directly load WSDL definitions into your project and generate the Axis2 stub code. In addition for each operation of the web service a corresponding call-template can be generated, that drastically simplified the calling of a web service from Java.
- **Bean Form Kit** - This kit allows to generate screens for Java Beans. In addition it allows to plug front end logic (e.g. mandatory properties, valid values resolving, ...) on top of the bean, using a generic approach based on Spring.
- **Touchscreen for ABAP Kit** - This kit allows to plug CaptainCasa user interface definitions directly in front of ABAP coding. Each user interface requires a corresponding remote function call on ABAP side. The main purpose of the kit is to provide touch screen user interface for ABAP based systems in a simple and efficient way - with all programming work on ABAP side.

Folgende Handelsmarken oder eingetragenen Markenzeichen werden innerhalb dieser Dokumentation referenziert:

Java - Sun Microsystems Inc.; UNIX - The Open Group; SAP, R/3, ABAP - SAP AG; Microsoft, Windows - Microsoft Corporation; BEA, WebLogic - Oracle Corporation; JBoss - Red Hat Middleware; IBM, Websphere - IBM Corporation; Tomcat - Apache Group

Alle anderen genannten Handelsmarken und Produkte sind Handelsmarken oder eingetragene Warenzeichen ihrer Eigentümer.

CaptainCasa GmbH

Hindemithweg 13
69245 Bammental

Tel +49 6223 484147

<http://www.CaptainCasa.com>
info@CaptainCasa.com