

Customization of Agent Home and Mini Client Displays

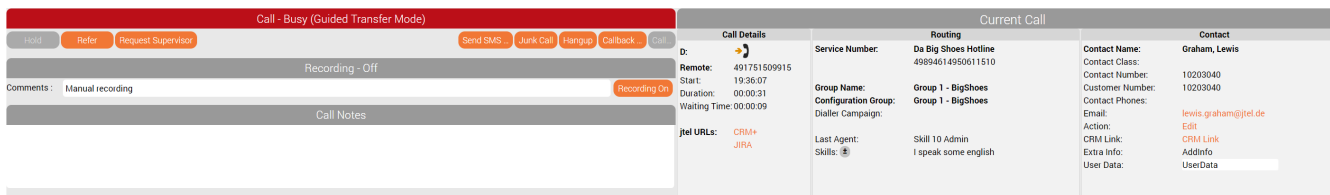
FROM RELEASE 3.34

From release 3.34 it is possible to customize the information displayed in Agent Home and the Mini Client when calls are received.

Layout

Agent Home

The new layout is shown in the next screenshot:



The call control / recording and notes functions have been moved to the left hand side.

All of the call data is now shown on the right hand side.

The three areas Call Details, Routing and Contact can be customized.

The portal contains default layouts for these areas, which result in the layout shown in the screenshot above.

Mini Client

The new layout of the mini client is shown in the next screenshot:

Call: Busy (Guided Transfer Mode)

Outbound :

Group 1 - BigShoes (356) ▾

Call...

Dialler :

▾

Start

Hold

Refer

Request Sup.

Callback ...

Hangup

Junk Call

Send SMS ...

Call Details

➔ 491751509915

Start / Duration:

19:36:07 / 00:03:02

Waiting Time:

00:00:09

Routing

Service Number:

Da Big Shoes Hotline

49894614950611510

Group:

Group 1 - BigShoes

Group 1 - BigShoes

Last Agent:

Skill 10 Admin

Skills: ±

I speak some english

Contact

Graham, Lewis

lewis.graham@jtel.de

CRM Link:

CRM Link

Extra Info:

AddInfo

User Data:

UserData

Call Notes

Recording - Off

Manual recording

The layout has been cleaned up to use up less space.

The three areas Call Details, Routing and Contact can be customized.

The portal contains default layouts for these areas, which result in the layout shown in the screenshot above.

Customization - Parameters

To customize the layout, the following parameters can be used:

Parameter	Defines
Layout.AgentHome.CurrentCall.CallDetails	The Agent Home "Call Details" area.
Layout.AgentHome.CurrentCall.Contact	The Agent Home "Contact" area.
Layout.AgentHome.CurrentCall.Routing	The Agent Home "Routing" area.
Layout.Miniclient.CurrentCall.CallDetails	The Mini Client "Call Details" area.
Layout.Miniclient.CurrentCall.Contact	The Mini Client "Contact" area.
Layout.Miniclient.CurrentCall.Routing	The Mini Client "Routing" area.

The parameters are either defined:

- **In ACD ... Groups ... Parameters** - in which case the layout for calls to a particular group will be affected.
- **In Client Master Data ... Parameters** - in which case the layout for all calls will be affected.

Settings made at the ACD group level override settings made at the client master data level when calls are processed. This allows for individualization on a per ACD group basis.



It is not necessary to define all six parameters. If no parameter is defined, in an ACD group, the system checks the client parameters, if no parameter is found here, the default layout is used.

The screenshot shows the definition of one parameter at the ACD Group level:

 Edit Client Parameter "Layout.AgentHome.CurrentCall.CallDetails"

ID :

123

Name :

Layout.AgentHome.CurrentCall.CallDetails

Value :

```
[cols="3,5"]
|===
|*#{acdmessages['AcdAgent.AcdAgentCallDirection']}:*
a|
|.callStatus#{acdUserStatusBean.currentRecord.AcdAgentCallClass}}
{nbsp}
|*#{acdmessages['AcdAgent.Remote']}:*
|
|#{(not empty acdUserStatusBean.currentRecord.ANumber and acdAgentCallHandlingBean.activeCallPresent) ? acdUserStatusBean.currentRecord.ANumber
: ( acdAgentCallHandlingBean.activeCallPresent ) ? messages['Common.ANumber.Unknown'] : ''}
|*#{acdmessages['AcdAgent.CallStart']}:
|#{dateTimeUtils.dateTimeToTime(acdUserStatusBean.currentRecord.dtCallStart)}
|*#{acdmessages['AcdAgent.CallDuration']}:
|#{dateTimeUtils.secondsToDuration(acdUserStatusBean.currentRecord.CallDuration)}
|*#{acdmessages['AcdAgent.WaitingTime']}:
|#{dateTimeUtils.secondsToDuration(acdUserStatusBean.currentRecord.WaitingTime)}
|{nbsp}
|{nbsp}
|*#{asciidocFunctions.nbsp(acdUserStatusBean.currentRecord.foreignSystemName)}:*
a|
```

Customization - Format

There are two parts to customization.

EL

The first part is understanding the syntax required to access data and functions within the portal.

This is based on EL - expression language.

Every expression in EL is contained between the following symbols:

Syntax	Means
<code>#{</code>	Start of expression
<code>}</code>	End of expression

EVERYTHING IS PARSED USING EL FIRST. This means, that the first step is to get the data you want being displayed, even if the formatting is horrible!

Examples, from the screenshot above:

Example	What it does
<code>{acdmessages['AcdAgent.Remote']}</code>	Gets the text for "Remote" in the current language for the user from the jtel portal standard texts.
<code>{asciidocFunctions.dateTimeToTime(acdUserStatusBean.currentRecord.dtCallStart)}</code>	Formats the field dtCallStart for the current call (which is a date time) using the function <code>asciidocFunctions.dateTimeToTime</code> as a time HH:MM:SS.

ASCIIDOC

Next, the template is passed to ASCIIDOC for formatting using the remaining markdown. At this stage, all of the EL expressions should have been "parsed out" and replaced with data, so only markdown should be left to be formatted.

How to use ASCIIDOC is explained here:

<https://docs.asciidoctor.org/asciidoc/latest/>

Useful jtel EL Functions

Function	Parameters	Description
<code>asciidocFunctions.dateTimeToTime</code>	DATETIME dtDateTime	Formats a datetime value to a time in HH:MM:SS format.

asciidocFunctions.secondsToDuration	INT nSeconds	Formats a number of seconds as a duration in HH:MM:SS format
asciidocFunctions.bold	STRING input	Surrounds the input with the ASCIIDOC markup for bold (the * character) but only if the input is not empty.
asciidocFunctions.italic	STRING input	Surrounds the input with the ASCIIDOC markup for italics (the _ character) but only if the input is not empty.
asciidocFunctions.monospace	STRING input	Surrounds the input with the ASCIIDOC markup for monospace (the ` character) but only if the input is not empty.
asciidocFunctions.highlight	STRING input	Surrounds the input with the ASCIIDOC markup for highlight (the # character) but only if the input is not empty.
asciidocFunctions.nbsp	STRING input	Either returns the input string if it is not blank, or an NBSP character (non breaking space) if it is empty. This is sometimes needed, because in html a space is not rendered unless absolutely necessary. Using an nbsp for empty strings forces the browser to actually render a space even when it is not deemed necessary.
asciidocFunctions.jsonField	STRING input STRING field	Given JSON data as input, extract the field from the data. This function is useful for parsing JSON data attached to the current call. JSON Data can be attached to the current call using the routing object Save addit. Info and Userdata. For example, given the following JSON data attached to the call: <pre>{ "name" : "Fred", "surname" : "Blogs" }</pre> The expression: <pre>#{asciidocFunctions:jsonField(acdUserStatusBean.currentRecord.JSONData, "name")}</pre> would return Fred.

Standard Definitions

Layout.AgentHome.CurrentCall.CallDetails

```

[cols="3,5"]
|===
|*#{acdmessages['AcdAgent.AcdAgentCallDirection']}:*
a|
|.callStatus#{acdUserStatusBean.currentRecord.AcdAgentCallClass}]
{nbsp}
|*#{acdmessages['AcdAgent.Remote']}:*
|
|#{(not empty acdUserStatusBean.currentRecord.ANumber and acdAgentCallHandlingBean.activeCallPresent) ? acdUserStatusBean.currentRecord.ANumber : (
acdAgentCallHandlingBean.activeCallPresent ) ? messages['Common.ANumber.Unknown'] : ''}
|#{acdmessages['AcdAgent.CallStart']}:
|#{asciidocFunctions.dateTimeToTime(acdUserStatusBean.currentRecord.dtCallStart)}
|#{acdmessages['AcdAgent.CallDuration']}:
|#{asciidocFunctions.secondsToDuration(acdUserStatusBean.currentRecord.CallDuration)}
|#{acdmessages['AcdAgent.WaitingTime']}:
|#{asciidocFunctions.secondsToDuration(acdUserStatusBean.currentRecord.WaitingTime)}
|{nbsp}
|{nbsp}
|*#{asciidocFunctions.nbsp(acdUserStatusBean.currentRecord.foreignSystemName)}:*
a|
|#{acdUserStatusBean.currentRecord.foreignSystemURL1}[#{acdUserStatusBean.currentRecord.foreignSystemURL1DisplayName},window=_blank]
|{nbsp}
a|
|#{acdUserStatusBean.currentRecord.foreignSystemURL2}[#{acdUserStatusBean.currentRecord.foreignSystemURL2DisplayName},window=_blank]
|===

```

Layout.AgentHome.CurrentCall.Contact

```

[cols="3,5"]
|===
|*#{acdmessages['AcdAgent.ContactName']}:*
|#{asciidocFunctions.bold(acdUserStatusBean.currentRecord.contactName)}
|#{acdmessages['AcdAgent.ContactClass']}:
|#{acdUserStatusBean.currentRecord.ContactClass}
|#{acdmessages['AcdAgent.ContactNumber']}:
|#{acdUserStatusBean.currentRecord.contactCustomerNumber}
|#{messages['ServiceNumbers.CustomerNumber']}:
|#{acdUserStatusBean.currentRecord.contactCustomerNumber}
|#{acdmessages['AcdAgent.ContactPhones']}:
|#{acdUserStatusBean.currentRecord.contactPhones}
|#{acdmessages['AcdAgentStatus.bEMail']}:
|#{acdAgentCallHandlingBean.email}
|#{messages['DataTable.Action']}:
a|
link:javascript:$('#M\\:AcdAgentEditDiallerContact').click();[#{messages['DataTable.Edit']}]
|#{acdmessages['AcdAgent.CRMLink']}:
a|
|#{acdUserStatusBean.CRMLink}[#{acdmessages['AcdAgent.CRMLink']],window=_blank]
|#{acdmessages['AcdAgent.AddInfo']}:
|#{acdAgentCallHandlingBean.readableAddInfo}
|#{loginBean.userDataCaption}:
a|
|.userDataEditor]
|#{acdUserStatusBean.currentRecord.UserData}
|===

```

Layout.AgentHome.CurrentCall.Routing

```

[cols="3,5"]
|===
|*#{acdmessages['AcdAgent.ServiceNumber']}:*
|#{asciidocFunctions.bold(acdUserStatusBean.currentRecord.ServiceNumbersName2)}
|{nbsp}
|#{acdUserStatusBean.currentRecord.ServiceNumbersName}
|{nbsp}
|{nbsp}
|*#{acdmessages['AcdAgent.AcdGroupsName']}:*
|#{asciidocFunctions.bold(acdUserStatusBean.currentRecord.AcdGroupsName)}
|*#{acdmessages['AcdAgent.ConfigurationGroup']}:*
|#{asciidocFunctions.bold(acdUserStatusBean.currentRecord.AcdConfigurationGroupsName)}
|#{acdmessages['AcdAgent.DiallerCampaignsName']}:
|#{acdUserStatusBean.currentRecord.DiallerCampaignsName}
|{nbsp}
|{nbsp}
|#{acdmessages['AcdAgent.LastAgent']}:
|#{acdUserStatusBean.currentRecord.LastAgent}
a|
|#{acdmessages['AcdAgent.Skills']}: [.agentHomeCurrentCallSkillChanger]#link:javascript:$('#M\\:AcdAgentChangeSkills').click();[#{messages['BTN.PlusMinus']}]#
|#{acdUserStatusBean.getCallSkills(1)}
|{nbsp}
|#{acdUserStatusBean.getCallSkills(2)}
|{nbsp}
|#{acdUserStatusBean.getCallSkills(3)}
|===

```

Layout.MiniClient.CurrentCall.CallDetails

```

[cols="35%,~,~"]
|===
|.2+a|
|.callStatus#{acdUserStatusBean.currentRecord.AcdAgentCallClass}]
=== #{{(not empty acdUserStatusBean.currentRecord.ANumber and acdAgentCallHandlingBean.activeCallPresent) ? acdUserStatusBean.currentRecord.ANumber : (
acdAgentCallHandlingBean.activeCallPresent ) ? messages['Common.ANumber.Unknown'] : ''}
|#{acdmessages['AcdAgent.CallStart']} / #{acdmessages['AcdAgent.CallDuration']}:
|#{asciidocFunctions.dateTimeToTime(acdUserStatusBean.currentRecord.dtCallStart)} / #{asciidocFunctions.secondsToDuration(acdUserStatusBean.currentRecord.
CallDuration)}
|#{acdmessages['AcdAgent.WaitingTime']}:
|#{asciidocFunctions.secondsToDuration(acdUserStatusBean.currentRecord.WaitingTime)}
|===

```

Layout.MiniClient.CurrentCall.Contact

```
[cols="35%,~"]
|===
|#{asciidocFunctions.bold(acdUserStatusBean.currentRecord.contactName)}
|#{acdAgentCallHandlingBean.email}
|#{acdmessages['AcdAgent.CRMLink']}:
a|
|#{acdUserStatusBean.CRMLink}|#{acdmessages['AcdAgent.CRMLink']],window=_blank]
|#{acdmessages['AcdAgent.AddInfo']}:
|#{acdAgentCallHandlingBean.readableAddInfo}
|#{loginBean.userDataCaption}:
a|
|.userDataEditor]
|#{acdUserStatusBean.currentRecord.UserData}
|===
```

Layout.MiniClient.CurrentCall.Routing

```
[cols="35%,~,~"]
|===
|#{acdmessages['AcdAgent.ServiceNumber']}:
|*#{acdUserStatusBean.currentRecord.ServiceNumbersName2}*
|#{acdUserStatusBean.currentRecord.ServiceNumbersName}
|#{acdmessages['AcdAgent.Tab.Transfer.Groups']}:
|#{asciidocFunctions.bold(acdUserStatusBean.currentRecord.AcdGroupsName)}
a|#{asciidocFunctions.bold(acdUserStatusBean.currentRecord.AcdConfigurationGroupsName)}
|#{not empty acdUserStatusBean.currentRecord.DiallerCampaignsName ? 'a|' : ''}
|#{not empty acdUserStatusBean.currentRecord.DiallerCampaignsName ? acdmessages['AcdAgent.DiallerCampaignsName'].concat(":") : ''}
|#{not empty acdUserStatusBean.currentRecord.DiallerCampaignsName ? '2.+a|' : ''}
|#{acdUserStatusBean.currentRecord.DiallerCampaignsName}
|#{acdmessages['AcdAgent.LastAgent']}:
2.+|#{acdUserStatusBean.currentRecord.LastAgent}
|#{acdmessages['AcdAgent.Skills']}: [.agentHomeCurrentCallSkillChanger]#link:javascript:${'#M\\:AcdAgentChangeSkills').click();[#{messages['BTN.PlusMinus']}]#
2.+|
|#{acdUserStatusBean.getCallSkills(1)}
|#{acdUserStatusBean.getCallSkills(2)}
|#{acdUserStatusBean.getCallSkills(3)}
|===
```