

Installation file .sopm

Installation file

The installation process contains a standard OTRS package installation file.

Basic installation file:

Basic installation doesn't differ much from the standard OTRS module installation if you do not plan to enable a multilanguage support mechanism.

```
<?xml version="1.0" encoding="utf-8" ?>
<otrs_package version="1.0">
  <Name>Contract Widget</Name>
  <Version>1.x.x</Version>
  <Framework>6.x.x</Framework>
  <Vendor>Company</Vendor>
  <URL>https://www.xyz.pl/</URL>
  <License>GNU AFFERO GENERAL PUBLIC LICENSE Version 3, November 2007</License>
  <Description Lang="en">XYZ Widget for Intalio Customer Panel.</Description>
  <IntroInstall Type="post" Lang="en" Title="Thank you!">Thank you for choosing the XYZ
Widget for Intalio Customer Panel module.</IntroInstall>
  <BuildDate>?</BuildDate>
  <BuildHost>?</BuildHost>

  <Filelist>
    <File Permission="644" Location="Kernel/Config/Files/XML/XYZWidget.xml"></File>
    <File Permission="644" Location="Kernel/Modules/XYZWidget.pm"></File>
    <File Permission="644" Location="Kernel/Language/pl_XYZ.pm"></File>

    <!-- FRONTEND -->
    <File Permission="644" Location="var/httpd/htdocs/customer-
panel/modules/WIDGET_XYZ.js"></File>

    <!-- TRANSLATION STRINGS -->
    <File Permission="644" Location="var/intalio-customer-
```

```
panel/xyz_translation_strings.json"/>
    </Filelist>
</otrs_package>
```

Multilanguage support:

Enabling additional language support requires adding additional translation strings to the global list. Include this code below in the installation file.

Script will look after every json file containing substring translation_strings.json!

The script workflow looks like that:

```
opening main translation_strings.json => looking for module's translations files => iterating over
list and appending translations which are not already in the main translation_strings.json
```

The translations are then correctly retrieved in the Customer Panel.

```
<CodeInstall Type="post"><![CDATA[

    ###
    # Adding new module's translations to the global phrase list of the otrs-frontend
module
    ###
    use JSON::PP;
    use Data::Dumper;

    my $TranslationStringsDir = '/opt/otrs/var/intalio-customer-panel/';

    my $MainObject = $Kernel::OM->Get('Kernel::System::Main');
    my $json = JSON::PP->new->pretty->allow_nonref;
    $json = $json->allow_blessed;
    $json = $json->allow_unknown;
    $json = $json->convert_blessed;

    my $MainTranslationStringsRef = $MainObject->FileRead(
        Location => '/opt/otrs/var/intalio-customer-panel/translation_strings.json',
    );

    my @MainTranslationStrings = @{$json->decode( ${$MainTranslationStringsRef} )};
```

```

opendir(DH, $TranslationStringsDir);
my @TranslationStringsFiles = readdir(DH);
closedir(DH);

foreach my $TranslationStringsFile (@TranslationStringsFiles) {
    # skip . and .. .. other than .json and other than main translation strings file
    next if($TranslationStringsFile =~ /^\.$/);
    next if($TranslationStringsFile =~ /^\.\.$/);
    next if(not $TranslationStringsFile =~ /\.json$/);
    next if($TranslationStringsFile =~ /^translation_strings\.json$/);

    print STDERR "-----\n";
    print STDERR "Opening and decoding file: $TranslationStringsFile\n";
    my $TranslationStringsRef = $MainObject->FileRead(
        Location => $TranslationStringsDir . $TranslationStringsFile,
    );

    my @TranslationStrings = @{$json->decode( ${$TranslationStringsRef} )};

    foreach my $TranslationString (@TranslationStrings) {
        my $IsTranslationStringAlreadyInMainTranslationStrings = 0;

        foreach my $MainTranslationString(@MainTranslationStrings) {
            $IsTranslationStringAlreadyInMainTranslationStrings = 1 if
($TranslationString eq $MainTranslationString);
        }

        if (not $IsTranslationStringAlreadyInMainTranslationStrings) {
            print STDERR "Adding string: $TranslationString\n";
            push(@MainTranslationStrings, $TranslationString);
        } else {
            print STDERR "String already in main translation strings:
$TranslationString\n";
        }
    }

    print STDERR "-----\n\n";
}

print STDERR "Final Main Translation Strings\n";
foreach my $MainTranslationString(@MainTranslationStrings) {

```

```

        print STDERR $MainTranslationString ."\n";
    }

    my $MainTranslationStringsResult = $json->encode( \@MainTranslationStrings );

    $MainObject->FileWrite(
        Directory => '/opt/otrs/var/intalio-customer-panel/',
        Filename  => "translation_strings.json",
        Content   => \$MainTranslationStringsResult,
    );

```

]]></CodeInstall>

<CodeReinstall Type="post"><![CDATA[

```

    ###
    # Adding new module's translations to the global phrase list of the otrs-frontend
module
    ###
    use JSON::PP;
    use Data::Dumper;

    my $TranslationStringsDir = '/opt/otrs/var/intalio-customer-panel/';

    my $MainObject = $Kernel::OM->Get('Kernel::System::Main');
    my $json = JSON::PP->new->pretty->allow_nonref;
    $json = $json->allow_blessed;
    $json = $json->allow_unknown;
    $json = $json->convert_blessed;

    my $MainTranslationStringsRef = $MainObject->FileRead(
        Location => '/opt/otrs/var/intalio-customer-panel/translation_strings.json',
    );

    my @MainTranslationStrings = @{$json->decode( ${$MainTranslationStringsRef} )};

    opendir(DH, $TranslationStringsDir);
    my @TranslationStringsFiles = readdir(DH);
    closedir(DH);

```

```

foreach my $TranslationStringsFile (@TranslationStringsFiles) {
    # skip . and .. .. other than .json and other than main translation strings file
    next if($TranslationStringsFile =~ /^\.$/);
    next if($TranslationStringsFile =~ /^\.\/$/);
    next if(not $TranslationStringsFile =~ /\.json$/);
    next if($TranslationStringsFile =~ /^translation_strings\.json$/);

    print STDERR "-----\n";
    print STDERR "Opening and decoding file: $TranslationStringsFile\n";
    my $TranslationStringsRef = $MainObject->FileRead(
        Location => $TranslationStringsDir . $TranslationStringsFile,
    );

    my @TranslationStrings = @{$json->decode( ${$TranslationStringsRef} )};

    foreach my $TranslationString (@TranslationStrings) {
        my $IsTranslationStringAlreadyInMainTranslationStrings = 0;

        foreach my $MainTranslationString(@MainTranslationStrings) {
            $IsTranslationStringAlreadyInMainTranslationStrings = 1 if
($TranslationString eq $MainTranslationString);
        }

        if (not $IsTranslationStringAlreadyInMainTranslationStrings) {
            print STDERR "Adding string: $TranslationString\n";
            push(@MainTranslationStrings, $TranslationString);
        } else {
            print STDERR "String already in main translation strings:
$TranslationString\n";
        }
    }

    print STDERR "-----\n\n";
}

print STDERR "Final Main Translation Strings\n";
foreach my $MainTranslationString(@MainTranslationStrings) {
    print STDERR $MainTranslationString ."\n";
}

my $MainTranslationStringsResult = $json->encode( \@MainTranslationStrings );

```

```

$MainObject->FileWrite(
    Directory => '/opt/otrs/var/intalio-customer-panel/',
    Filename  => "translation_strings.json",
    Content   => \$MainTranslationStringsResult,
);
]]></CodeReinstall>

```

```

<CodeUpgrade Type="post"><![CDATA[
    ###
    # Adding new module's translations to the global phrase list of the otrs-frontend
module
    ###
    use JSON::PP;
    use Data::Dumper;

    my $TranslationStringsDir = '/opt/otrs/var/intalio-customer-panel/';

    my $MainObject = $Kernel::OM->Get('Kernel::System::Main');
    my $json = JSON::PP->new->pretty->allow_nonref;
    $json = $json->allow_blessed;
    $json = $json->allow_unknown;
    $json = $json->convert_blessed;

    my $MainTranslationStringsRef = $MainObject->FileRead(
        Location => '/opt/otrs/var/intalio-customer-panel/translation_strings.json',
    );

    my @MainTranslationStrings = @{$json->decode( ${$MainTranslationStringsRef} )};

    opendir(DH, $TranslationStringsDir);
    my @TranslationStringsFiles = readdir(DH);
    closedir(DH);

    foreach my $TranslationStringsFile (@TranslationStringsFiles) {
        # skip . and .. .. other than .json and other than main translation strings file
        next if($TranslationStringsFile =~ /^\.$/);
        next if($TranslationStringsFile =~ /^\.\.$/);
        next if(not $TranslationStringsFile =~ /\.json$/);
        next if($TranslationStringsFile =~ /^translation_strings\.json$/);
    }

```

```

print STDERR "-----\n";
print STDERR "Opening and decoding file: $TranslationStringsFile\n";
my $TranslationStringsRef = $MainObject->FileRead(
    Location => $TranslationStringsDir . $TranslationStringsFile,
);

my @TranslationStrings = @{$json->decode( @{$TranslationStringsRef} )};

foreach my $TranslationString (@TranslationStrings) {
    my $IsTranslationStringAlreadyInMainTranslationStrings = 0;

    foreach my $MainTranslationString(@MainTranslationStrings) {
        $IsTranslationStringAlreadyInMainTranslationStrings = 1 if
($TranslationString eq $MainTranslationString);
    }

    if (not $IsTranslationStringAlreadyInMainTranslationStrings) {
        print STDERR "Adding string: $TranslationString\n";
        push(@MainTranslationStrings, $TranslationString);
    } else {
        print STDERR "String already in main translation strings:
$TranslationString\n";
    }
}

print STDERR "-----\n\n";
}

print STDERR "Final Main Translation Strings\n";
foreach my $MainTranslationString(@MainTranslationStrings) {
    print STDERR $MainTranslationString ."\n";
}

my $MainTranslationStringsResult = $json->encode( \@MainTranslationStrings );

$MainObject->FileWrite(
    Directory => '/opt/otrs/var/intalio-customer-panel/',
    Filename => "translation_strings.json",
    Content => \$MainTranslationStringsResult,
);

```

```
]]></CodeUpgrade>
```

Revision #5

Created 13 July 2020 13:55:34 by editor

Updated 11 August 2020 13:37:21 by editor