

# Service components

## Cases

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

# Table of Contents

<b>Cases</b>	<b>4</b>
Introduction	4
<b>Adding a new rule for calculating case deadline</b>	<b>4</b>
Case description	4
Source code of the case:	4
Case implementation algorithm	4

# Cases

 Advanced

## Introduction

Creatio enables implementing custom logic of receiving parameters for calculating case deadline. When calculating or recalculating a case deadline, a developer implemented strategy is used instead of one of the base calculation strategies.

You can select a specific calculation rule in the [ *Case deadline calculation rules* ] lookup. Follow these steps to add a new calculation rule:

1. Create an object schema and add columns necessary for storage of response and resolution deadlines, links to the calendar, service agreement and service.
2. Based on the created object schema, add a lookup and populate it with values needed to calculate the deadline parameters.
3. Add the source code schema and declare the class inherited from the `BaseTermStrategy` abstract class. Implement custom mechanism of receiving response and resolution deadline parameters in the class.
4. Add a new rule.

## Adding a new rule for calculating case deadline

 Medium

### Case description

Add a custom rule for calculating case deadline parameters for the [ *Lost data recovery* ] service as per the [ *78 – Elite Systems* ] agreement. Set the following values for the new rule:

- response time - 2 working hours
- resolution time - 1 working day
- used calendar - [Default calendar]

### Source code of the case:

You can download the package with case implementation using the following [link](#).

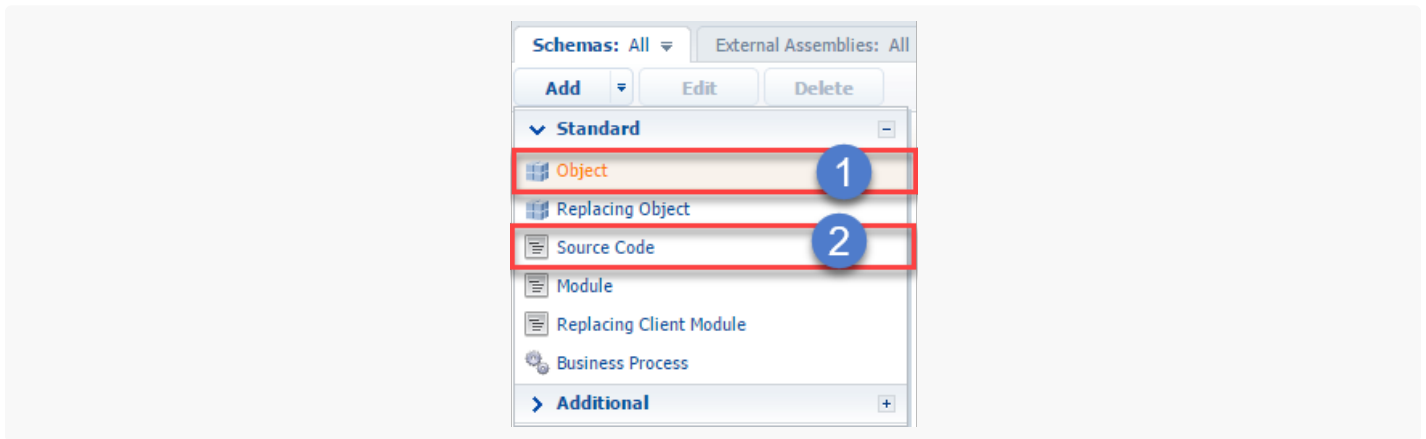
### Case implementation algorithm

1. Creating an object schema containing the necessary columns for

## calculation

Perform the [ Add ] - [ Object ] action on the [ Schemas ] tab of the [ Configuration ] section.

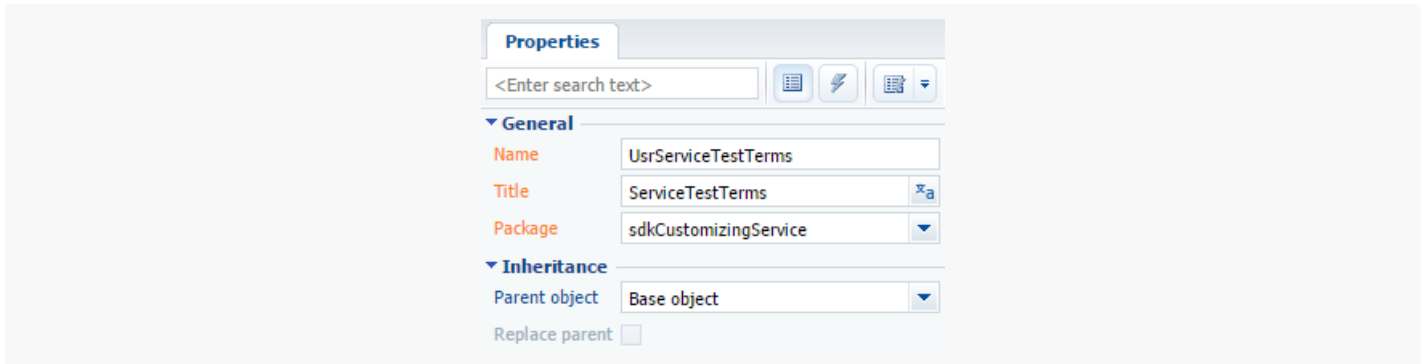
Fig. 1. Adding the schema



Set the following properties for the created object schema (Fig. 2):

- [Name] - "UsrServiceTestTerms"
- [Title] - "ServiceTestTerms"
- [Parent object] - the [Base object] schema

Fig. 2. Properties of the added object schema



In the created schema, create a number of columns, whose primary properties are listed in table 1.

Table 1. Properties of the added columns

Name	Title	Type	Description
UsrReactionTimeUnit	Response time unit	The [Time unit] lookup	Specifies the time unit (calendar days, hours, etc.) that will be used for calculating the [Response time] parameter.
UsrReactionTimeValue	Response time value	Integer	A column for storage the response time value.
UsrSolutionTimeUnit	Response time unit	The [Time unit] lookup	Specifies the time unit (calendar days, hours, etc.) that will be used for calculating the [Response time] parameter.
UsrSolutionTimeValue	Resolution time	Integer	A column for storage the response time value.
UsrCalendarId	Calendar that is used	The [Calendar] lookup	The calendar used for calculating the case deadline.
UsrServicePactId	Service agreement	The [Service agreement] lookup	Link to the [Service agreement] object. Added for enabling filtration.
UsrServiceItemId	Service	The [Service] lookup	Link to the [Service] object. Added for enabling filtration.

Publish the schema after adding the columns.

## 2. Adding a lookup and populating it with values needed to calculate the deadline parameters

Provide specific values to calculate the case response and resolution deadline. To do this, add a lookup with the following values based on the added schema (fig.3):

- [Name] - "Custom response and resolution deadlines"
- [Object] - ServiceTestTerms

Fig. 3. Properties of the added lookup

Name *	Custom response and resolution deadlines
Object *	ServiceTestTerms
List page	
Description	

Add a record with the following data to the added lookup (as per the case conditions) (fig.4):

Fig. 4. A record in the created lookup that meets the case conditions

Response time un...	Respo...	Resolve time unit	Resolu...	Service	Calendar that is us...	Service agreement
Calendar hours	2	Calendar days	1	Lost data recovery	Default calendar	78 — Elite Syste...

### 3. Implementing a class with the mechanism of receiving deadline parameters

Add the source code schema (fig.1, 2) Add the class inherited from the `BaseTermStrategy` abstract class (declared in the `Calendar` package) to the schema source code. Implement a parameterized constructor with the following parameters in the class:

- `UserConnection userConnection` – user current connection
- `Dictionary args,>` – arguments that are the base of performing calculation

Implement the `GetTermInterval()` abstract method declared in the base class. This method accepts the mask of populated values as the incoming parameter, which is the base of taking a decision about populating the specific deadline parameters of the `TermInterval` returned class implementing the `ITermInterval` interface.

The complete schema source code:

```
namespace Terrasoft.Configuration
{
    using System;
    using System.Collections.Generic;
    using Terrasoft.Common;
    using Terrasoft.Configuration.Calendars;
    using Terrasoft.Core;
```

```

using Terrasoft.Core.Entities;
using CalendarsTimeUnit = Calendars.TimeUnit;
using SystemSettings = Terrasoft.Core.Configuration.SysSettings;
public class ServiceTestTermsStrategy: BaseTermStrategy<CaseTermInterval, CaseTermStates>
{
    // Container class for storage of data received from the entrance point.
    protected class StrategyData
    {
        public Guid ServiceItemId {
            get;
            set;
        }
        public Guid ServicePactId {
            get;
            set;
        }
    }
    // The field for storage of data received from the entrance point.
    protected StrategyData _strategyData;
    // Parameterized constructor necessary for the correct
    // initialization by selector class.
    public ServiceTestTermsStrategy(UserConnection userConnection, Dictionary<string, object
        : base(userConnection) {
        _strategyData = args.ToObject<StrategyData>();
    }
    // Method that receives data and returns them in the CaseTermInterval class instance.
    public override CaseTermInterval GetTermInterval(CaseTermStates mask) {
        var result = new CaseTermInterval();
        // Creating the EntitySchemaQuery query.
        var esq = new EntitySchemaQuery(UserConnection.EntitySchemaManager, "UsrServiceTest")
        // Adding columns to the query.
        string reactionTimeUnitColumnName = esq.AddColumn("UsrReactionTimeUnit.Code").Name;
        string reactionTimeValueColumnName = esq.AddColumn("UsrReactionTimeValue").Name;
        string solutionTimeUnitColumnName = esq.AddColumn("UsrSolutionTimeUnit.Code").Name;
        string solutionTimeValueColumnName = esq.AddColumn("UsrSolutionTimeValue").Name;
        string calendarColumnName = esq.AddColumn("UsrCalendarId.Id").Name;
        // Adding filters to the query.
        esq.CreateFilterWithParameters(FilterComparisonType.Equal, "UsrServiceItemId", _stra
        esq.CreateFilterWithParameters(FilterComparisonType.Equal, "UsrServicePactId", _stra
        // Execution and processing of query results.
        EntityCollection entityCollection = esq.GetEntityCollection(UserConnection);
        if (entityCollection.IsNotEmpty()) {
            // Adding response time to the nurtured value.
            if (!mask.HasFlag(CaseTermStates.ContainsResponse)) {
                result.ResponseTerm = new TimeTerm {
                    Type = entityCollection[0].GetTypedColumnValue<CalendarsTimeUnit>(reacti
                    Value = entityCollection[0].GetTypedColumnValue<int>(reactionTimeValueCc
                    CalendarId = entityCollection[0].GetTypedColumnValue<Guid>(calendarColu
                };
            }
        }
    }
}

```





Fig. 6. Case result

Case #SR00000004

What can I do for you? >

Creatio

SAVE CANCEL ACTIONS TAKE IT VIEW

Resolution time  
4/12/2017 12:22 PM 1d 00:00

Priority  
↑ Medium

Contact

Account  
Elite Systems

SLA  
78 — Elite Systems

Category  
Incident

Service

Configuration item

Assignees group

New In progress Waiting for response Resolved Closed

NEXT STEPS (0)

PROCESSING CLOSURE AND FEEDBACK CASE INFORMATION ATTACHMENTS FEED

Subject \* Lost data

Description

Source Call

Support line 1st-line support

Terms

Registration date 4/11/2017 12:22 PM

Response time 4/11/2017 2:22 PM

Resolution time 4/12/2017 12:22 PM

Actual response time

Actual resolution time

Remaining: 02:00 Remaining: 1d 00:00