

Примеры

- [Обработка входящего Договора](#)
- [Обработка исходящего Договора с физ. лицом](#)
- [Отправка документа сценарием на подписание в Диадок](#)
- [Добавление скан-копии](#)
- [Автоматический переход в процессной задаче](#)

Обработка входящего Договора

В веб дизайнера ELMA4 (в ELMA3 обычном дизайнера) реализуем Глобальный модуль, где определяем нужную точку расширения, например, **ReceivableContract**.

The screenshot shows the 'Дизайнер' (Designer) interface with a search bar and a breadcrumb trail: 'Все сценарии > R&M HMS'. The main content area is titled 'Глобальный модуль "R&M HMS"'. On the left, there is a 'Настройки' (Settings) sidebar. The main configuration area includes:

- Название ***: R&M HMS
- Пространство имен**: HMS.Configuration
- Описание**: (empty text area)
- Структура модуля**:
 - Корневая папка модуля
 - Contracts
 - Diadoc
 - SendableRNMOOutContractSubLic
 - ReceivableContract
 - DiadocIntegration
 - ReceivableAdditionalAgreement
 - SendableRNMOOutContractInd
 - SendableRNMOOutContractLic

Реализация:

```
using System;
using EleWise.ELMA.ComponentModel;
using EleWise.ELMA.Documents.Contracts.Models;
using EleWise.ELMA.Model.Services;
using ITino.ELMA.Documents.Diadoc.Components;
using ITino.ELMA.Documents.Diadoc.Models;

namespace HMS.Configuration.Diadoc
```

```

{
    /// <inheritdoc />
    [Component]
    public class ReceivableContract : ReceivableDiadocDocument
    {
        /// <inheritdoc />
        public override bool Enabled
        {
            /// Точка расширения включена и обрабатывается
            get { return true; }
        }

        /// <inheritdoc />
        public override DiadocDocumentType DocumentType
        {
            /// Какой тип документа Диадока обрабатываем
            get { return DiadocDocumentType.Contract; }
        }

        /// <inheritdoc />
        public override Guid TypeUid
        {
            /// К какому типу документа в ELMA привязываем
            get { return InterfaceActivator.UID<IContractIncoming>(); }
        }
    }
}

```

Атрибут [Component] обязателен.

Обработка исходящего Договора с физ. лицом

В веб дизайнера ELMA4 (в ELMA3 обычном дизайнера) реализуем Глобальный модуль, где определяем нужную точку расширения, например, **SendableRNMOContractInd**.

The screenshot shows the 'Дизайнер' (Designer) interface with a search bar and a dropdown menu for 'Сценарии' (Scenarios). The current scenario is 'R&M HMS'. The main area displays the configuration for the 'Глобальный модуль "R&M HMS"'. The configuration is divided into 'Настройки' (Settings) and 'Структура модуля' (Module Structure). The 'Настройки' section includes fields for 'Название*' (Name), 'Пространство имен' (Namespace), and 'Описание' (Description). The 'Структура модуля' section shows a tree view of the module structure, with 'SendableRNMOContractInd' highlighted in red.

Настройки	Название *	R&M HMS
	Пространство имен	HMS.Configuration
	Описание	
	Структура модуля	
	▾ Корневая папка модуля	
	▸ Contracts	
	▾ Diadoc	
	SendableRNMOContractSubLic	
	ReceiveableContract	
	DiadocIntegration	
	ReceiveableAdditionalAgreement	
	<u>SendableRNMOContractInd</u>	
	SendableRNMOContractLic	
	▸ Products	

Реализация:

```
using System;
using EleWise.ELMA.ComponentModel;
using EleWise.ELMA.ConfigurationModel;
using EleWise.ELMA.Documents.Models;
using EleWise.ELMA.CRM.Models;
using EleWise.ELMA.Model.Services;
```

```

using EleWise.ELMA.Services;
using ITino.ELMA.Documents.Diadoc.Components;
using ITino.ELMA.Documents.Diadoc.Exceptions;
using ITino.ELMA.Documents.Diadoc.Managers;
using ITino.ELMA.Documents.Diadoc.Models;
using ITino.ELMA.Documents.Diadoc.Services;

namespace HMS.Configuration
{
    /// <inheritdoc />
    [Component]
    public class SendableRNMOutContractInd : SendableDiadocDocument
    {
        /// <inheritdoc />
        public override bool Enabled
        {
            /// Точка расширения включена и обрабатывается
            get { return true; }
        }

        /// <inheritdoc />
        public override DiadocDocumentType DocumentType
        {
            /// Какой тип документа Диадока обрабатываем
            get { return DiadocDocumentType.Contract; }
        }

        /// <inheritdoc />
        public override Guid TypeUId
        {
            /// К какому типу документа в ELMA привязываем
            get { return InterfaceActivator.UID<RNMOutContractInd>(); }
        }

        public override IContractor GetSender(IDocument doc)
        {
            /// Получить отправителя документа
            [var contract = doc as RNMOutContractInd;
            return contract != null ? contract.Contract : null;
            }
    }
}

```

□□

```
/// <inheritdoc />
public override EODiadocCanSendStatus CanSendDocument(IDocument doc)
{
    □// Проверить, можно ли отправить документ в Диадок
    if (doc == null)
        return EODiadocCanSendStatus.DocIsNull;
    var service = Locator.GetServiceNotNull<IDIadocService>();
    var contract = doc as RNMOutContractInd;
    if (contract != null)
        return service.CanSendDocument(contract.ContractId, contract.Executor);
    return base.CanSendDocument(doc);
}

/// <inheritdoc />
public override ICODiadocDocument Send(IDocument doc, Guid? departmentId, Guid?
userId, string comment = null)
{
    □// Отправить документ в Диадок на подписание нами
    var contract = doc as RNMOutContractInd;
    if (contract == null)
        throw new DiadocException("Договор не может быть null");
    var edata = DiadocDocumentManager.Instance.CheckContractors(contract.ContractId,
contract.Executor, null, null);
    var result = edata.Service.SendContract(edata.FormBoxId, edata.ToBoxId, doc,
contract.Number, contract.ContractDate.Value, "ГПХ/Физ.лицо/Самозанятый", null, departmentId,
userId, comment);
    return DiadocDocumentManager.Instance.SendDocHistory(result, doc,
contract.ContractId, contract.Executor, contract.Number, contract.ContractDate);
}
}
}
```


Атрибут [Component] обязателен.

Отправка документа сценарием на подписание в Диадок

TODO

Добавление скан-копии

В веб дизайнера ELMA4 (в ELMA3 обычном дизайнера) реализуем Глобальный модуль, где определяем нужную точку расширения, например, **DiadocIntegration**.

 **Дизайнер** Сценарии ▾

[Все сценарии](#) [R&M HMS](#) ✕

Глобальный модуль "R&M HMS"

Настройки	Название *	<input type="text" value="R&M HMS"/>
	Пространство имен	<input type="text" value="HMS.Configuration"/>
	Описание	<input type="text"/>

Структура модуля

- ▾ Корневая папка модуля
 - Contracts
 - ▾ Diadoc
 - [SendableRNMOutContractSubLic](#)
 - [ReceivableContract](#)
 - [DiadocIntegration](#)
 - [ReceivableAdditionalAgreement](#)

Реализация:

```
using EleWise.ELMA.ComponentModel;  
using EleWise.ELMA.Documents.Models;  
using EleWise.ELMA.Files;  
using ITino.ELMA.Documents.Diadoc.Extensions;  
using ITino.ELMA.Documents.Models;  
using ITino.ELMA.Documents.Managers;  
  
namespace HMS.Configuration.Diadoc  
{  
    [Component]
```


Автоматический переход в процессной задаче

Начиная с 4.0.28 и 3.15.48

При обработке документов Диадока в процессе может потребоваться отслеживать состояние подписания (как для входящих, так и исходящих). Чтобы задача автоматически выполнялась по нужному переходу при наступлении события по документу в Диадоке, необходимо реализовать в сценарии процесса следующий код:

```
using ITino.ELMA.Diadoc.Helpers;

public override void OnTaskCreate(ITaskBase task, Context context)
{
    // Где,
    // 13b27a4a-b94f-4bc9-9390-d7ff4fcb58bd - идентификатор задачи на диаграмме процесса
    // 309f52b9-087d-4971-9fc0-9cbffc61c737 - идентификатор перехода в случае, если документ
    Подписан
    task.LinkDiadocEvents(new Guid("13b27a4a-b94f-4bc9-9390-d7ff4fcb58bd"),
        context.DiadocDocument, new Guid("309f52b9-087d-4971-9fc0-9cbffc61c737"));
}
```

Указанный пример обрабатывает событие **Подписан**. При необходимости можно указать переходы для событий **Отклонен** и **Удален**.