

DIRECTUM. Рекомендации по разработке модулей расширения

Назначение документа

Документ содержит описание средств, требования, а также рекомендации по разработке модулей расширения. Предназначен для разработчиков, модифицирующих систему в соответствии с бизнес-процессами организации.

Подробное описание свойств и методов приведено в документе «DIRECTUM. Описание интерфейсов модулей расширения», входит в комплект документации.

Общие сведения

Модуль расширения – это библиотека, предназначенная для дополнения стандартной функциональности системы DIRECTUM. Состав подключаемых модулей расширения может быть разным. На текущий момент поддерживаются следующие возможности:

- [наблюдение за открытыми документами](#)
- [проверка орфографии](#)
- [интеграция с системами мгновенных сообщений](#)
- [шифрование и подписание](#)

Требования к разработке

- Платформа с поддержкой dispatch-интерфейсов. Рекомендуется использовать Embarcadero Delphi или .NET. Подробнее см. статью [Dispatch Interfaces](#);
- Работа модуля должна быть организована без использования run-time пакетов (исключить при работе в Delphi или C++ Builder).

Разработка модулей расширения

Модуль для наблюдения за открытыми документами

Создание модуля

1. В среде разработки создайте проект для реализации библиотеки модуля расширения с учетом [требований к разработке](#).
2. Импортируйте библиотеку типов SBPluginInterfaceLibrary.tlb.
3. Создайте класс, реализующий интерфейс **IDocumentObserverPlugin**.

4. Создайте класс, реализующий интерфейс **IDocumentObserver**.

Механизм работы модуля

1. Загрузка модуля расширения выполняется автоматически и происходит дважды: для проверки окружения **IPlugin.CheckEnvironment** и реальной работы. Необходимо учитывать, если предполагается выполнение каких-либо действий, например инициализации, что отнимет дополнительное время при загрузке модуля.
2. Составление списка имеющихся модулей наблюдения. Происходит при открытии документа.

Библиотеку модуля расширения следует располагать в папке:

<Папка установки клиентской части IS-Builder>\Plugins\<<Тип модуля расширения>\<Имя модуля расширения>\

где <Тип модуля расширения>:

- **Encryption** – для модулей расширения шифрования/подписания;
- **SpellCheck** – для модулей расширения проверки орфографии;
- **IM** – для модулей интеграции с системами мгновенных сообщений;
- **DocumentObserver** – для модулей наблюдения за открытыми документами.
- <Имя модуля расширения> – любое уникальное в рамках директории имя.

3. Поочередно у каждого модуля вызывается метод **IDocumentObserverPlugin.SupportsExtension**.

Если метод возвращает значение **True**, то перебор списка прекращается и первый найденный подходящий плагин назначается в качестве обработчика.

4. Для выбранного плагина вызывается метод **GetObserver**, который должен вернуть класс наблюдателя за документом, реализующий интерфейс **IDocumentObserver**.
5. Для полученного класса вызывается метод **OpenFile**.
6. Для полученного класса вызывается метод **WaitFile**.

Открытие каждого документа происходит в отдельном новом потоке. Поток существует до тех пор, пока не завершится выполнение метода **WaitFile**. Только после этого закрытие документа будет считаться корректным и изменения документа, если они были, будут сохранены в базе данных на сервере.

Модуль для проверки орфографии

Средства проверки орфографии:

- проверка корректности написания слов;
- составление списка вариантов замены ошибочных слов;
- подключение дополнительных словарей.

Ошибочными считаются слова, не найденные в словарях модуля.

Создание модуля

1. В среде разработки подготовьте проект для создания библиотеки модуля расширения с учетом [требований к разработке](#).

2. Из папки с установленной клиентской частью системы DIRECTUM импортируйте библиотеку типов SBPluginInterfaceLibrary.tlb в созданный проект.
3. Создайте класс, реализующий интерфейс **ISpellCheckerPlugin**.
4. Если планируется использование пользовательских словарей проверки орфографии, то создайте интерфейс **ISpellCheckerPlugin2**.

Подробнее о стандартных модулях и используемых словарях см. в руководстве администратора, в главе «Настройка модулей системы DIRECTUM», раздел «Словари стандартных модулей», входит в комплект документации.

Механизм работы модуля

1. Загрузка модуля расширения выполняется автоматически и происходит дважды: для проверки окружения **IPlugin.CheckEnvironment** и реальной работы. Необходимо учитывать, если предполагается выполнение каких-либо действий, например инициализации, что отнимет дополнительное время при загрузке модуля.
2. Проверка орфографии:
 - a) вызов метода **ISpellCheckerPlugin.CheckSpelling**. – проверка текста и формирование списка ошибок;
 - b) при последовательной обработке орфографических ошибок вызывается метод **ISpellCheckerPlugin.GetSuggestions** – формирование списка замен для каждого ошибочного слова.
3. Если поддерживается работа с пользовательскими словарями (**ISpellCheckerPlugin2.CanAddSpelling** возвращает значение **True**), то вызываются методы добавления или удаления слов из словаря.

Модуль для интеграции с системами мгновенных сообщений

Модуль предназначен для интеграции систем мгновенных сообщений и системы DIRECTUM. Например:

- просмотр статусов участников задачи, отображаемых в дереве задачи;
- отправка мгновенного сообщения участнику задачи;
- проведение конференции из карточки совещания;
- отправка мгновенного сообщения пользователю с просьбой освободить заблокированный документ.

Создание модуля

1. В среде разработки создайте проект для реализации библиотеки модуля расширения с учетом [требований к разработке](#).
2. Создайте новую библиотеку типов и подключите к ней библиотеку типов SBPluginInterfaceLibrary.tlb.
3. Создайте класс, реализующий интерфейс **IMessagingPlugin** или любого его наследника.
4. Создайте класс, реализующий интерфейс **IContact** или любого его наследника.
5. Создайте класс, реализующий интерфейс **IStatus** или любого его наследника.
6. Создайте класс, реализующий интерфейс **IConversation** или любого его наследника.

Примечание

Реализовывать интерфейс **IGroup** не требуется. Данный интерфейс устарел и оставлен для сохранения обратной совместимости.

7. Реализуйте и экспортируйте функцию **GetInterface**.
8. Запросите ключ системы с поддержкой модуля расширения. Подробнее см. раздел [«Лицензирование»](#). Зарегистрируйте ключ в целевой системе.
9. Запустите целевую систему.

Механизм работы модуля

1. [Загрузка модуля расширения](#).
2. [Лицензирование](#)
3. [Получение контактов](#)
4. [Работа со статусами](#)
5. [Инициация бесед](#)

Загрузка модуля расширения

Загрузка модуля расширения выполняется автоматически и происходит дважды: для проверки окружения **IPlugin.CheckEnvironment** и реальной работы. Необходимо учитывать, если предполагается выполнение каких-либо действий, например инициализации, что отнимет дополнительное время при загрузке модуля.

Лицензирование

Все модули расширения интеграции с системами мгновенных сообщений подлежат лицензированию. Для работы с модулем расширения необходимо установить ключ системы с поддержкой данного модуля. Чтобы его получить:

1. Запросите новый ключ системы DIRECTUM. При обращении в службу поддержки с запросом ключа укажите, что необходимо добавить поддержку нового модуля расширения интеграции с системами мгновенных сообщений. При этом обязательно укажите имя нового модуля - **IPlugin.Name**.
2. На основе имени модуля сотрудник службы поддержки сгенерирует код модуля для лицензирования.
3. Сгенерируйте ключ с поддержкой модуля с использованием данного кода.
4. Установите в системе ключ. В результате станет доступна работа с новым модулем расширения.

Если необходима установка данного модуля расширения в других системах, то для них также потребуется запросить ключи с поддержкой этого модуля. При запросе ключа для них в обращении можно указывать уже полученный код модуля расширения, что позволит сократить время на генерацию ключа.

Получение контактов

При получении контакта **IMessagingPlugin.GetContact** указывается логин пользователя SQL-сервера и признак аутентификации (Windows- или SQL-аутентификация). При Windows-аутентификации логин пользователя включает в себя имя домена и имя пользователя.

Способ сопоставления логина с именем контакта определяется самостоятельно. Например, можно использовать информацию из Active Directory или каких-то других специализированных каталогов.

Работа со статусами

Поддерживается произвольный набор статусов для контактов. Обязательным является только специальный статус «Состояние присутствия не определено». Данный статус будет возвращен, если полученные данные о контакте некорректны **IContact.Valid = False**. Например, если не удалось установить связь с системой мгновенных сообщений.

Для значков статусов контактов следует использовать изображения, корректно обрабатываемые браузером Internet Explorer. Рекомендуемый размер значков составляет от 10*10 до 16*16 пикселей.

Инициация бесед

При создании беседы допускается создавать неинициализированный объект беседы **IConversation.Initialized = False**. В этом случае периодически будет поступать запрос на предмет инициализации. Сообщения будут отправлены только после инициализации беседы **IConversation.Initialized = True**.

Модуль для расширения шифрования и подписания

Модуль предназначен для использования в системе DIRECTUM шифрования текстов документов, задач, заданий и уведомлений, а также для установки ЭП на данные объекты.

Создание модуля

1. В среде разработки создайте проект для реализации библиотеки модуля расширения с учетом [требований к разработке](#).
2. Из папки с установленной клиентской частью системы DIRECTUM импортируйте библиотеку типов SBPluginInterfaceLibrary.tlb в созданный проект.
3. Создайте класс, реализующий интерфейс **IEncryptionPlugin**.
4. Создайте класс, реализующий интерфейс **ICertificate**.
5. Добавьте реализацию **ICertificate2**. При этом могут использоваться и другие классы, не реализующие интерфейс **ICertificate**. При его отсутствии IS-Builder будет работать с интерфейсом **ICertificate**.
6. Если планируется работа с хранилищем сертификатов, то создайте класс, реализующий интерфейс **ICertificateStorage**.

Механизм работы модуля

1. Загрузка модуля расширения выполняется автоматически и происходит дважды: для проверки окружения **IPlugin.CheckEnvironment** и реальной работы. Необходимо учитывать, если предполагается выполнение каких-либо действий, например инициализации, что отнимет дополнительное время при загрузке модуля.
2. Шифрование и/или подписание. Поддержка шифрования является опциональной и должна быть реализована, если свойство **IEncryptionPlugin.CanEncrypt** возвращает значение **True**.

3. Использование хранилища сертификатов. Поддержка является опциональной и должна быть реализована, если свойство **IEncryptionPlugin.IsCertificateStorageSupported** возвращает значение **True**.

Модуль файлового хранилища

Модуль предназначен для поддержки различных типов файловых хранилищ в системе DIRECTUM.

Создание модуля

1. В среде разработки создайте проект для реализации библиотеки модуля расширения с учетом [требований к разработке](#).
2. Импортируйте библиотеку типов SBPluginInterfaceLibrary.tlb.
3. Создайте класс, реализующий интерфейс **IFileStoragePlugin**.
4. Если файловое хранилище поддерживает работу с теневыми копиями, создайте класс, реализующий интерфейс **IShadowCopySupportPlugin**.

Механизм работы модуля

1. Загрузка модуля расширения выполняется автоматически и происходит дважды: для проверки окружения **IPlugin.CheckEnvironment** и реальной работы. Необходимо учитывать, если предполагается выполнение каких-либо действий, например инициализации, что отнимет дополнительное время при загрузке модуля.
2. При работе с документами служба файловых хранилищ вызывает методы интерфейса **IFileStoragePlugin**.
3. Для работы с теневыми копиями служба файловых хранилищ вызывает методы интерфейса **IShadowCopySupportPlugin**. Не все файловые хранилища поддерживают работу с теневыми копиями, поэтому данный интерфейс опционален.