

# Инструкция по эксплуатации Flexberry ORM

## 1. Общие сведения

Flexberry ORM — технологический компонент, реализующий уровень доступа к данным с помощью механизма объектно-реляционного отображения (ORM). Он предназначен для упрощения взаимодействия объектов .NET-приложений с реляционными базами данных.

## 2. Архитектура и компоненты, используемые при эксплуатации

При эксплуатации Flexberry ORM основную роль играют два аспекта:

- **Компоненты времени выполнения (Runtime Components):** это библиотеки, представленные в виде NuGet-пакета NewPlatform.Flexberry.ORM. Они встраиваются в конечное приложение и обеспечивают непосредственную работу ORM.
- **Сгенерированные или созданные вручную классы объектов данных:** эти классы, представляющие сущности предметной области, используются для взаимодействия с базой данных через ORM. Классы могут быть сгенерированы с помощью Flexberry Designer или созданы вручную.

## 3. Настройка и конфигурация

Для корректной эксплуатации Flexberry ORM необходимо:

- Настроить подключение к базе данных. Это делается через объект DataService, который указывает используемый провайдер данных (например, для MS SQL Server, PostgreSQL и т.д.) и строку подключения.
- Убедиться, что сгенерированные или созданные классы объектов данных правильно отражают структуру базы данных и связаны с соответствующими таблицами и полями через атрибуты отображения.
- При необходимости настроить представления (Views), которые определяют подмножество свойств сущностей, используемых в конкретных операциях (например, для отображения в списке).

## 4. Принципы работы и примеры эксплуатации

Основные действия при работе с Flexberry ORM:

**Создание, обновление, удаление объектов:**

```
// Создание нового объекта
Страна новаяСтрана = new Страна();
новаяСтрана.Наименование = "Новая Земля";
```

```
// Сохранение объекта в базу данных
```

```
DataServiceProvider.DataService.UpdateObject(новаяСтрана);
```

**Чтение данных:**

```
// Выборка списка объектов по условию
List<Страна> страны =
DataServiceProvider.DataService.Query<Страна>(Страна.Views.СтранаL)
    .Where(s => s.Наименование.Contains("Земля"))
    .ToList();
```

**Настройка запросов:** Flexberry ORM предоставляет широкие возможности для настройки запросов к базе данных, включая использование LINQ-выражений, фильтров, сортировок и проекций (Views).

**Перехват событий:** можно внедрять дополнительную логику перед или после обращения к базе данных, используя события и перехватчики.

## 5. Особенности эксплуатации

- **Поддержка различных СУБД:** Flexberry ORM может работать с различными базами данных (MS SQL Server, PostgreSQL, Oracle и др.), что требует соответствующей настройки провайдера данных.
- **Гибкая настройка отображения:** имена таблиц, столбцов и других элементов БД можно настраивать, что позволяет адаптировать ORM под существующую структуру БД.
- **Поддержка кастомизации:** пользователь может изменять поведение ORM, включая логику формирования SQL-запросов, если стандартного функционала недостаточно.
- **Интеграция с OData:** ORM может использоваться для создания OData-сервисов, что расширяет возможности по интеграции и предоставлению данных внешним системам.

## 6. Ресурсы для получения дополнительной информации

- Примеры использования и демонстрационные приложения доступны в репозиториях на GitHub.
- Для проектирования и генерации кода рекомендуется использовать Flexberry Designer, который упрощает создание и изменение модели данных.

## 7. Контакты для получения технической поддержки

- По вопросам приобретения лицензий и технической помощи обращайтесь по адресу [support@flexberry.net](mailto:support@flexberry.net).