DevOps Dreamin’

Welcome to DevOps Dreamin’

Mastering Salesforce Industries DevOps: A Guide to Omnistudio CLI and IDX Workbench

Sponsored by Gearset

Agenda

- Introduction
- Intro to Omnistudio
- Omnistudio Developer Tools
- Omnistudio Settings
- Enablement Resources
- Q&A

DevOps Dreamin'

Sponsored by Gearset

Luca Belligoli

- 7+ years in Salesforce/Salesforce Industries
- Ex-Salesforce Professional Services
- 14x Certifications, 3x Accredited Professionals
- Part of Media Cloud AP Team
- Dubai, United Arab Emirates
- Linkedin: in/lucabelligoli

Stratus Carta

salesforce PARTNER SINCE 2022

ACCREDITED Communications Cloud

ACCREDITED Media Cloud

ACCREDITED Energy and Utilities

DevOps Dreamin’

Sponsored by Gearset

Brought you by

Stratus Carta

Salesforce Industries Expert Services

Expert Services

Stratus Carta is the go-to hands-on advisor for Salesforce Industries and Revenue Cloud customers globally, tackling the hardest challenges in the ecosystem. Our areas of expertise include DevOps, EPC / PCM / CPQ / OM / DRO / BRE design, proof of concepts, performance tuning, scale testing, best practices reviews, upgrades, data migration, greenfield vs brownfield assessments, and more.

Stratus Carta

salesforce PARTNER SINCE 2022

CERTIFIED Technical Architect

ACCREDITED Communications Cloud

ACCREDITED Media Cloud

ACCREDITED Energy and Utilities

DevOps Dreamin'

Sponsored by Gearset

Intro to Omnistudio

What is Omnistudio?

Omnistudio is a **suite of digital engagement tools** that simplify the creation of complex, industry-specific experiences on Salesforce. It enables architects and developers to create **scalable**, **efficient**, and **user-friendly** applications through a **low code** environment. Omnistudio provides set of **services**, **components** and **predefined data model objects** that combine to create industry specific solutions.

Where can I use Omnistudio?

- Communications
- Media
- Energy & Utilities
- Public Sector
- Net Zero
- Financial Services
- Automotive
- Health
- Manufacturing
- Education
- Revenue Cloud

Omnistudio Components

Omniscripts

Guide users through business processes.

FlexCards

Display contextual information at a glance.

Integration Procedures

Orchestrate multi-step server-side processes.

Data Mappers

Extract, transform, and load (ETL) data.

History: Omnistudio for Vlocity

- Originally part of the **Vlocity ecosystem**
- A set of tools for building **Salesforce Industry Cloud apps**
- Includes:
  - Prebuilt **services**
  - Reusable **UI components**
  - Industry-specific **data models**
- Delivered as a **managed package**
  - For **Healthcare**, **Insurance**, **Communications**, **Media**, **Energy**, and **Public Sector**
- Built using **custom Salesforce objects**

DevOps challenges

Vlocity Deployment: Key Differences

- Vlocity (Omnistudio for Vlocity) uses data-based configuration (JSON) called **DataPacks**, not Salesforce **metadata**
- Cannot deploy using **Metadata API** or **Change Sets**
- Requires specialized tools:
  - **Vlocity Build Tool** (CLI)
  - **IDX Workbench** (UI)

Challenges with Vlocity Deployments

- Separate tools for metadata vs. Vlocity
- Extra steps: Must run **separate deployments** for Vlocity DataPacks
- More tools = more to learn
- Split **audit trail** across tools
- Harder to fully integrate into **DevOps pipelines**

Today: (Standard) Omnistudio

- Since Summer ’21, Salesforce introduced the **Omnistudio**, built on the **standard Salesforce data model** for better compatibility and integration.
- Omnistudio application suite are **Flexcards**, **Omniscripts**, **Data Mappers**, **Integration Procedures**, **Business Rules Engine** and **Document Generation** (with right license).
- Not all Omnistudio for Managed Packages **features** are available in Omnistudio.
- There are two components to this - **Standard Omnistudio Data Model** to store the metadata (e.g. Omniscript definition) into standard objects instead of previously managed packages and **Standard Omnistudio Run-Time** to move the business logic out of the managed package to Salesforce core

DevOps Dreamin'

Sponsored by Gearset

Omnistudio Developer Tools

DevOps Dreamin'

Sponsored by Gearset

IDX Workbench

A **GUI-based desktop application** that simplifies working with **Omnistudio** and **Salesforce Industries components** in Salesforce.

Migration Flow

1. Configure Workspace
    a. **Repository**: A repository saves relevant configurations, project items to be migrated, ..
    b. **Source**: definition of Source environment
    c. **Target**: definition of Target environment
    d. **Project**: container of all components to migrate
2. Create new **Project**
3. Select SF/SFI components to **migrate**
4. Migrate **with/without dependencies**
5. **Validate** migration

IDX Workbench Supported Deployment

IDX Workbench supports migrations between the following sources and targets.

Source	Target
Salesforce org	Salesforce org
Salesforce org	Git repository
Git repository	Salesforce org
Omnistudio Process Library*	Salesforce org
JSON file	Salesforce org
JSON file	Git repository

* Omnistudio Process Library aka Vlocity Process Library (GitHub repo)

DevOps Dreamin'

Sponsored by Gearset

Demo
IDX Workbench

DevOps Dreamin'

Sponsored by Gearset

Omnistudio Build Tool

A Node.js **command-line interface automation** tool that packages and deploys **Salesforce** and **Salesforce Industries components**.

Org to Org Migration Flow

1. Install Build Tool
2. Create Export YAML file
3. Run the Export
4. Export Troubleshooting
5. Check Exported Data
6. Run the Deploy
7. Deploy Troubleshooting

1 - Install Build Tool

1. npm install --global vlocity
2. vlocity help

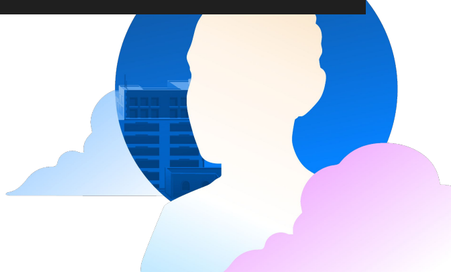
```
[lucabelligoli@Mac ~ % vlocity help                                                ]
  Vlocity Build v1.17.12
  All available commands:
  packExport >> Export all DataPacks
  packExportSingle >> Export a Single DataPack
  packExportAllDefault >> Export All Default DataPacks
  packDeploy >> Deploy all DataPacks
  packDeployMultiple >> Deploy all DataPacks from array of different project paths
  cleanOrgData >> Run Scripts to Clean Data in the Org and Add Global Keys to SObjec
ts missing them
  validateLocalData >> Check for Missing Global Keys in Local Data. Use argument --f
ixLocalGlobalKeys to additionally add missing or change duplicate keys.
  refreshProject >> Refresh the Project's Data to the latest format for this tool
  packRetry >> Continue Running a DataPacks Job Resetting Errors to Ready
  packContinue >> Continue a DataPack Job
  packUpdateSettings >> Update the DataPack Settings in your Org
  packGetDiffs >> Find all Diffs in Org Compared to Local Files
  packGetDiffsAndDeploy >> Find all Diffs then Deploy only the Changed DataPacks
  packGetDiffsCheck >> Find all Diffs in Org Compared to Local Files and fail if any
 found
  packBuildFile >> Build a DataPack File from all DataPacks
  packGetAllAvailableExports >> Get list of all DataPacks that can be exported
  refreshVlocityBase >> Deploy and Activate the Vlocity Authored Cards and Templates
 included in the Managed Package
  installVlocityInitial >> Deploy and Activate all the initial Vlocity DataPacks inc
```

2 - Export YAML file

- Create a Job file which identifies **projectPath**
- Export project file identifies which components to be **extracted**
- Review GitHub repo for **Supported DataPack Types**
- Review **DataPack key** for selected Supported DataPack Types

export.yaml

```yaml
projectPath: ./vlocity
expansionPath: .

manifest:
- OmniScript/Test_Flow_English
- OmniScript/Demo_SubmitOrder_English
- OmniScript/Demo_CreateAccount_Multi-Language

- FlexCard/accountButtons
- FlexCard/quoteProcessCard

- IntegrationProcedure/OS_CreateAccountHierarchy
- IntegrationProcedure/OS_CreateOrder

- DecisionMatrix/BaseDecisionMatrix

- Product2/fa3f4327-d474-6f96-9306-87a6cb148840
- Product2/49c7c39e-e355-b883-4cfe-d741fc11c656

```



3 - Run the Export

```
vlocity -sfdx.username source_org@devOps25.com -job
export.yaml packExport
```

where:

1. `-sfdx.username`: to use a Salesforce DX Authorized Org for authentication.
2. `-job`: specify the export YAML File path
3. `packExport`: to retrieve all Vlocity Metadata from the org as Vlocity DataPacks as defined in the Job File and write them to the local file system in a Version Control friendly format.

4 - Export Troubleshooting

If you encounter errors during export, review them carefully - they may indicate issues that will also occur during deployment.

After fixing the errors, use the `packRetry` command to re-export failed components.

If the export was interrupted (e.g., due to a connection issue), use `packContinue` to resume the process without starting over.

5 - Check Exported Data

After exporting, run a validation check to identify issues like duplicate or missing Global Keys using the `validateLocalData` command.

You can auto-fix these issues with `--fixLocalGlobalKeys`, but it's generally recommended to correct the data directly in the source org for better accuracy and control.

6 - Run the Deploy

```
vlocity -sfdx.username target_org@devOps25.com -job
deploy.yaml packDeploy
```

where:

1. `-sfdx.username`: to use a Salesforce DX Authorized Org for authentication.
2. `-job`: specify the deploy YAML File path
3. `packDeploy`: will deploy all contents in the projectPath of the Job File to the Salesforce Org.

7 - Deploy Troubleshooting

If deployment errors occur, review them carefully - they may have been resolved by earlier uploads of missing data.

Use `packRetry` to retry failed DataPacks, which can often resolve issues caused by deploy order or Salesforce governor limits.

Run `packRetry` multiple times until the error count no longer decreases. For unresolved issues, refer to the troubleshooting guide.

DevOps Dreamin'

Sponsored by Gearset

Demo Omnistudio Build Tool

Best Practices

1. Use DataPack keys in the `manifest` - check documentation for the DataPack key / SObject mapping
2. Always create 2 Job Files: `export.yaml` and `deploy.yaml`
3. To export components, it is recommended to use `manifest`. Alternative approach, `queries`.
4. Use `gitCheck: true` to deploy only DataPacks that have changed since the last commit - improves speed and avoids unnecessary updates.
5. `buildFile` - specifies a file to create from the data pack directory.
6. Use `preJobApex` to run async Apex before the DataPack Job starts.

IDX Workbench vs Omnistudio Build Tool

- It is recommended the Omnistudio Build Tool for deploying Omnistudio components between orgs. It ensures that the deployment includes dependencies, it validates deployment success, and you can use it with automated deployment, such as scripts or Jenkins. (Omnistudio with Metadata API = false)

DevOps Dreamin'

Sponsored by Gearset

Omnistudio Settings

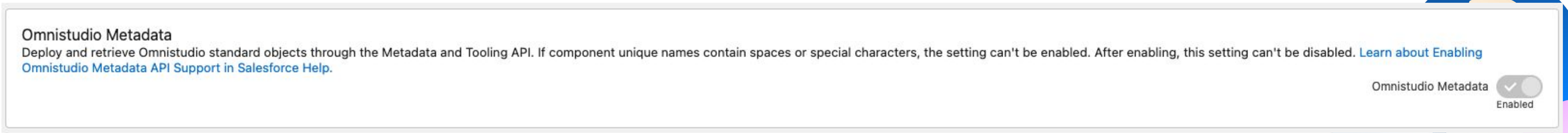
Omnistudio Metadata

Omnistudio Metadata API allows **Omnistudio components** - like OmniScripts, FlexCards, Data Mappers, Integration Procedures, and others - to be treated as Salesforce metadata.

Components can be **deployed, retrieved,** and **version-controlled** using **standard Salesforce** tools like: Metadata API, CI/CD

Benefits:

- **Facilitating the use of Salesforce DevOps tools:** This allows for streamlined deployment and version control of Omnistudio components.
- **Supporting standard deployment mechanisms:** This includes utilizing Metadata API-based deployments.



Omnistudio Naming Convention

When enabling the Omnistudio Metadata API, Salesforce validates that all component names follow required conventions. Names must be **alphanumeric only**, with **no spaces or special characters** (e.g., underscores). If any names are invalid, the process fails.

Product	**Object**	**Unique Name**
Integration Procedures	Omni Process	Type + SubType *Example*: *Auto_CreateUpdateQuote*
Data Mappers	Omni Data Transformation	Interface Name *Example*: *DRGetAccount*
Omniscripts	Omni Process	Type + Subtype + Language *Example*: AccountCreateEnglish
Flexcards	Omni Ui Card	Name *Example*: AccountButtons

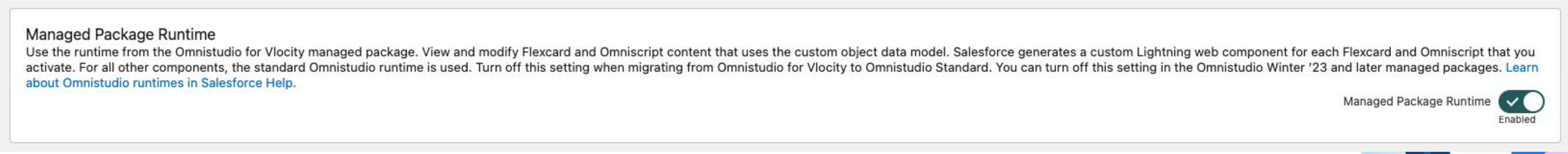
Metadata API enabled

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2021/10/metadata">
   <types>
      <members>*</members>
      <name>OmniScript</name>
    </types>
    <types>
      <members>*</members>
      <name>OmniUICard</name>
   </types>
   <types>
      <members>*</members>
      <name>DecisionMatrixDefinition</name>
   </types>
   <types>
      <members>*</members>
      <name>ExpressionSetDefinition</name>
   </types>
   <version>62.0</version>
</Package>
```

Omnistudio Runtime

The **Managed Runtime** executes Omnistudio components outside the Salesforce core, using a Salesforce-managed AWS infrastructure. It provides enhanced performance, scalability, and control, making it ideal for external-facing apps, high traffic volumes, and real-time experiences.

The **Standard Runtime** runs Omnistudio components (like OmniScripts and FlexCards) within the Salesforce Lightning Platform. It leverages the org's native resources and is ideal for internal users and low-to-moderate traffic use cases.



DevOps Dreamin'

Sponsored by Gearset

Enablement Resources

Let's Unwrap the Goodies! 🎉

- Omnistudio Build Tool documentation
- To learn more about the setup visit Salesforce Help documentation
- On Trailhead, start with Build Guided Experiences with Omnistudio trail and Omnistudio Developer Tools module
- Our Delhi Dreamin' 25 Omnistudio architecture session
- To access hands-on videos, check out the Omnistudio Apex Hours series:
  - Getting Started with Omnistudio

Q&A

DevOps Dreamin'

Thank you!

Contact details:

Email: *luca@stratuscarta.com*
Linkedin: *in/lucabelligoli*

Sponsored by Gearset