**OmniStudio DataRaptors**

**Tuning for Performance**

As Cards, VIPs and OmniScripts depend on DataRaptors, if the DataRaptors do not perform

neither will the Cards or OmniScripts

**Key issues we have seen at customer projects**

* Teams are trying to create “Uber” DataRaptors to cover as many of their data needs as possible. (as high as 12 Objects)
* Teams are adding objects to the DataRaptors to get data that could be accomplished by using relationship queries

**Recommendations**

* Create targeted DataRaptors to only get the data that is needed for any one operation
* Use relationship queries whenever possible to pull data from other objects.
* Try to keep the number of objects to 3 or less
* Ensure that all filters and sorts have supporting indexes

**DataRaptor Turbo Extract**

Retrieves and filters data from a single Salesforce object type with support for fields from related

objects.

**Doesn’t support:**

* Formulas
* Complex output mappings.

**Advantages over a standard DataRaptor Extract.**

* Simpler configuration
* Better performance at runtime

**OmniStudio Integration Procedure**

**Best Practices**

* Allocate space to ViocityMetadata and ViocityResponse Cache Partitions
* Use Integration Procedures for data calls
* Send Only necessary Data
* Use Response Action to return only necessary data
* Use DataRaptor Turbo Extract
* Use Chain on Step only if it’s absolutely necessary
* Check necessary indexes for CPQ Objects
* Switch off Pricing on CPQ API calls when not used.
* Limit where possible the CPQ Pricing calls
* Use Remote Action only when necessary
* Use Cache Block and Cache features where possible
* Where possible, use Integration Procedure for SOAP and Rest Callouts
* Where possible, expose Integration Procedure as Rest API
* Use Named Credential
* When managing security to run Integration Procedure and DataRaptor, use custom permission
* since Summer ‘19 \*\*
* Secure cached data
* Avoid unnecessary SetValues
* Define and apply a common naming convention
* Use conditional Block to group actions

**OmniScripts Best Practices**

* Server Side
  + Trim JSON requests
  + Use Integration Procedures to reduce server roundtrips
  + Perform Calls Asynchronously where possible
  + Distribute Actions across OmniScript
  + Use Cache for DataRaptor and IPs
  + Use Action Block where possible
  + Avoid Long Running Transaction where possible
  + Avoid using chain on step on SetValues element
* Client Side
  + Reduce Conditional Views, Merge Fields,
  + Formulas where possible
  + Trim JSON responses
  + Make sure OmniOut application and
  + JSON definitions are loaded from CMS.
  + Remove spaces from all UI element names
  + Reduce number of OmniScript elements.
  + Run logic on the Server where possible
  + Time Tracking Flag