AJA:

Have you guys noticed anything working incorrectly

Would you like me to record this session?

**Update sets**

Is there anything you want me to cover specifically? I’ll throw that out there so you can think about it

But I’ll leave time so we can request clones and pull update sets from dev

**Let’s go over basic configuration to help distinguish between your dev and prod instances:**

\*What I’m about to cover are one time set ups\*

—> type in basic configuration

* click into basic configurations UI16
* you can change the ui here to distinguish dev from prod
* then click the gear icon, hit the developer tab, show update set picker in the header
* we’ll talk more about that in a little bit

**Go to users to set up a new update set user:**

list of users

type user in filter navigator

* type in user
* administration —> users —> new
* user ID: updateset.admin
* First name: Update Set
* Last name: admin
* we’ll give it a password
* then give it **admin** role

**Now jump into prod instance to set up new update source**

* type in update
* update sources
* this is just one time set up — if you ever get a test instance, you would need to do this again

**New**

* Name oxford Dev
* type dev
* username updateset.admin
* password
* and URL of dev instance
* then test connection

**In prod:**

* go to —> clone targets
* new
* url - dev
* updateset.admin
* password
* then go to —> request clone
* exclude audit and log data TRUE
* exclude large data TRUE - i recommend you deselect this the first time, if you find it’s taking a long time this time, you can keep it checked next time, but it would be valuable to have this data
* preserve theme TRUE
* email upon completion - your email
* date and time is autoselected for you — make it 4.5 hours ahead

**we always clone down from prod to dev**

**next, we’ll look at update sets - *what are update sets*?**

* do you have any experience with GitHub or a system where you’re making changes and saving them
* containing those changes in a container
* and then when we’re ready, we move those changes over from dev to prod
* does that concept sound familiar to you at all?

type in update in filter nav

* we’re going to look at local update sets
* click into the **default** update set
* every SN instance will always have an update set
* never delete it
* the surface that we’re

update set - not a 1:1 alignment, instead it looks and says what type of sequel call is this, insert or update, it looks to see if there is already a record

* click into a record int he default update set form - what we have here in the XML payload, not very human readable
* every form we use in SN has an XML
* if you click the hamburger icon
* and click show XML, we can view it for this record

**create new update set**

* **name**: it can have spaces, make is human readable, 24 character limit
* you want to establish a protocol for everyone to use
* start with a date, i go backwards
  + then a dot or dash then my initials
  + then we could do dot test for today
  + 20180530.LH.test
* then a **description** - detailed account of whats in this update set, what changes you have made
* then if there are any special instruction for applying in prod
  + “don’t apply until wed, or needs to be applied after wed”

it’s really important we keep track of what update set we’re in

**2 important questions to ask yourself as an admin:**

1. which instance am i in
2. which update set am i in - if there’s ever any doubt if you should be in one or not, then just make one - much easier than accidentally working in your default update set or in the wrong set

if you say complete, as “state” you’re saying it’s ready to move to prod - so never mark it as complete if you’re not ready to move it

**now go to PROD instance**

* update sources
* let’s bookmark this page — you’ll be going here often — create favorite
* oxford dev —> retrieved update set (related tab)
* show all preview record
* if you ever need to back up an update set, you can just export the XML
* you can import XML from any table in SN
  + —> right click int he table column header
  + —> has to be a list view, not a form
  + —> click in line with the column header

one thing that’s helpful to know is you do have upgrades in enterprise, two releases per year, you can schedule these yourself, you don’t want to get more than 2 version behind, most recent release is kingston

it is a process similar to what we’ve gone thru here, get all your instances, performa c lone, make them all identical, then you would do the upgrade one at a time, do the dev instance first

* because we can script in enterprise, there might be issues that we haven’t anticipated with the upgrades and things not being compatible
* whenever you do a modification to something OOB in enterprise, you want to do insert and stay instead of modifying OOB, never modify the OOB — that way if there is ever problems with an upgrade you can refer back to that OOB version
* process to kick off an upgrade — after you’ve done the prep work, you would create a high ticket
* Patches are applied automatically in most cases
* i would request the upgrade for you dev instance so dev and production are at the same patch (one was patch 7 one was patch 8 )

**be very careful with whom you give admin role to**

* good to establish a system for yourselves, ex: every friday at noon, someone will pull them in, that’s what they do - if you want your work to go in, make sure it’s complete by friday at noon

**Are you familiar with the concept of Plugins?**

* type plugins into filter navigator
* be aware: some are free, some have a fee associated
* with your package, it’s possible some are part of your package, so whenever you’re in doubt, check with your sales rep, they will have the best information for you and most up to date
* Let’s type in change in the “name”

**there are a whole bunch of plug ins related to change**

* collision detector - active - will help you see when a scheduled change is in conflict with a CI elsewhere
* risk assessment - inactive - comes with OOB questions or you can customize it
* CAB workbench — do you guys have a CAB? could be something to experiment with — brings it all into one space — pulls in all the changes
* change management workflow - also inactive,
* **if you’re looking to implement some functionality, it’s definitely worth your time and money to see if there’s already a plugin that does that for you**
* if you don’t load demo data when you activate it, you can always come back and upload it
* **one thing to note, once you activate a plugin, you can’t deactivate it — so you’re in a good place**
* \*\*\* if you activate in Dev, can you clone back to original and then essentially deactivate it

**Tell me more about your experience with scripting, javascript**

* there are APIs unique to SN
* —> client script in filter nav
* do you have your own PDI?? good thing to have for your own learning, you could activate a plugin there, etc
* search for “sample:warn on priority one” in “name” of client script  —> runs on incident table, on submit, running on browser when the form is submitted, so on update, save —> it is grabbing value of priority field, checking if it’s equal to priority 1 —> all of the variable have a number value
* here you could type g\_form
* type field at top — this is onSubmit, they are pretty much what they sound like
  + describe onCell Edit
  + OnChange
  + onLoad
  + onSubmit
* Field name - specify the field name that the changes are being made on

**UI policies**

* —> show close notes
* now we can script
* we can check of Inherit and have it extend any child table
* if you put a UI policy on task — then all the tables that extend from it, incident, problem change will all have the same UI policy

**let’s jump over to server side scripting**

* —> incident form
* right click in a form, configure —> biz rules
* click into a biz rule
* walk thru —> you can click the advanced and script
* when to run — so it’s only running when necessary
* if you don’t need to script, don’t script it — best practices — there’s a lot you can do without scripting — we can now do biz rules at delete and query ni adding to insert and update
* give some examples of **before** biz rules — before the data is submitted to the data base - validation is a perfect example of this
* the **after biz rule** - after the data has been submitted to the data base
* **async** - put stuff in a queue - when the system has some idle time, run this - if there is a choice between after and async, we would always want to choose async
* **display** — ...

let’s go to script includes:

* —> script includes
* type “incident” in “name”
* click into one —> if you scroll down, you can see the function we run in the script
* i’ll get that recording to you and follow up on your questions —> if you run into anything unexpected, feel free to reach out through that ticket, even if I can resolve it for you, i can get you what you need
* CONGRATS and WELCOME to ENTERPRISE!!!