knowledge14®

April 27–May 1, 2014 • Moscone Center • San Francisco, CA

Update Sets: Best Practices to Move Code Smoothly

Matt Gaide
Expert Services Consultant
ServiceNow

Chris Terzian
Sr. Professional Services Architect
ServiceNow

servicenow®

© 2014 ServiceNow All Rights Reserved

Bio



- **Matt Gaide**
- 3 years with ServiceNow
- Based in San Diego, California
- Infrastructure and Operations background with a ServiceNow customer previously



- **Chris Terzian**
- 3 years with ServiceNow
- 7th Knowledge Event
- Based in San Jose, California
- IT Ops and Systems Design background previously with a ServiceNow early-adopter customer.

© 2014 ServiceNow All Rights Reserved

Agenda

What are Update Sets?

XML in Relation to Update Sets

Merging Update Sets – When, Why, and How

Collisions – No insurance needed

Related functionality – Present and Future

© 2014 ServiceNow All Rights Reserved

What are Update Sets?

An update set is a logical grouping of customizations that can be moved from one ServiceNow instance to another:

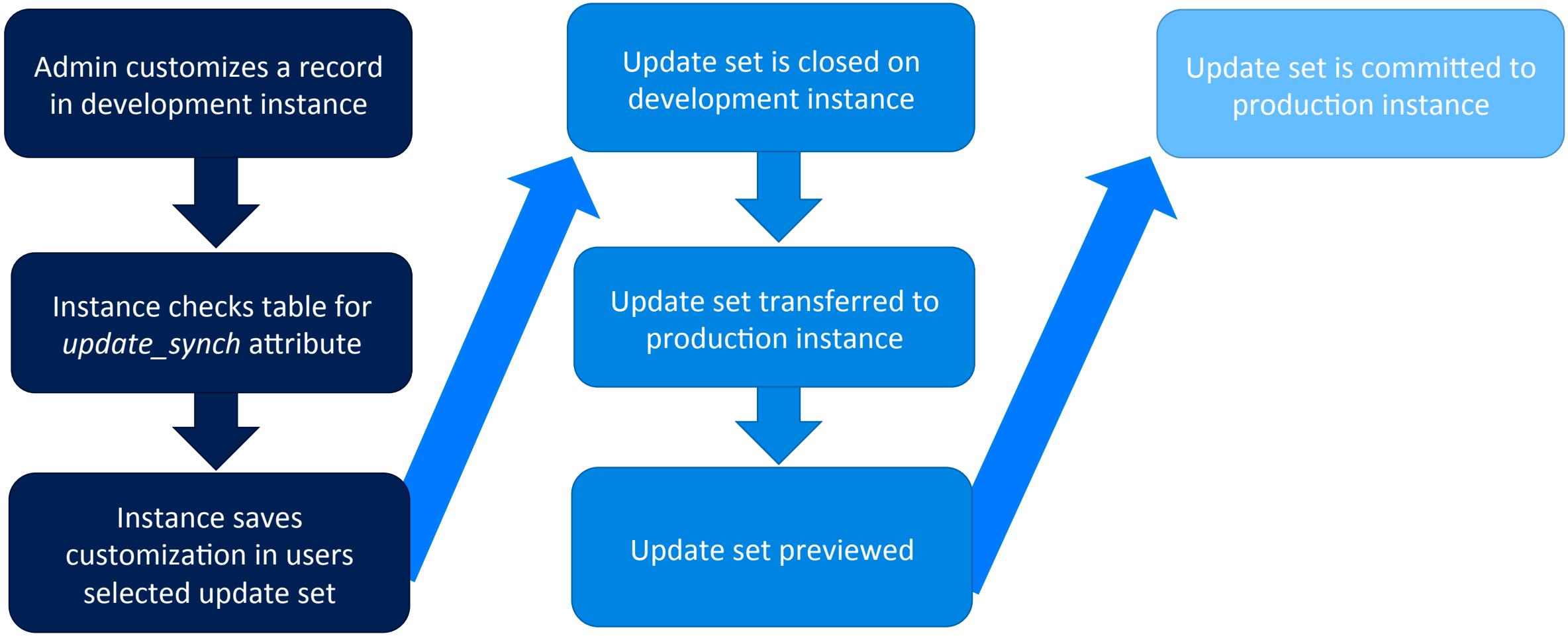
- Mechanism for building and deploying customizations.
- Single unit of code.

What they're NOT:

- They are not a method of 'publishing' apps.
- They are different than import sets.

© 2014 ServiceNow All Rights Reserved

The life of a customization

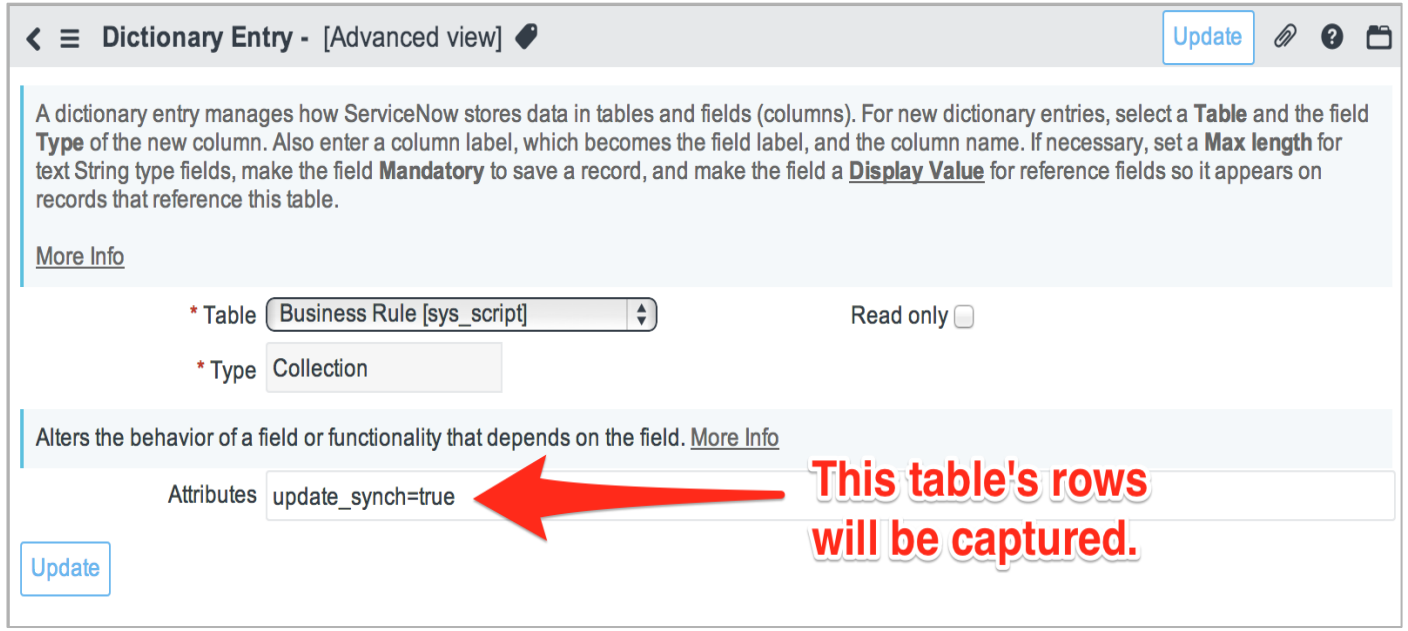


© 2014 ServiceNow All Rights Reserved

Customizing your instance

- The basics.
  - Collection of ‘point in time’ snapshot records.
  - **In Progress** Update Sets capture current changes.
  - **Complete** Update Sets are closed and ready to be transferred to another instance.
  - Records technical changes such as:
    - Table, form, field
    - Business rule, client script, UI policy
    - View, label, section

    Note: Row-level data is not captured!



© 2014 ServiceNow All Rights Reserved

What types of records are NOT captured?

- Attachments
- Schedules
- Users
- Groups
- Incidents, Problems, Changes, CI's (row-level data)

We have a nice trick to show you to deal with cases where you need these …

© 2014 ServiceNow All Rights Reserved

Manually Add Data to an Update Set

```
// Insert an Incident into your current update set
// Introduced in Calgary
// Run this via “Background Script” (requires elevated privs)

var gr = new GlideRecord("incident");
gr.get("9d385017c611228701d22104cc95c371");

var update = new GlideUpdateManager2();
update.saveRecord(gr);
```

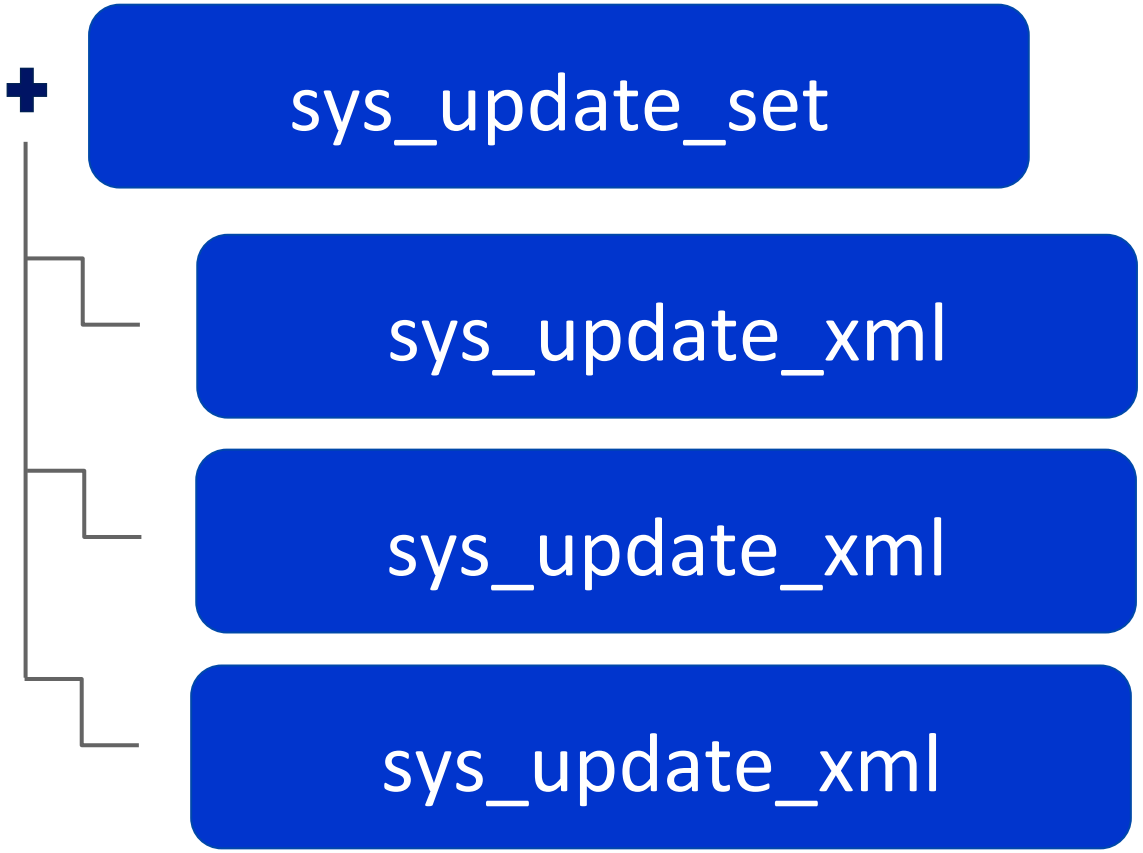
© 2014 ServiceNow All Rights Reserved

XML in Relation to Update Sets

- Exporting to XML is a handy utility.
- But – having multiple versions of the update set in different .xml files might be dangerous
- How do you know which version you are about to apply?

© 2014 ServiceNow All Rights Reserved

XML in Relation to Update Sets



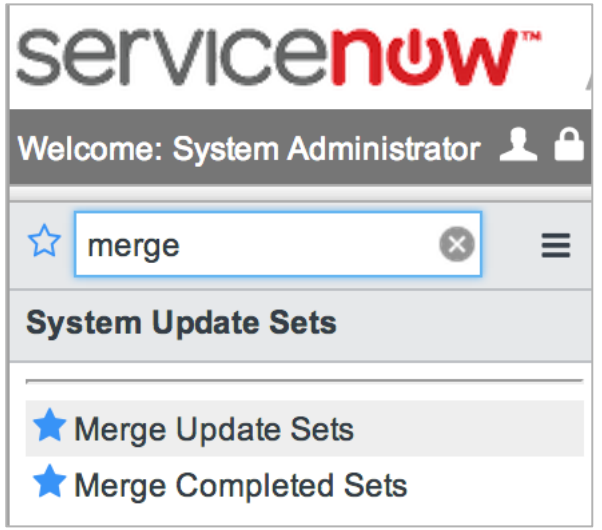
XML Payload

```
<?xml version="1.0" encoding="UTF-8”>
<record_update table="sys_db_object">
<sys_db_object action="INSERT_OR_UPDATE">
<create_access_controls>false<create_access_controls>
<is_extendable>false</is_extendable>
<label>AIX Server</label> … etc
```

© 2014 ServiceNow All Rights Reserved

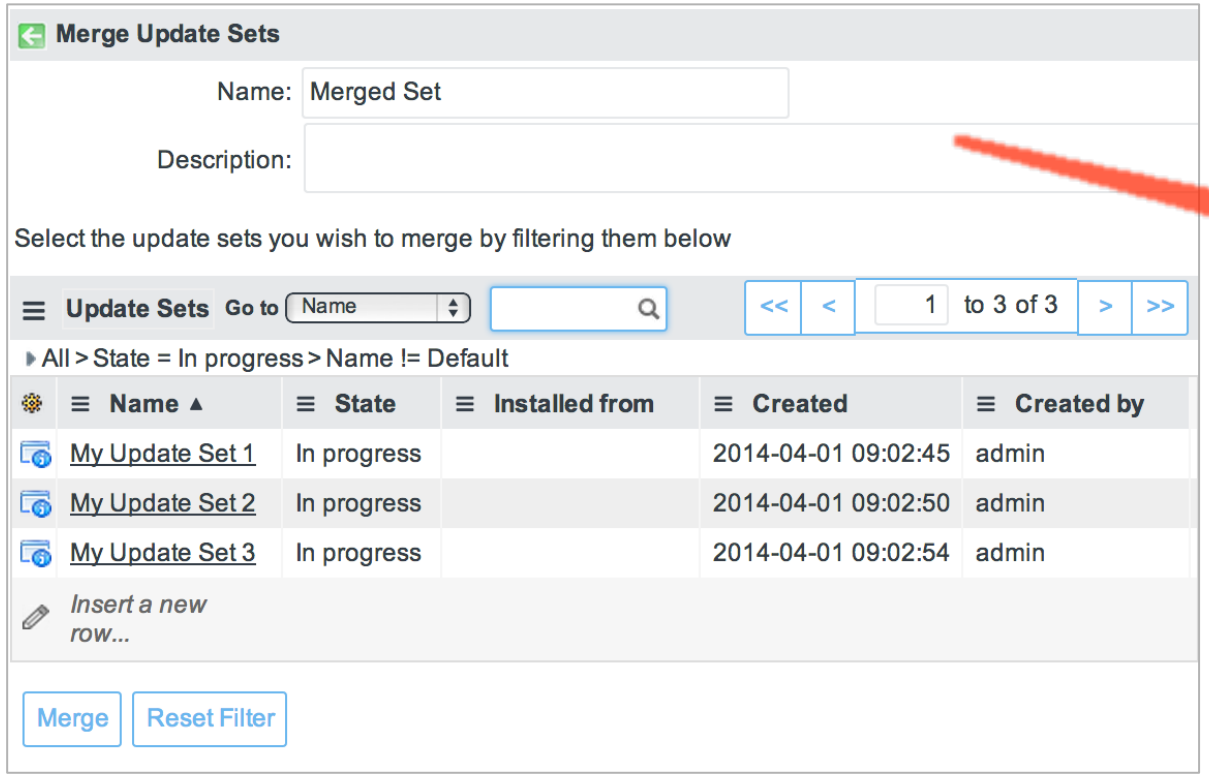
Merging Update Sets – When, Why, and How

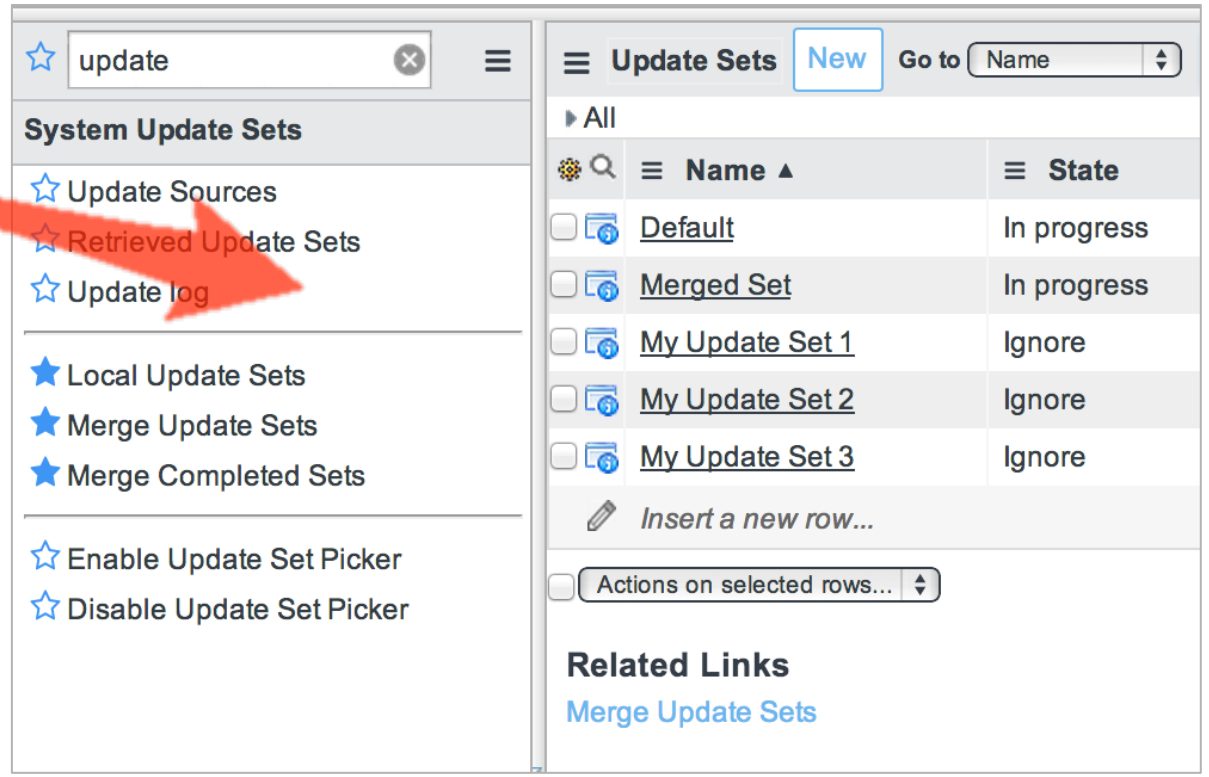
- When?
  - Prior to moving multiple Update Sets from one instance to another.
- Why?
  - Simplifies migration.
  - Reduces the amount of preview and commits.
  - Reduces Update Set conflicts.
  - No more worrying about applying Update Sets in a particular order!
- How?
  - Use Merge Update Sets or Merge Completed Sets in the left Nav
  - Note: Only Merge Completed will auto-set merged sets to 'Ignore'



© 2014 ServiceNow All Rights Reserved

Merging Update Sets – When, Why, and How





© 2014 ServiceNow All Rights Reserved

Collisions – No Insurance Needed

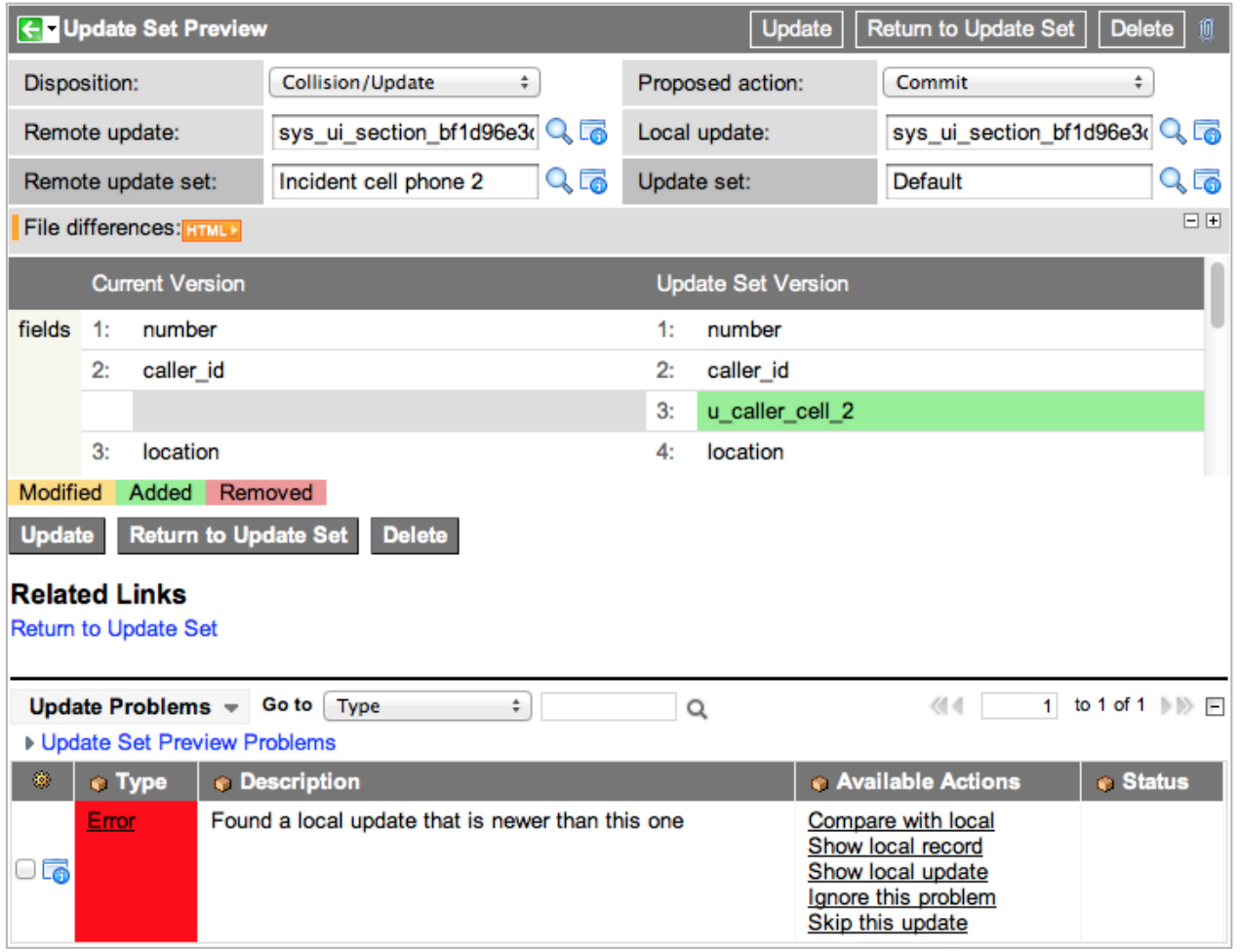
Previewing an Update Set detects problems that may occur if you commit the updates on the local instance. You must address all problems before you can commit the update set.

1. Preview the update set.
2. Open the update set record and scroll down to the **Update Set Preview Problems** related list.
3. Review a problem description to determine the cause.
4. Resolve the problem or choose one of the following options:
   - **Accept remote update:** commit the remote update set without fixing the problem.
   - **Skip remote update:** skip the update when you commit the update set.
5. Repeat steps 2 – 4 for each problem.

© 2014 ServiceNow All Rights Reserved

Collisions – No Insurance Needed

- **Missing object:** the object or a referenced object does not exist on the target instance.
- **Type mismatch:** the type of a column on the target instance does not match the type in the update set.
- **Collision:** a change in the update set is *older* than a change to the same object on the local instance.
- **Uncommitted update:** the object exists in another remote update set that has not been committed.



© 2014 ServiceNow All Rights Reserved

Collisions – No Insurance Needed

1. Preview the update set.
2. Open the update set record and scroll down to the **Update Set Preview Problems** related list.
3. Review a problem description to determine the cause.
4. Resolve the problem or choose one of the following options:
   - **Accept remote update:** commit the remote update set without fixing the problem.
   - **Skip remote update:** skip the update when you commit the update set.
5. Repeat steps 2 – 4 for each problem.

© 2014 ServiceNow All Rights Reserved

Related Functionality – Present and Future

- Application Creator
  - Introduced in the Calgary release
- Team Development
  - Introduced in the Dublin release

© 2014 ServiceNow All Rights Reserved

Application Builder

- New in Calgary.
- Provides a definition of an application… all configuration, data structures and logic in a bundled record.
- Uses Update Sets to transfer applications between instances.
- Can be used to capture past functionality to port to another instance!

© 2014 ServiceNow All Rights Reserved

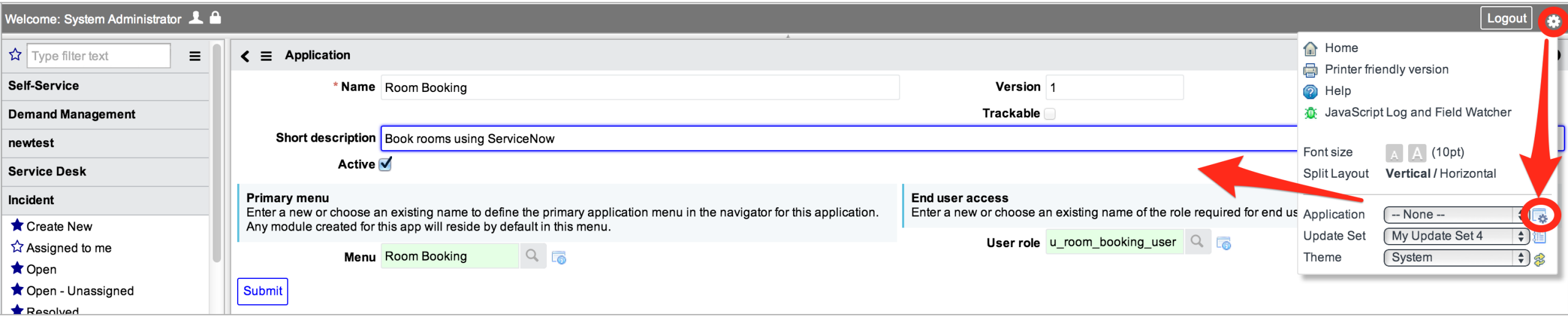
Creating an Application

1) Define the application…

Note:

-‘Menu’ and ‘User’ roles are auto-populated based on the App Name.

-‘Version’ is used in the Update Set name when publishing the App. Useful to identify different App versions.

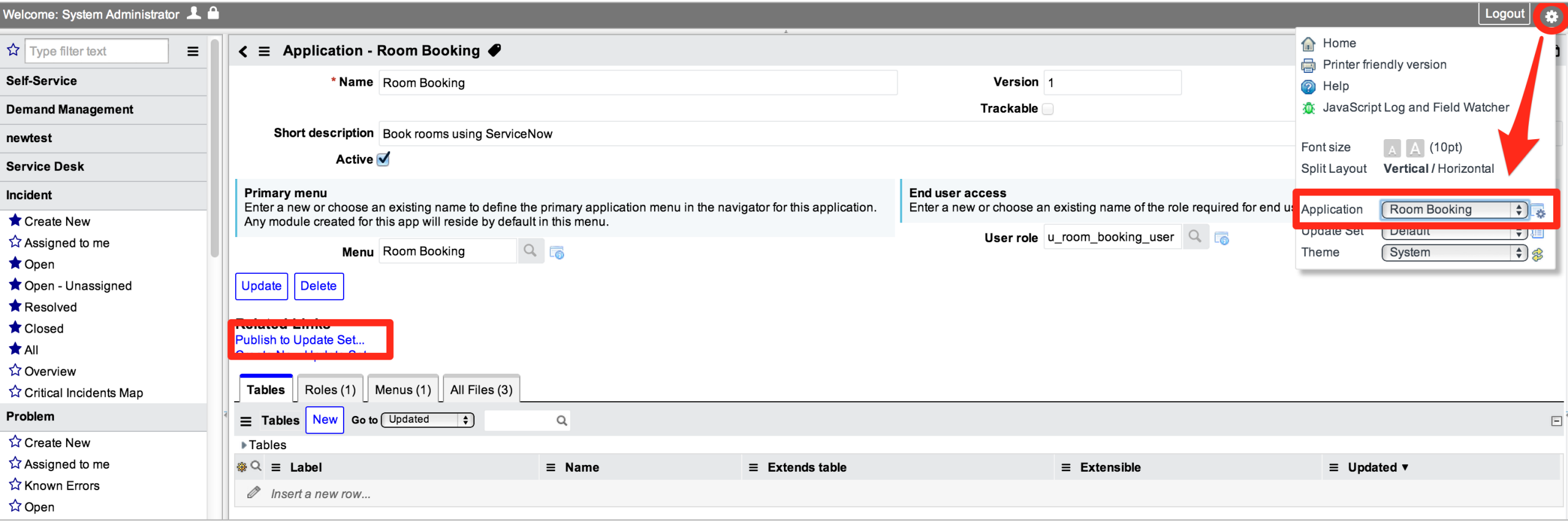


© 2014 ServiceNow All Rights Reserved

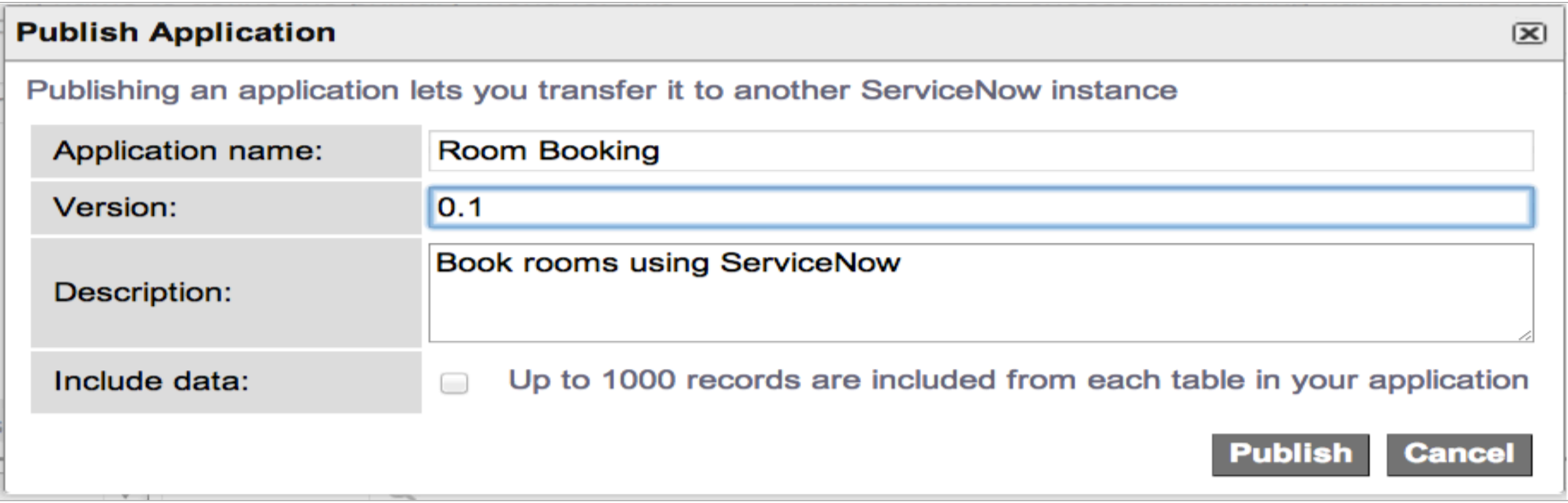
Creating an Application

2) Select your application to add functionality to it's definition file.
3) Publish to a new Update Set (Update Set name based on App Name + Version)

Note: Create an Update Set from the related links has no bearing on the application



© 2014 ServiceNow All Rights Reserved



© 2014 ServiceNow All Rights Reserved

Capture past functionality

Potential use-cases:

- Built something as a proof-of-concept or during a demo and didn't capture everything in an update set.
- Continuing development on delayed or postponed work.
- Experiencing issues with missing components after promoting an update set or merged set.

© 2014 ServiceNow All Rights Reserved

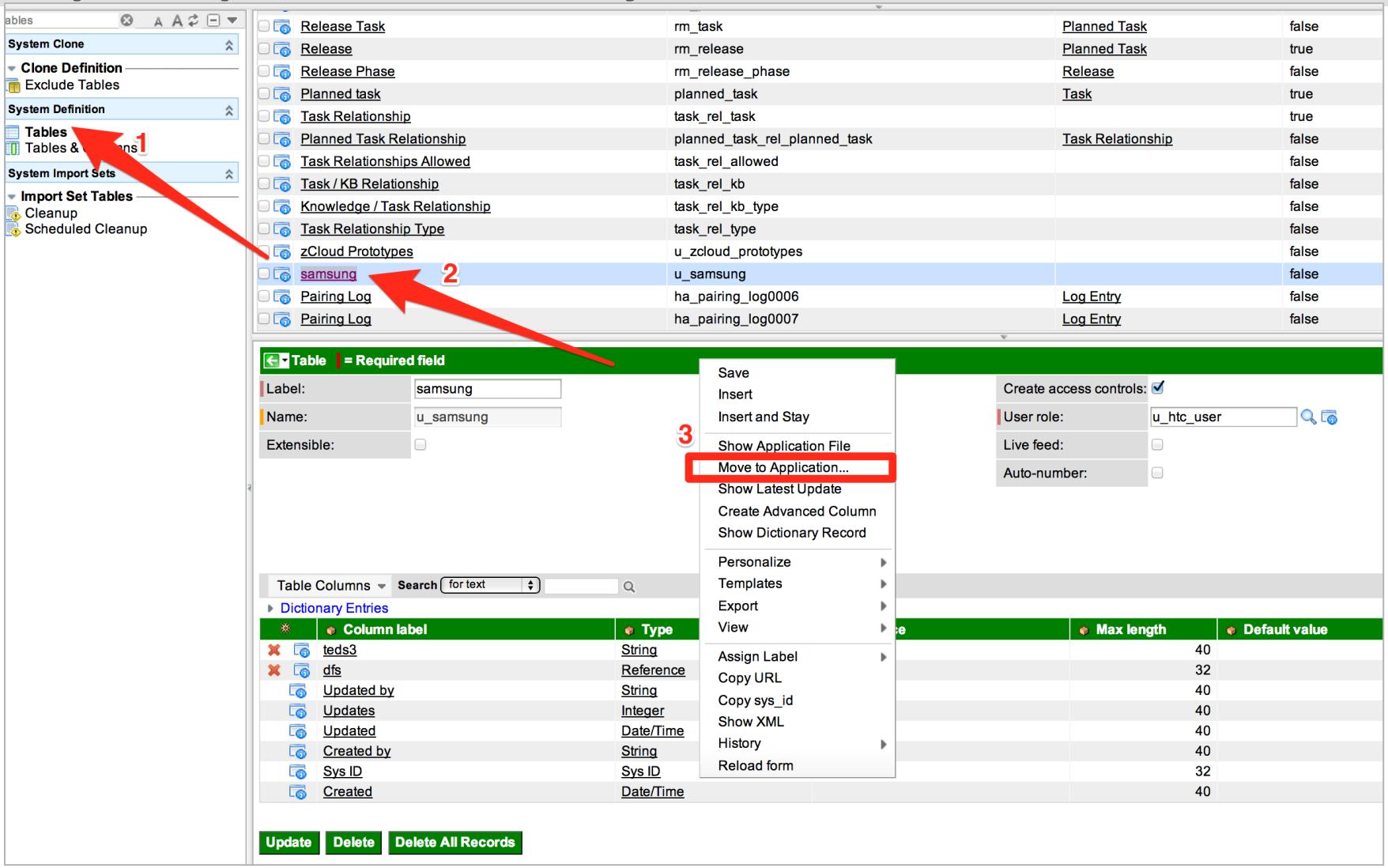
Related Functionality – Present and Future

- Application Creator
  - Introduced in the Calgary release

- Team Development
  - Introduced in the Dublin release

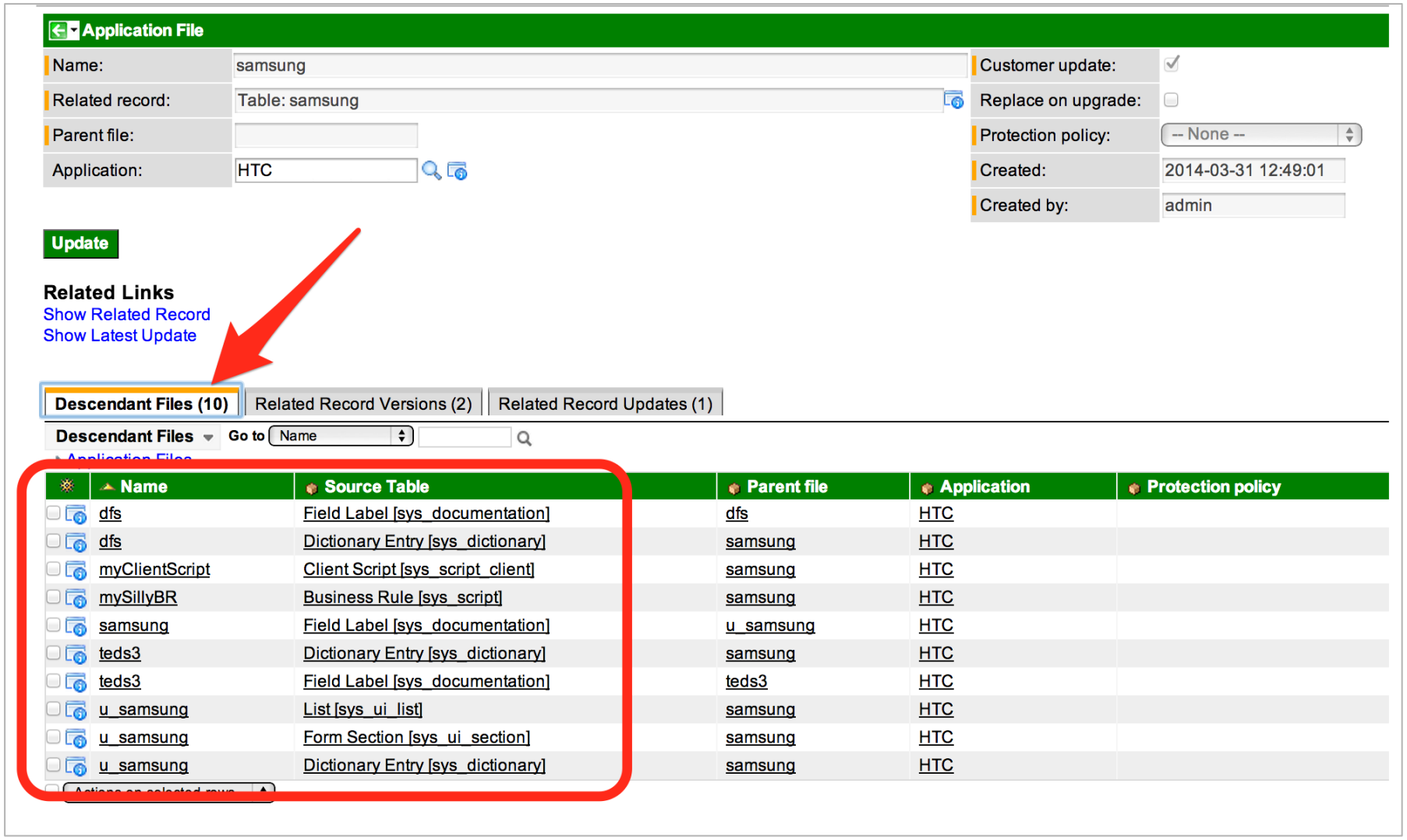
© 2014 ServiceNow All Rights Reserved

Capture past functionality



© 2014 ServiceNow All Rights Reserved

Capture past functionality



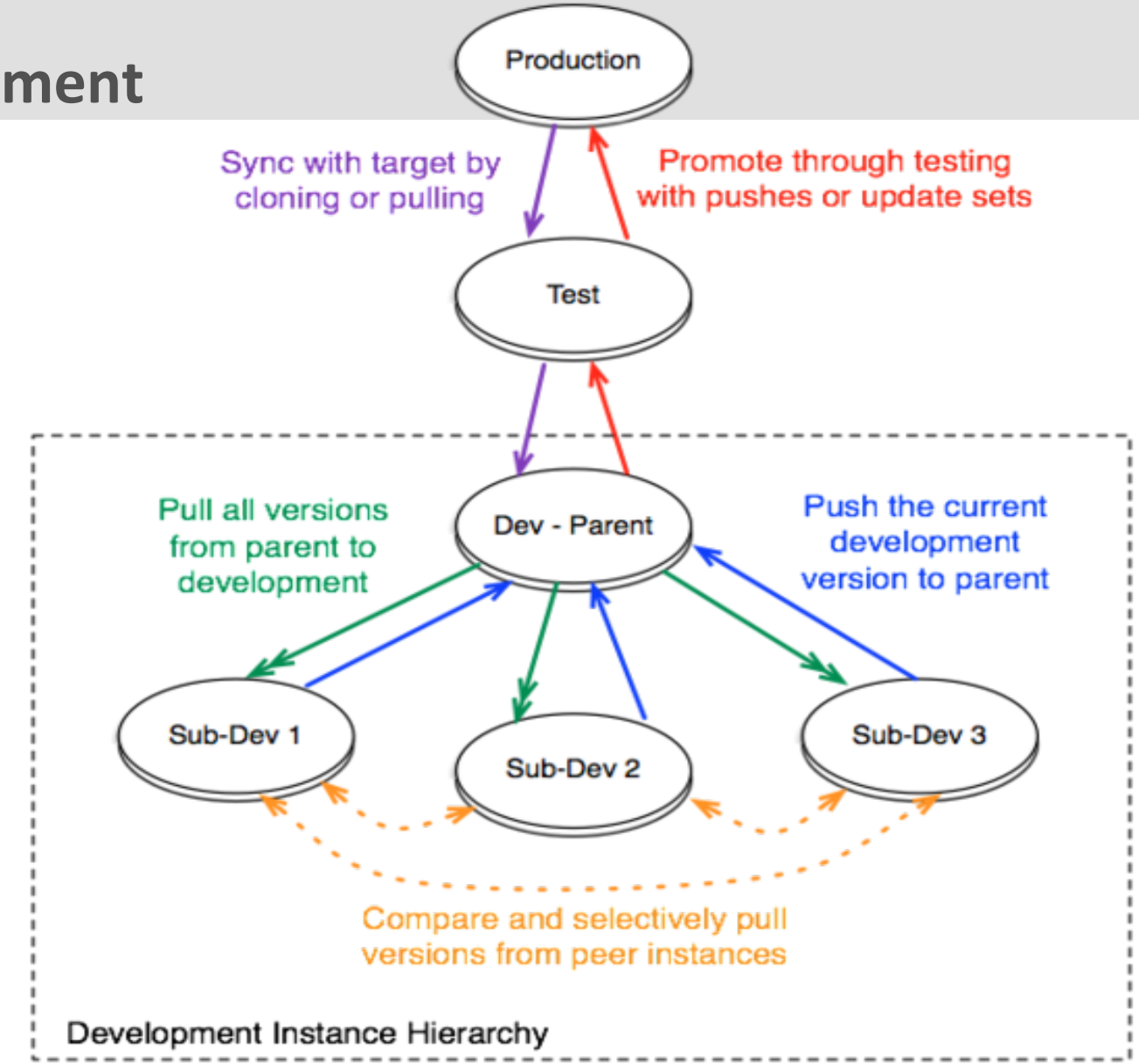
© 2014 ServiceNow All Rights Reserved

Team Development

- New in Dublin – active by default.
- Branching operations, including pushing and pulling record versions between instances.
- Ability to compare a development instance to other development instances.
- Provides a central dashboard for all team development activities.
- Use when you have multiple SN admins building separate functionality across the platform.
- Uses Update Sets to move functionality between instances.

© 2014 ServiceNow All Rights Reserved

Team Development



© 2014 ServiceNow All Rights Reserved

Common mistakes

- Changing one value dumps the entire choice list into the update set
- **task.state** is shared by multiple processes
- Easy to mangle **sys_choice** if you aren't aware of its behavior
- Do you customize choice lists directly on your production instance? ☺

© 2014 ServiceNow All Rights Reserved

sys_choice fields

- Changing one value dumps the entire choice list into the update set
- **task.state** is shared by multiple processes
- Easy to mangle **sys_choice** if you aren't aware of its behavior
- Do you customize choice lists directly on your production instance? ☺

© 2014 ServiceNow All Rights Reserved

Update source behavior

- Transferring update sets using **Update Source** transfers ALL completed update sets.
- Without a strict SDLC process this might not be your intention
- Consider exporting to XML and importing

© 2014 ServiceNow All Rights Reserved

Views and forms rely on each other

- One customization can create multiple update records
- Moving a form without the form section isn't going to work properly
- Same with database fields and dictionary records

© 2014 ServiceNow All Rights Reserved

Update Sets – Our recommendations

- Obsess about Update Set hygiene.
- Always preview before commit.
- Implement a release process which is adhered to and tests the update sets in the migration between instances.
- Use Comments field to document what the update set is aimed at.
- Keep Update Sets small and specific, i.e. per Story, Incident, etc

© 2014 ServiceNow All Rights Reserved

Don’t abuse the Default update set

- Customers sometimes prototype and develop in the Default Update set…
- …And then have to attempt to clean up by moving update records around.
- Be aware of which update set you are currently working in

© 2014 ServiceNow All Rights Reserved

Top Takeaways

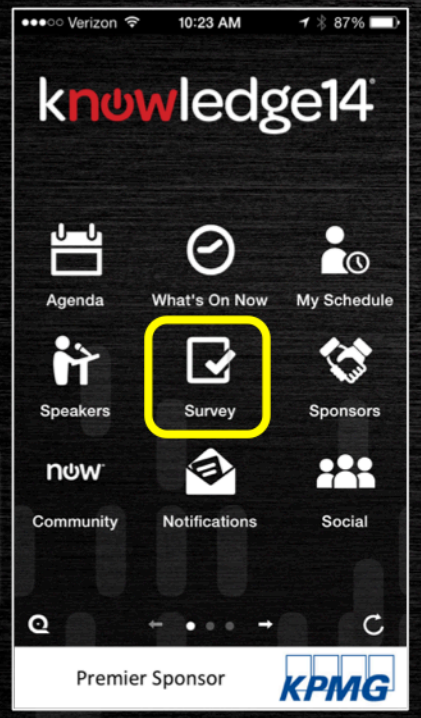
1. Have a strict **DEV -> UAT -> PROD** regime

2. Promote update sets through all instances

3. Don’t let production be the first time you’ve implemented your customizations!

© 2014 ServiceNow All Rights Reserved

Tell us what you think!



Click on the survey link in your Knowledge14 mobile app. Help us make Knowledge 15 even more valuable!

Thank You

Matt Gaide

Expert Services Consultant

ServiceNow

Matt.Gaide@ServiceNow.com

Chris Terzian

Sr. Professional Services Architect

ServiceNow

Chris.Terzian@ServiceNow.com

Want a copy of this deck?



Sign into the BuzzHub to unlock all the slides and lab guides from Knowledge14 sessions

