knowledge14®

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

Technical Best Practices 201: Avoid Pitfalls

Chuck Tomasi
Sr. Services Enablement Program Manager
ServiceNow

Dave Slusher
Sr. Technical Consultant
ServiceNow

servicenow®

© 2014 ServiceNow All Rights Reserved

Agenda

Business Rules Best Practices

Client Script Best Practices

Coding Best Practices



© 2014 ServiceNow All Rights Reserved

Chuck Tomasi



Sr. Services Enablement Program Manager

- 30 years IT experience
- ServiceNow customer 2008-10
- First Innovation of the Year Award @ K10
- ServiceNow since 2010
- Lots of deployments, custom apps, and special projects
- Technical Best Practices owner
- Co-host of ServiceNow web series *TechNow*
- Other stuff:
  - Author, Podcaster, golfer, downhill skier, photographer, sci-fi geek, martial artist

© 2014 ServiceNow All Rights Reserved

Dave Slusher



Sr. Technical Consultant, Expert Services

- 20 years of software development
- Developed core infrastructure for Dell Secureworks, Intel factory automation, GuardedNet
- Developed UI/Web Apps for Orbitz, Intertrust, WebMD
- ServiceNow since 2013
- Other stuff
  - 4th podcaster in the world, radio host and producer, author, co-founder of CREATE South conference

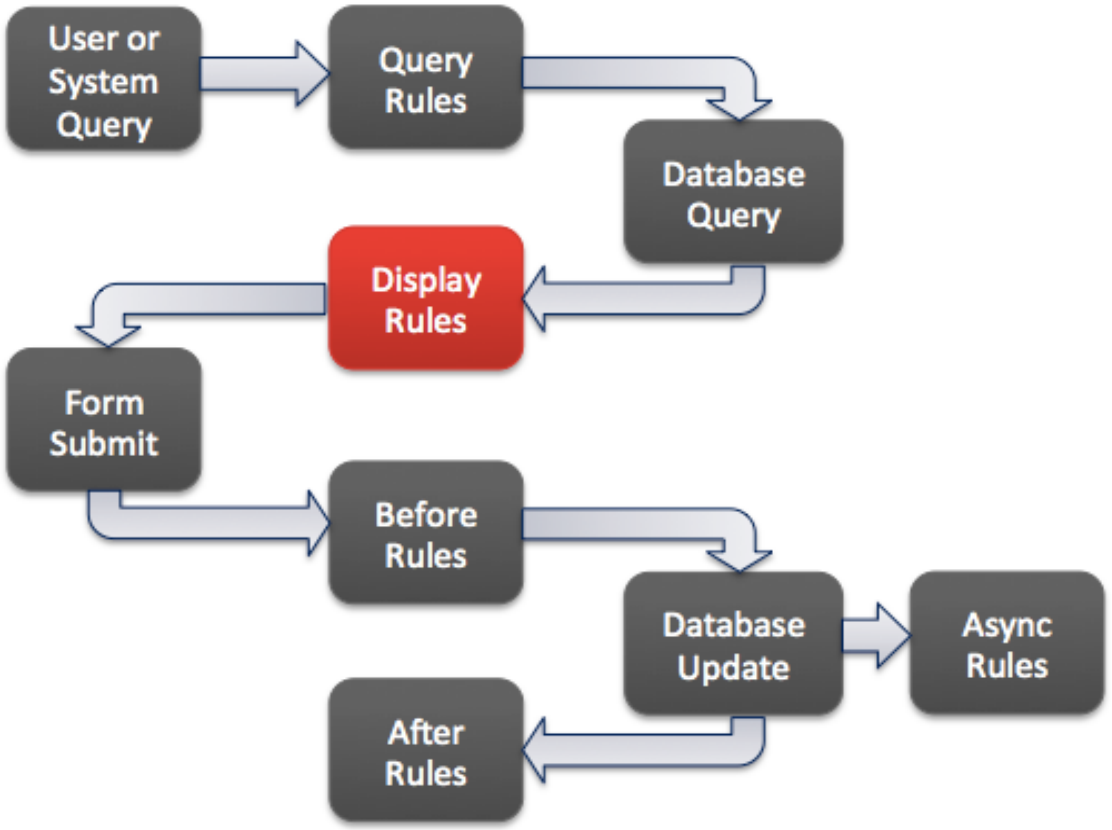
© 2014 ServiceNow All Rights Reserved

Business Rule Best Practices

© 2014 ServiceNow All Rights Reserved

Business Rules

- Display
  - Run just before the form is loaded
  - Typical use: passing g_scratchpad information to client scripts

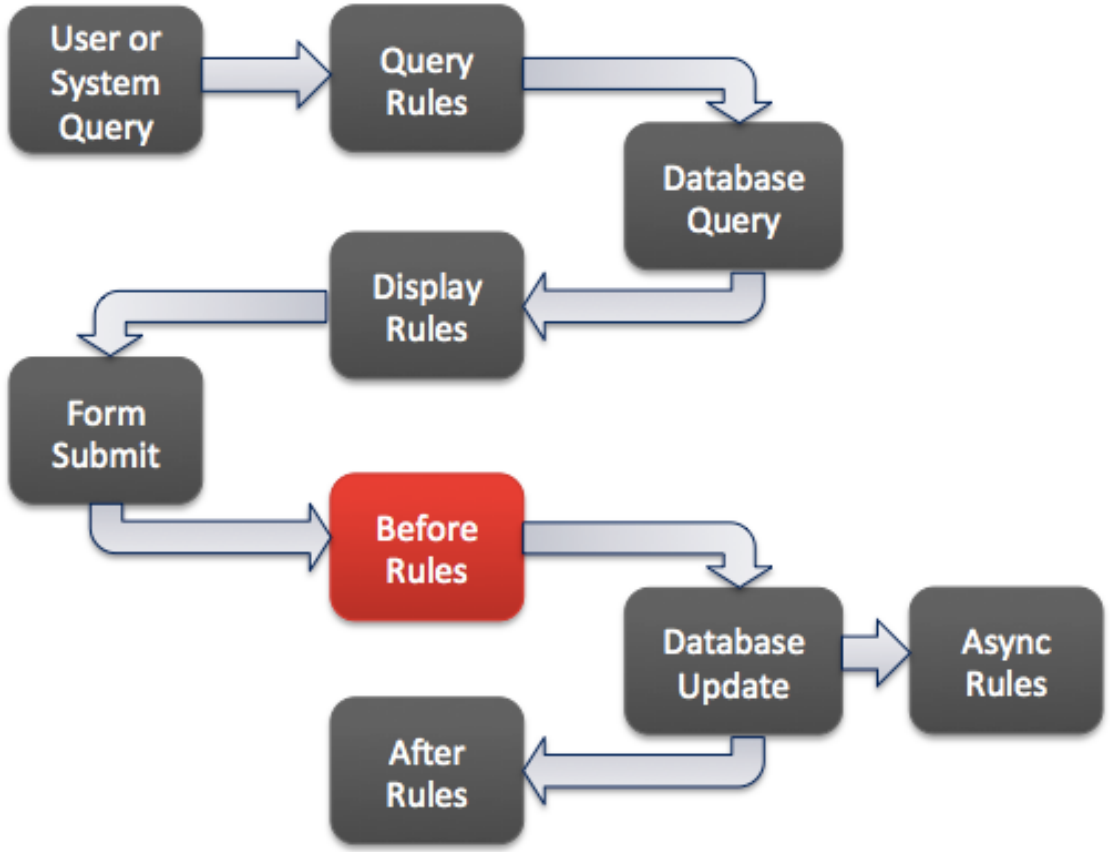


© 2014 ServiceNow All Rights Reserved

Business Rules

- Before
  - Implicit update – **don't use current.update()**
  - Generally used for manipulating the current record



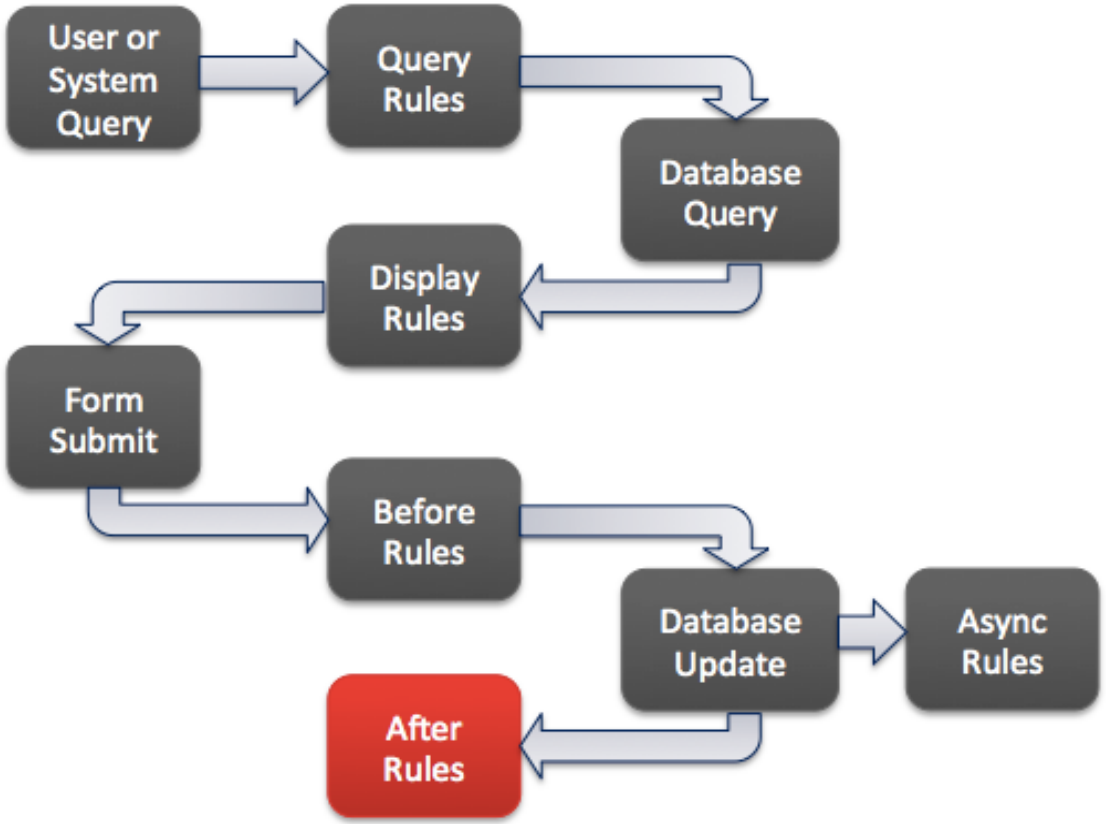


© 2014 ServiceNow All Rights Reserved

Business Rules

- After
  - Generally used for managing related tables/fields

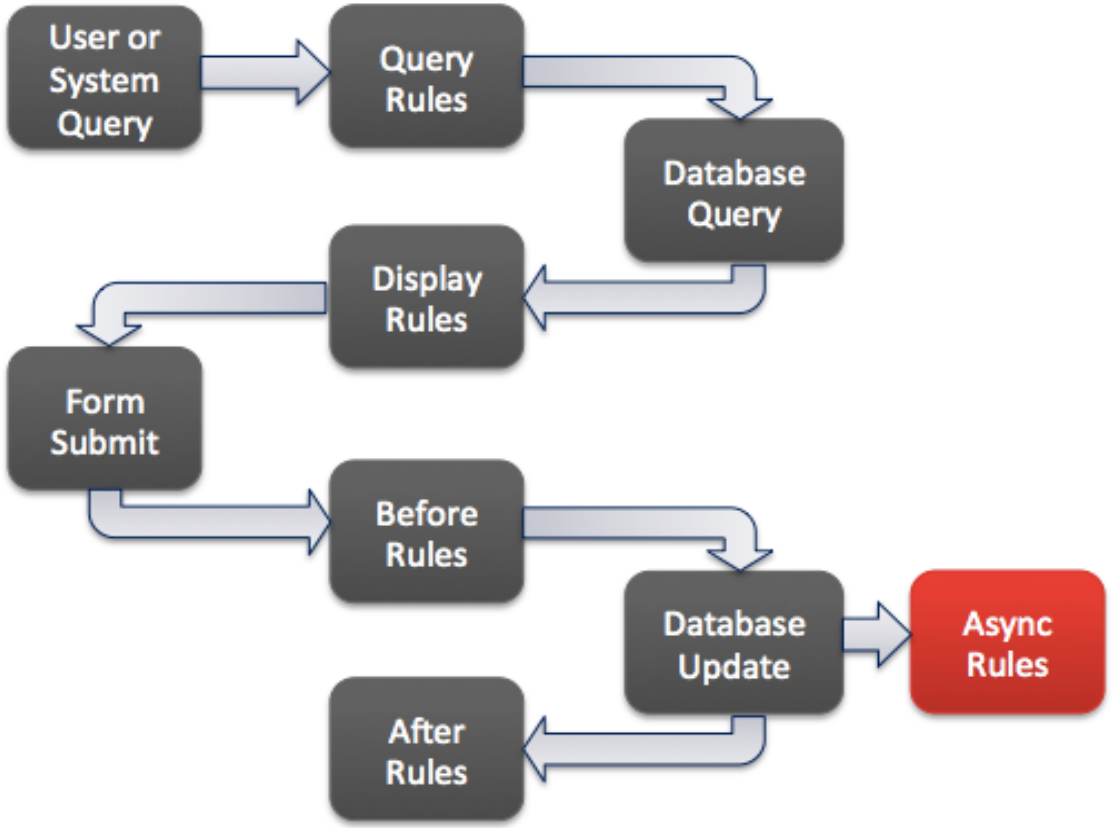




© 2014 ServiceNow All Rights Reserved

Business Rules

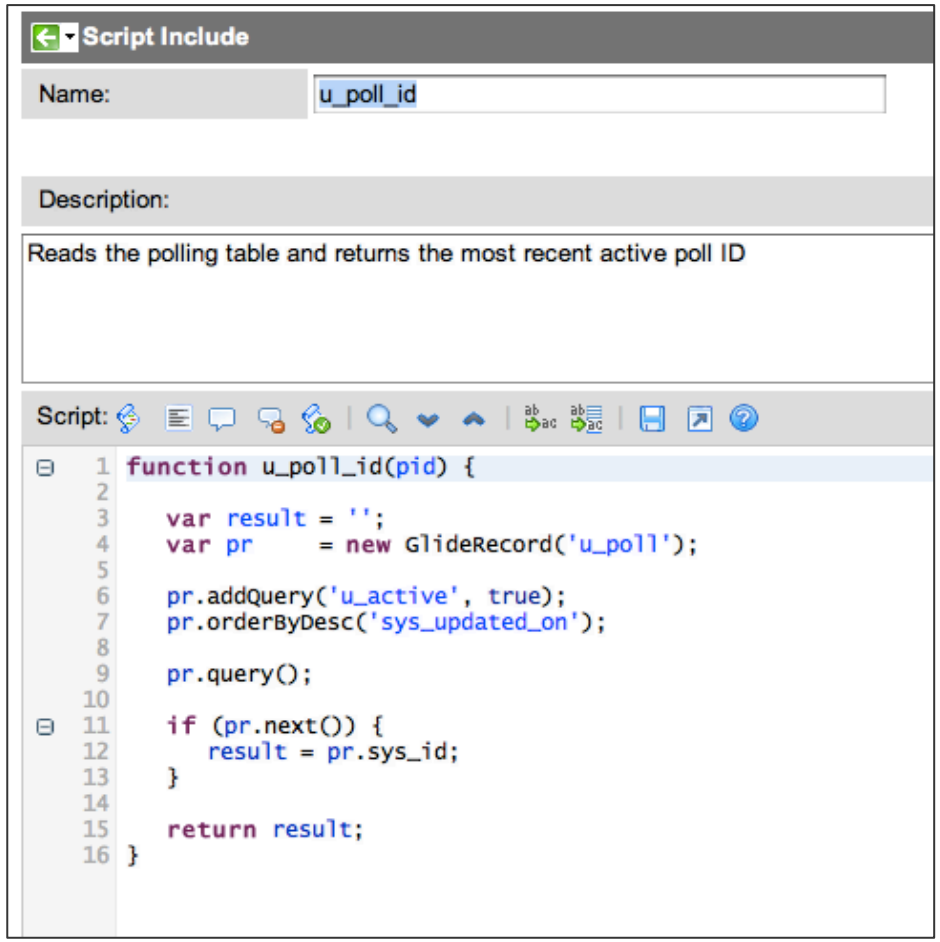
- Async
  - Consider as a replacement for after rules
    - Key indicator: Is the updated info required on the screen immediately?
  - Examples
    - Processing Email
    - Calculating SLAs
    - Generating report fields



© 2014 ServiceNow All Rights Reserved

Avoid Global Business Rules

- GBRs are loaded for every page
- Script includes only get loaded when required
- Converting is easy



© 2014 ServiceNow All Rights Reserved

Encapsulate Code In Functions

Particularly true for business rules

- By default, all variables and functions are global!
- Several business rules running with the statement…

```
var gr = new GlideRecord('incident');
```

- …can have unpredictable conflicts

© 2014 ServiceNow All Rights Reserved

Encapsulate Code In Functions

- The variable ‘gr’ is only known to the function doStuff()
- Now, the “doStuff()” function is in the global name space…
- Use good function naming standards and conditions to mitigate this
  - Ex: acme_chg_closure();

```
doStuff(); // Do 1 thing - call the function

/*
 * doStuff - a simple example of something
 *
 * @param - None
 * @return - None
 *
 */
function doStuff() {

    var gr = new GlideRecord('incident');

    gr.addQuery('active', true);
    gr.query();

    while (gr.next()) {
        // Do something important here
        // with each active incident record
    }
}
```

© 2014 ServiceNow All Rights Reserved

Anonymous Functions

- If you're not good at naming functions and want to avoid the issue of the global name space, you can also use "anonymous functions" with format similar to the following example:

```
(function() {

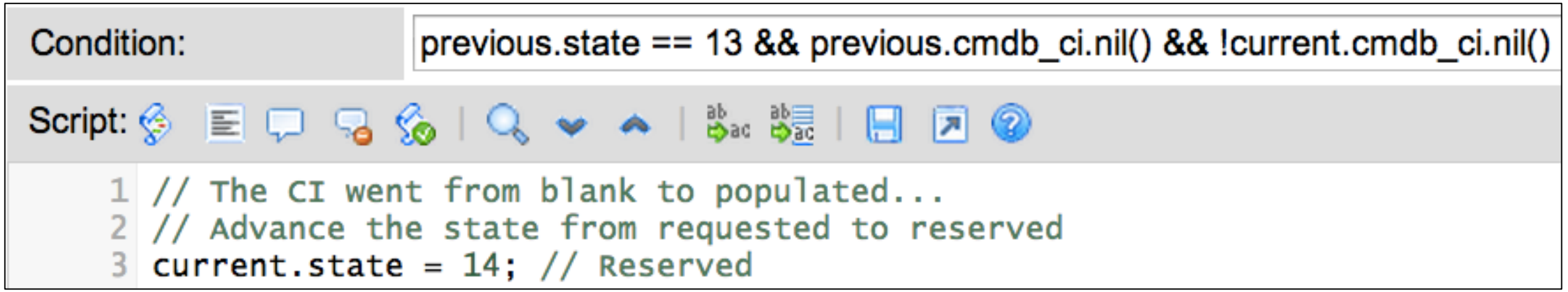
    var gr = new GlideRecord('u_my_table');

    gr.state = 3; // Closed complete
    gr.active = false;
    gr.update();
})();
```

© 2014 ServiceNow All Rights Reserved

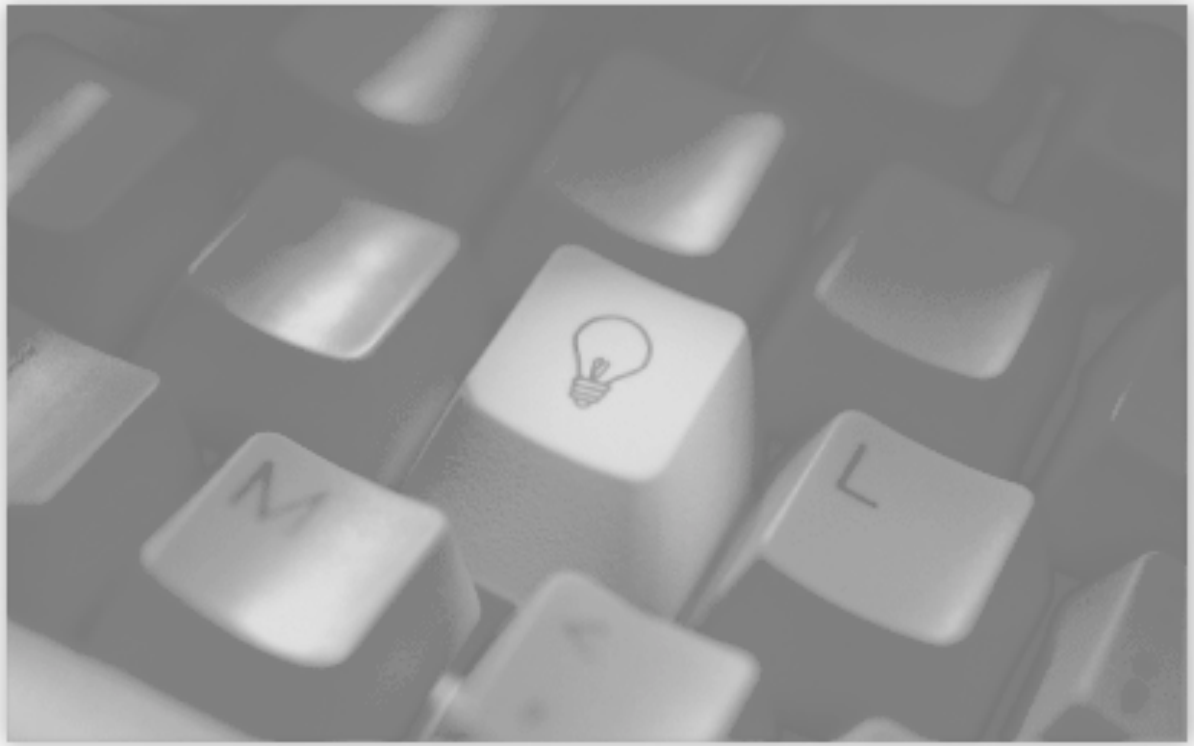
Other Business Rule Best Practices

- Make business rules small and specific
  - Easier to debug and maintain
- Use conditions to prevent unnecessary execution



© 2014 ServiceNow All Rights Reserved

Lab 1:



1.1 Review Business Rules
1.2 Convert a global business rule to a script include

© 2014 ServiceNow All Rights Reserved

Client Script Best Practices

© 2014 ServiceNow All Rights Reserved

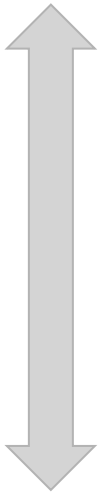
Reduce Client Side Logic

- Client side = JavaScript code that runs on the browser
- When well designed, it can reduce the time it takes to complete a form
- Try using server side when ever possible
  - Not everything needs to be a client script
- Client scripts and UI policies run on forms only
  - Disable list editing (with an ACL) if you must use client side scripting
- Minimize server lookups

© 2014 ServiceNow All Rights Reserved

Getting Server Data

Options to get server side data to the client



Most Preferred

g_scratchpad

GlideAjax

g_form.getReference()

GlideRecord Query

Least Preferred

© 2014 ServiceNow All Rights Reserved

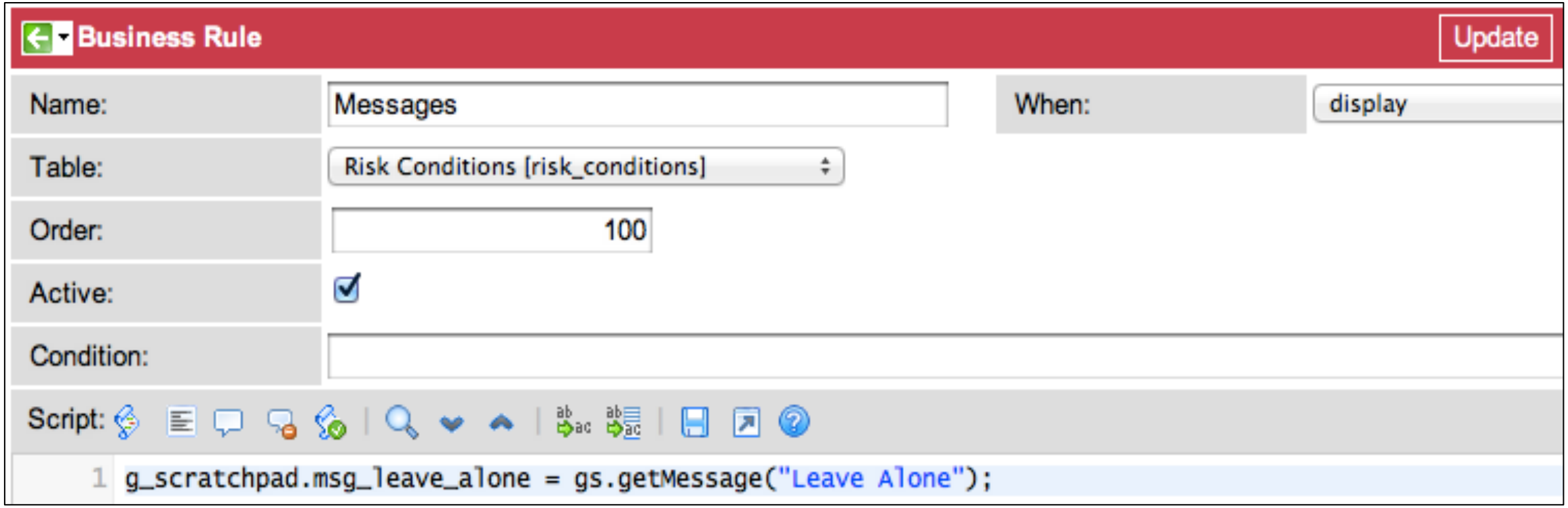
g_scratchpad

- Great for passing information to your form (on load)
- No round trip required
- Does not update fields dynamically (onChange)



© 2014 ServiceNow All Rights Reserved

g_scratchpad – Business Rule



© 2014 ServiceNow All Rights Reserved

g_scratchpad – Client Script

```
function onLoad() {
  var msg = g_scratchpad.msg_leave_alone;
  var r   = g_form.getValue('risk');
  var i   = g_form.getValue('impact');

  g_form.removeOption('risk', '');
  g_form.removeOption('impact', '');
  g_form.addOption('risk', '', msg, 0);
  g_form.addOption('impact', '', msg, 0);
  g_form.setValue('risk', r);
  g_form.setValue('impact', i);
}
```

© 2014 ServiceNow All Rights Reserved

GlideAjax

- Pros
  - Callback (async/background) functionality
  - More flexibility in what's returned
- Cons
  - A little more complex to create
    - Client and Server side scripts to write/maintain
  - Requires a round trip from the client to server

Key differentiator between g_scratchpad and GlideAjax:

- GlideAjax can be used for dynamic updates
- g_scratchpad only passes information when the form is loaded

© 2014 ServiceNow All Rights Reserved

GlideAjax Example

- Server

```
var HelloWorld = Class.create();
HelloWorld.prototype = Object.extendsObject(AbstractAjaxProcessor, {
    alertGreeting: function() {
        return "Hello " + this.getParameter('sysparm_user_name') + "!";
    }
});
```

- Client

```
var ga = new GlideAjax('HelloWorld');
ga.addParam('sysparm_name','alertGreeting');
ga.addParam('sysparm_user_name',"Bob");
ga.getXML(HelloWorldParse);

function HelloWorldParse(response) {
    var answer = response.responseXML.documentElement.getAttribute("answer");
    alert(answer);
}
```

© 2014 ServiceNow All Rights Reserved

setValue() with Reference Fields

- Usage: g_form.setValue(fieldName, value, displayValue);
- Incorrect:

```
var id = '5137153cc611227c000bbd1bd8cd2005';
g_form.setValue('assigned_to', id);
```

CAUSES A SYNCHRONOUS AJAX CALL TO GET THE DISPLAY VALUE

- Correct

```
var id = '5137153cc611227c000bbd1bd8cd2005';
var name = 'Fred Luddy';
g_form.setValue('assigned_to', id, name);
```

© 2014 ServiceNow All Rights Reserved

GlideAjax - Example

Passing back a reference and display name

```
var GetUserStuff = Class.create();
GetUserStuff.prototype = Object.extendsObject(AbstractAjaxProcessor, {
   getLocation : function() {

      var usr    = new GlideRecord('sys_user');
      var usr_id = this.getParameter('sysparm_caller_id');

      if (usr.get(usr_id)) {
         return usr.location + ':' + usr.location.getDisplayValue();
      } else {
         return '';
      }
   },
});
```

© 2014 ServiceNow All Rights Reserved

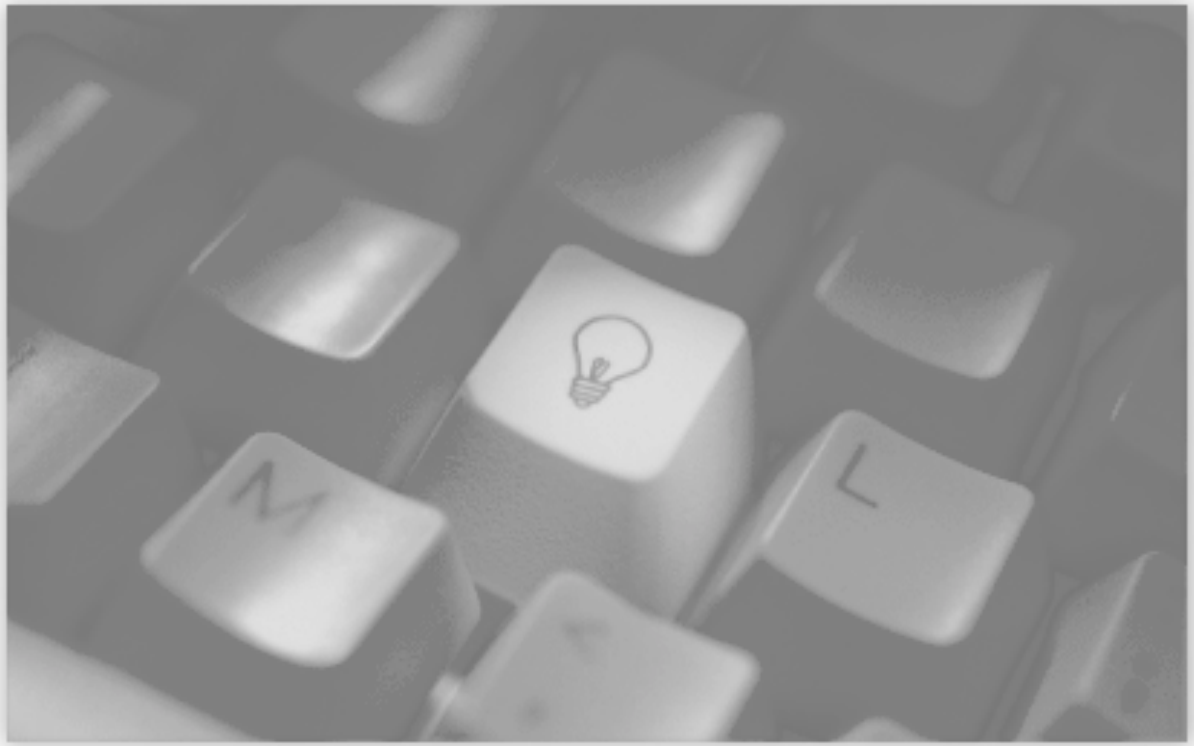
Avoid DOM manipulation

- Avoid getElementById() or gel() calls
- Use g_form methods such as:
  - g_form.getControl()
  - g_form.setValue()

g_form insulates you from browser compatibility issues

© 2014 ServiceNow All Rights Reserved

Lab 2:



2.1 Convert a getReference to GlideAjax
2.2 Use g_scratchpad

© 2014 ServiceNow All Rights Reserved

Coding Best Practices

© 2014 ServiceNow All Rights Reserved

Comment Your Work

- Get into the habit now
- You **will** forget why you did something

```
// a single line comment
var gr = new GlideRecord('incident'); // here, too!
```

```
/*
 * multi line comments
 */
```

```
////////////////////////////
// BOXES ARE EASY TO SPOT //
////////////////////////////
```

© 2014 ServiceNow All Rights Reserved

Avoid Hard Coded Values

- Avoid using hard coded values in scripts

```
var taskID = '0b74...';
```

```
var groupName = '<...>';
```

- Alternative

```
var taskID = gs.getProperty('acme.task.default');
var groupName = gs.getProperty('acme.group.name');
```

- Make a mental note as you go: *"What if this thing in quotes changes?"*

© 2014 ServiceNow All Rights Reserved

Avoid Hard Coded Values

Passing back a reference and display name

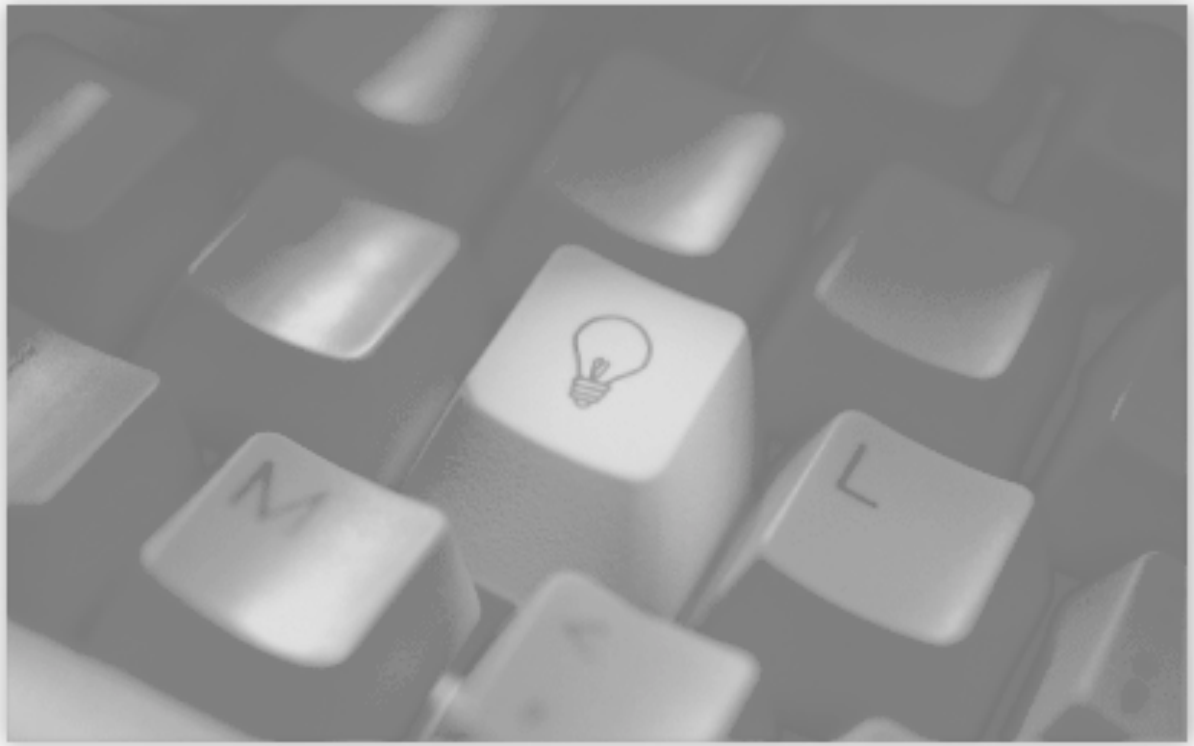
- Initial Approach
  - Create a user approval activity for Sara
  - Challenge: Dale is the new IT Director
  - Requires you to update the workflow!

- Better Approach:
  - Use Group Approval Activity
  - Create a group “IT Director”
  - Make Sara a member
  - When Dale becomes the new IT Director, just update the group membership, not the workflow



© 2014 ServiceNow All Rights Reserved

Lab 3:



3.1 Hard Coded Sys_id

© 2014 ServiceNow All Rights Reserved

Counting Records

GlideRecord

- Slow!
- Not scalable – the more records, the longer it takes
- Easy to write

GlideAggregate

- Fast
- Scalable: Leverages a database function to count records
- 3 more lines of code

© 2014 ServiceNow All Rights Reserved

Row Count Example

Worst Case

- GlideRecord – don't use: while (gr.next())

```
gs.print('>>>DEBUG: ' + countEm() + ' records read');

function countEm() {

    var gr = new GlideRecord('sys_audit');
    var count = 0;

    gr.query();
    while (gr.next()) {
        count++;
    }

    return count;
}
```

- 490K records: 1m 40s

© 2014 ServiceNow All Rights Reserved

Row Count Example

- GlideRecord – getRowCount() is not as bad, but…

```
gs.print('>>>DEBUG: Starting row count...');
gs.print('>>>DEBUG: ' + countEm() + ' records read');

function countEm() {

    var gr = new GlideRecord('sys_audit');

    gr.query();
    return gr.getRowCount();
}
```

- 490K records: >2.1 seconds

© 2014 ServiceNow All Rights Reserved

Row Count Example

BEST WAY!

- GlideAggregate

```
gs.print('>>>DEBUG: Starting row count...');
gs.print('>>>DEBUG: ' + countEm() + ' records read');

function countEm() {

    var count  = new GlideAggregate('sys_audit');
    var retVal = 0;

    count.addAggregate('COUNT');
    count.query();

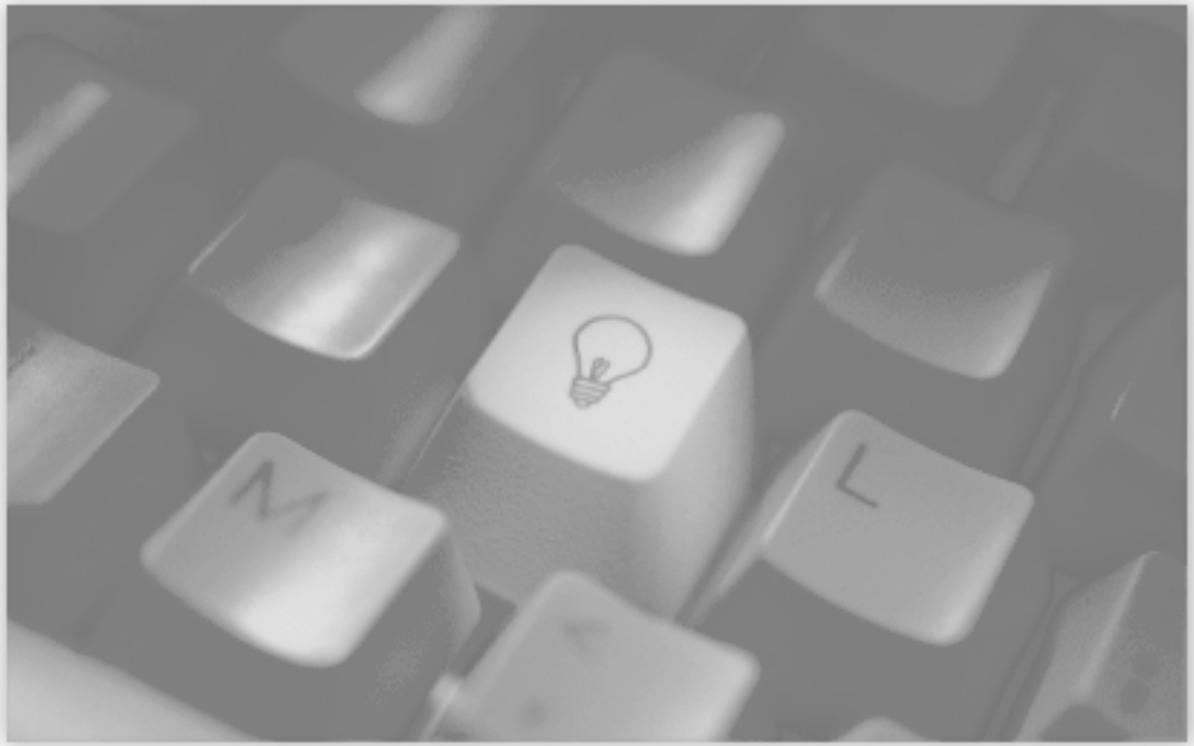
    if (count.next()) {
        retVal = count.getAggregate('COUNT');
    }

    return retVal;
}
```

- 490K records: 1-3 ms

© 2014 ServiceNow All Rights Reserved

Lab 4:



4.1 Row Counting

© 2014 ServiceNow All Rights Reserved

More Information

ServiceNow Wiki

servicenow

CATEGORIES

- **Get Started**
- Use
- Administer
- Script
- Build
- Deliver
- Integrate
- Release Notes
- Technical Support
- Books
- Video Tutorials

Technical Best Practices

Learn to increase the effectiveness of your ServiceNow experience.

System Performance Best Practices
Debugging Tools Best Practices
Customizing the UI Appearance
[more]

© 2014 ServiceNow All Rights Reserved

Top Takeaways

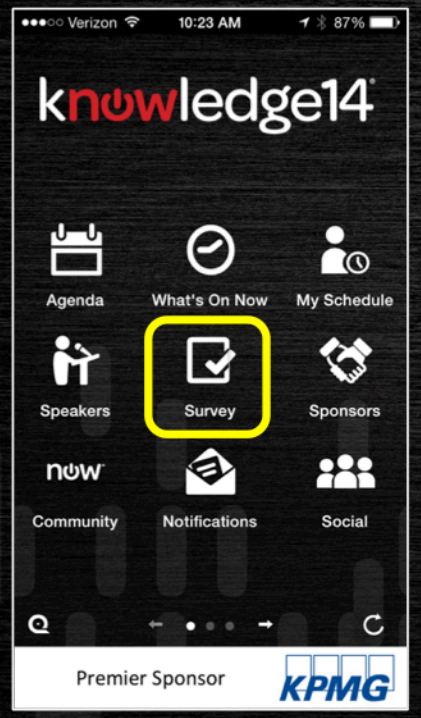
1. Use before/after/async business rules effectively

2. Use asynchronous GlideAjax in client scripts to retrieve data from the server

3. Manage data & properties to reduce maintenance overhead

© 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

Chuck Tomasi

Sr. Services Enablement Program Manager

ServiceNow

chuck.tomasi@servicenow.com

Dave Slusher

Sr. Technical Consultant

ServiceNow

dave.slusher@servicenow.com

Want a copy of this deck?

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

