ServiceNow Application Developer

Server-side Scripting > Test Your Server-side Scripting Knowledge

Want to verify your understanding of server-side scripting? These questions will help you assess your progress. For each question, determine your response then click anywhere in the question to see the answer.

▼ **QUESTION**: Which of these classes are part of the ServiceNow server-side API?

- *GlideSystem (gs)*
- *GlideUser (g_user)*
- *GlideDateTime*
- *GlideDate*
- *GlideForm (g_form)*

ANSWER: **GlideSystem**, **GlideDateTime**, and **GlideDate** are part of the ServiceNow server-side API. The server-side API also has a *GlideUser* class but the server-side GlideUser class does not use the g_user object. If you are not sure whether a class is part of the client-side or server-side API, check the **API Reference (https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=no-namespace)**.

▼ **QUESTION**: When do before Business Rules execute their script logic?

1. Before a form loads

2. After a form loads but before control is given to the user

3. Before onChange Client Scripts

4. Before Business Rule *Actions*

5. Before records are written to the database

ANSWER: **5**. onBefore Business Rules execute their script logic before records are updated in the database.

▼ **QUESTION**: What is the difference between an after Business Rule and an async Business Rule?

ANSWER: Both after and async Business Rules execute their script logic after records are written to the database. after Business Rules execute their logic *immediately after* a record is written to the database. async Business Rules create scheduled jobs that run *soon after* a record is written to the database.

▼ **QUESTION**: Which of the following are NOT methods from the *GlideRecord* API? More than one response may be correct.

1. addQuery()

2. addEncodedQuery()

3. addOrQuery()

4. addAndQuery()

5. query()

ANSWER: **3** and **4**. addOrQuery() and addAndQuery() are not methods from the *GlideRecord* API. If a script contains multiple statements that use the addQuery() method the queries are ANDed. To explicitly AND or OR a condition in a query, use the methods from the ***GlideQueryCondition*** **(https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=c_GlideQueryConditionScopedAPI)** class.

▼ **QUESTION**: Which of the following are NOT true about the *current* object? More than one response may be correct.

1. The *current* object is automatically instantiated.
2. The *current* object property values never change after a record is loaded from the database.
3. The *current* and *previous* objects are always identical.
4. The *current* and *previous* objects are sometimes identical.
5. The properties of the *current* object are the same for all Business Rules.

ANSWER: **2**, **3**, and **5**. Although the *current* object's property values do not have to change, they can. The *current* object's property values are sometimes identical to the *previous* object's properties. For example, the *current* and *previous* objects are identical immediately after a record is loaded from the database. The properties of the *current* object for Business Rules are dependent on table the Business Rule is for.

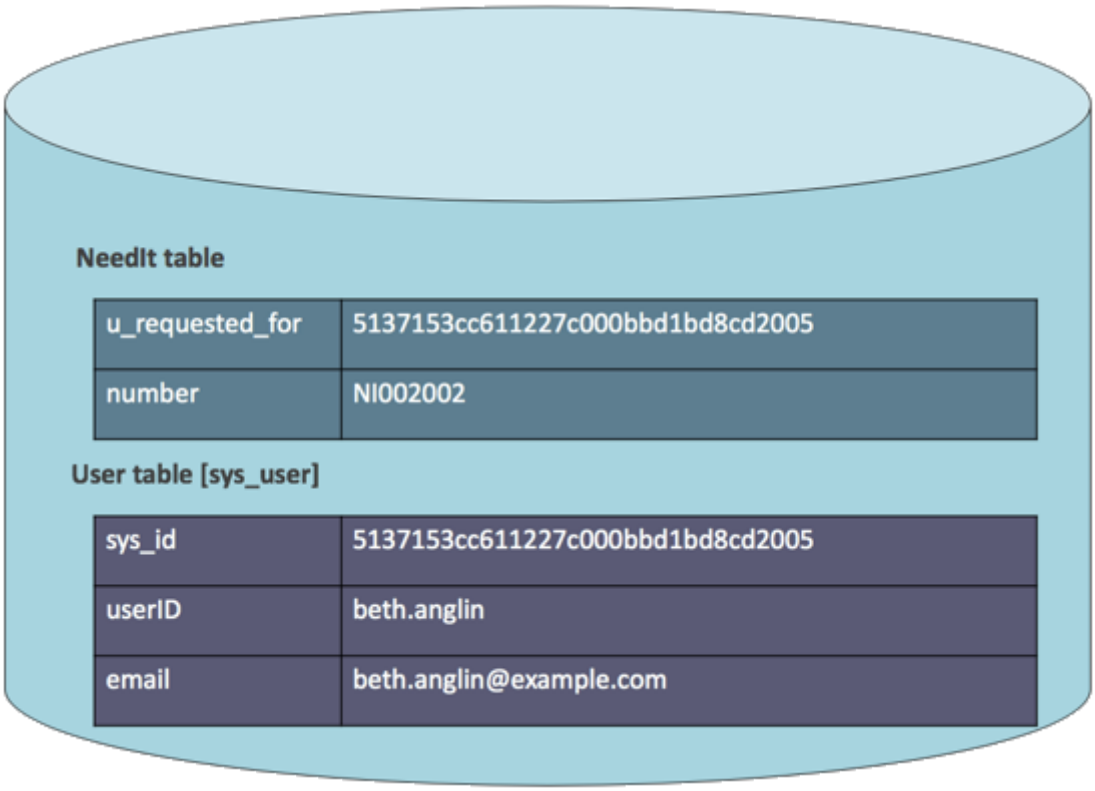
▼ **QUESTION**: What value does a Business Rule *Condition* field return if the field is empty?

1. True
2. False
3. Neither

4. Both

ANSWER: **1**. If there is no value in the *Condition* field, the field returns *true*. Business Rule scripts only execute their script logic if the *Condition* field returns *true*.

▼ **QUESTION**: Examine the database tables and fields.



Based on the database, which one of the following is valid dot-walking syntax?

1. u_requested_for.userID
2. current.u_requested_for.userID
3. number.userID
4. current.number.userID

ANSWER: **2**. Dot-walking allows direct scripting access to fields and field values on related records.

The dot-walking syntax is:

```
<object>.<related_object>.<field_name>
```

▼ **QUESTION**: Which of the following are true about Script Includes? More than one response may be correct.

1. Script Includes are reusable server-side script logic
2. Script includes can extend an existing class
3. Script includes can define a new class or function
4. Script includes can be client callable
5. Script includes execute their script logic only when explicitly called

ANSWER: **1**, **2**, **3**, **4**, and **5**. All of the statements are true for Script Includes.
