

### ◆ Level 3 – Advanced Developer Role (L3)

---

- **What is the difference between GlideRecord and GlideAggregate?**

GlideRecord is used for general database operations like reading, writing, and updating records. GlideAggregate is a subclass of GlideRecord that's used to perform database aggregation operations such as COUNT, SUM, AVG, etc. efficiently.

- **What is the use of Async Business Rules and when to use them?**

Async Business Rules run in the background after the database action is complete. They're useful for non-critical tasks like sending notifications or updating related records, where user wait time is not a concern.

- **What is the purpose of the Display Business Rule?**

Display Business Rules run when a form loads and are used to pass data from the server to the client side using g\_scratchpad. This is useful for pre-loading values into client scripts.

- **How can you call a Script Include from a Client Script?**

You can use GlideAjax to call a Script Include. The Script Include must be client-callable (set to Client Callable = true) and include a type: 'class' structure.

- **What are Scoped Applications and how are they different from Global applications?**

Scoped Applications are isolated from other applications in terms of scripts and tables. They are useful in multi-developer or vendor-built apps. Global apps share scripts and resources across the platform.

- **Explain the difference between Event and Notification in ServiceNow.**

An Event is a system trigger that signals something has occurred. Notifications are used to respond to these events and send messages to users through various channels like email or SMS.

- **What are Script Actions in ServiceNow?**

Script Actions are server-side scripts that execute in response to Events. They allow you to execute complex logic after a specific system condition is met.

- **What is a Catalog Client Script and how is it different from normal Client Script?**

Catalog Client Scripts are used specifically for Service Catalog items. They behave similarly to Client Scripts but are scoped to catalog forms and variables.

- **How can you secure data at the row and column level in ServiceNow?**

Use Access Control Rules (ACLs) to secure table rows and specific fields (columns). You can define conditions, required roles, or scripts to restrict data access.

- **What are Scripted REST APIs and when do you use them?**

Scripted REST APIs allow developers to define custom RESTful endpoints. They're used when out-of-the-box integration isn't enough and custom logic is needed.

- **What is the difference between Inbound and Outbound Web Services?**

Inbound Web Services allow external systems to call ServiceNow (e.g., using REST/SOAP APIs). Outbound Web Services enable ServiceNow to call external systems.

- **What is a GlideRecordSecure?**

GlideRecordSecure is similar to GlideRecord but automatically enforces ACL checks. It ensures that security rules are applied during data access.

- **How do you use try/catch in server-side scripting in ServiceNow?**

Use `try { ... } catch(ex) { gs.error(ex); }` to handle exceptions gracefully and prevent script failures from affecting system performance.

- **How can you improve the performance of your scripts?**

- Avoid unnecessary GlideRecord queries.
- Use indexed fields in filters.
- Batch updates using Scheduled Jobs.
- Avoid synchronous long-running scripts.

- **Explain Domain Separation and its implications in scripting.**

In Domain-separated environments, your scripts must consider domain-specific data access and scoping. Use `gs.getUserDomainID()` and `gs.getSession().currentDomainID` when needed.

- **What is a Flow Designer Subflow and how is it used?**

A Subflow is a reusable sequence of actions that can be called from other flows or subflows. It promotes modularity and reusability in workflow automation.

- **What are Caching strategies in ServiceNow scripting?**

Use variables like `g_scratchpad` or `GlideCache` to temporarily store data during a session to avoid repetitive `GlideRecord` queries.

- **How do you test and debug Script Includes and Business Rules?**

Use `gs.info()` statements and monitor output in system logs. You can also use Background Scripts to test Script Includes directly.

- **What is the best practice for writing Client Scripts?**

- Avoid hardcoding values.
- Use `g_form` API efficiently.
- Limit use of synchronous `GlideAjax` calls.
- Always test performance and impact on user experience.

- **Explain the difference between current and previous in Business Rules.**

`current` refers to the record as it is being saved, while `previous` refers to the state of the record before any changes. This helps in detecting changes.

- **What is the use of Flow Logic like 'If' and 'For Each' in Flow Designer?**

These logical controls allow conditional execution and iteration over a list of items within a flow. They are useful for branching and looping logic.

- **How do you pass data from a flow to a subflow?**

You define input variables in the subflow and map them from the calling flow. The subflow can also return outputs to the parent flow.

- **What is Data Stream Action in IntegrationHub?**

It is used to fetch and process large datasets from external systems incrementally, ideal for performance and avoiding timeout issues.

- **How do you control Script Include visibility in different scopes?**

Set the Accessible from option to 'All application scopes' or 'This application scope only' depending on whether other scoped apps should access it.

- **What is the importance of Error Handling in Integration Scripts?**

Error handling ensures that integrations can fail gracefully. Use try/catch blocks, log errors, and send alerts or fallback actions to avoid data loss or system failures.