splunk>

# Splunk Fundamentals 1 Lab Exercises

Lab typographical conventions:

[`sourcetype=db_audit`] OR [`cs_mime_type`] indicates either a source type or the name of a field.

**NOTE:** Lab work will be done on your personal computer or virtual machine, no lab environment is provided. We suggest you **DO NOT** do the lab work on your production environment.

The lab instructions refer to these source types by the types of data they represent:

| Type | Sourcetype | Fields of interest |
|---|---|---|
| Web Application | `access_combined_wcookie` | `action, bytes, categoryId, clientip, itemId, JSESSIONID, productId, referer, referer_domain, status, useragent, file` |
| Database | `db_audit` | `Command, Duration, Type` |
| Web server | `linux_secure` | `COMMAND, PWD, pid, process` |

## Lab Module 12 – Creating Lookups

**NOTE:** This lab document has two sections. The first section includes the instructions without answers. The second section includes instructions with the expected search string (answer) in red.

Description
In this lab exercise, you will create a new automatic lookup that provides additional information for Buttercup Games products.

**Scenario**: The web application data does not contain name and price information for the products being sold. Users of your reports would like to see product names used in your reports, not just product Ids.

### Task 1: Download and examine the lookup file.

1. Open a new browser window and direct it to http://splk.it/productdata
2. The file **products.zip** will be downloaded to your system.
3. Use an archive tool to unarchive the file.
4. Once unarchived, you will see a file named products.csv.
5. Return to the browser window for your instance of Splunk Web or open a new one.
6. Navigate to the Search view. (If you are in the **Home** app, click **Search & Reporting** from the column on the left side of the screen. You can also access the Search view by clicking the **Search** menu option on the green bar at the top of the screen.)

### Task 2: Add a lookup file and create a lookup definition.

7. Navigate to: **Settings > Lookups > Lookup table files.**
8. Click **New**.

9. Save the lookup table file with these values:
   - Destination app: *search*
   - File: products.csv
   - Destination filename**:** *products.csv*
10. Navigate to **Settings > Lookups > Lookup definitions**.
11. Make sure **Search & Reporting** is selected for **App context** and Click **New**.
12. Save the lookup table file with these values:
    - Destination app: *search*
    - Name: products_lookup
    - Type**:** File-based
    - Lookup file: *products.csv*
13. Return to the Search view.
14. Use `inputlookup` command to verify the lookup definition was created correctly.

*Example Results:*

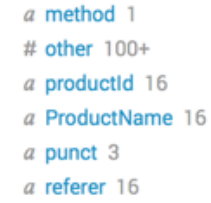
| Code | categoryId | price | productId | product_name |
|---|---|---|---|---|
| A | STRATEGY | 24.99 | DB-SG-G01 | Mediocre Kingdoms |
| B | STRATEGY | 39.99 | DC-SG-G02 | Dream Crusher |
| C | STRATEGY | 24.99 | FS-SG-G03 | Final Sequel |
| D | SHOOTER | 24.99 | WC-SH-G04 | World of Cheese |
| E | TEE | 9.99 | WC-SH-T02 | World of Cheese Tee |
| F | STRATEGY | 4.99 | PZ-SG-G05 | Puppies vs. Zombies |

**Task 3: Use the lookup in a search.**

**NOTE:** For this course, you will be searching across all time using the main index. This is NOT a best practice in a production environment, but needed for these labs due to the nature of the limited dataset.

15. Search the web application data for all events where a user purchased a product successfully.
16. Use the `lookup` command and reference the lookup table you just created. Match the `productId` in lookup table to the `productId` field in the event data. Use the OUTPUT function to output the `product_name` lookup table data to a `ProductName` field.
17. Notice that there is now a `ProductName` field in the fields list.

*Example:*




18. Change the search to use a `stats count` function to count events by `ProductName`.

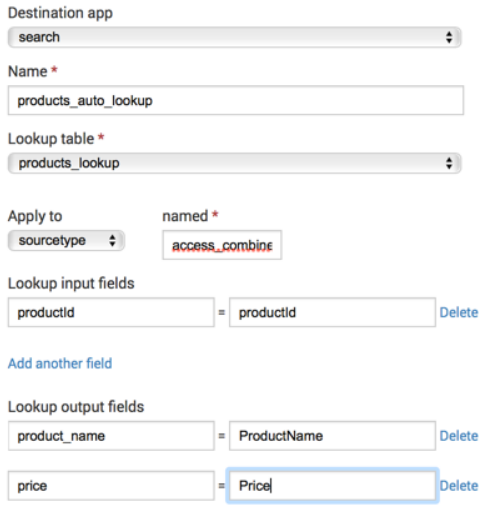
*Example Results:*

| ProductName | count |
|---|---|
| Benign Space Debris | 935 |
| Curling 2014 | 935 |
| Dream Crusher | 1308 |
| Final Sequel | 1155 |
| Fire Resistance Suit of Provolone | 1192 |

## Task 4: Create an automatic lookup definition.

19. Navigate to **Settings > Lookups > Automatic lookups**
20. Make sure **Search & Reporting** is selected for **App context** and Click **New**.
21. Save the automatic lookup with these values:
    - Destination app: *search*
    - Name: *products_auto_lookup*
    - Lookup table**:** *products_lookup*
    - Apply to: *sourcetype*
    - named: *access_combined_wcookie*
    - Lookup input fields: *productId = productId*
    - Lookup output fields: *product_name = ProductName*
      *price = Price*

*Example:*




## Task 5: Verify your automatic lookup is working.

22. Return to the Search view.
23. Search the web application data for all events where a user purchased a product successfully. Use the `stats sum` function to sum the `price` field by `ProductName`. Name the resulting field `Revenue`.
24. Use the `sort` command to find the product that has generated the largest revenue. Take note of the `ProductName` as you might be asked to recall it in the module quiz.
25. Save the report as a dashboard panel on your `Sales Dashboard`.

splunk>

# Splunk Fundamentals 1 Lab Exercises

Lab typographical conventions:

[sourcetype=db_audit] OR [cs_mime_type] indicates either a source type or the name of a field.

**NOTE:** Lab work will be done on your personal computer or virtual machine, no lab environment is provided. We suggest you **DO NOT** do the lab work on your production environment.

The lab instructions refer to these source types by the types of data they represent:

| Type | Sourcetype | Fields of interest |
|---|---|---|
| Web Application | access_combined_wcookie | action, bytes, categoryId, clientip, itemId, JSESSIONID, productId, referer, referer_domain, status, useragent, file |
| Database | db_audit | Command, Duration, Type |
| Web server | linux_secure | COMMAND, PWD, pid, process |

## Lab Module 12 – Creating Lookups with Solutions

**NOTE:** This lab document has two sections. The first section includes the instructions without answers. The second section includes instructions with the expected search string (answer) in red.

Description
In this lab exercise, you will create a new automatic lookup that provides additional information for Buttercup Games products.

**Scenario**: The web application data does not contain name and price information for the products being sold. Users of your reports would like to see product names used in your reports, not just product Ids.

### Task 1: Download and examine the lookup file.

1. Open a new browser window and direct it to http://splk.it/productdata
2. The file **products.zip** will be downloaded to your system.
3. Use an archive tool to unarchive the file.
4. Once unarchived, you will see a file named products.csv.
5. Return to the browser window for your instance of Splunk Web or open a new one.
6. Navigate to the Search view. (If you are in the **Home** app, click **Search & Reporting** from the column on the left side of the screen. You can also access the Search view by clicking the **Search** menu option on the green bar at the top of the screen.)

### Task 2: Add a lookup file and create a lookup definition.

7. Navigate to: **Settings > Lookups > Lookup table files.**
8. Click **New**.

9. Save the lookup table file with these values:
   - Destination app: *search*
   - File: products.csv
   - Destination filename**:** *products.csv*
10. Navigate to **Settings > Lookups > Lookup definitions**.
11. Make sure **Search & Reporting** is selected for **App context** and Click **New**.
12. Save the lookup table file with these values:
    - Destination app: *search*
    - Name: products_lookup
    - Type**:** File-based
    - Lookup file: *products.csv*
13. Return to the Search view.
14. Use `inputlookup` command to verify the lookup definition was created correctly.

    (| inputlookup products_lookup)

*Example Results:*

| Code | categoryId | price | productId | product_name |
|---|---|---|---|---|
| A | STRATEGY | 24.99 | DB-SG-G01 | Mediocre Kingdoms |
| B | STRATEGY | 39.99 | DC-SG-G02 | Dream Crusher |
| C | STRATEGY | 24.99 | FS-SG-G03 | Final Sequel |
| D | SHOOTER | 24.99 | WC-SH-G04 | World of Cheese |
| E | TEE | 9.99 | WC-SH-T02 | World of Cheese Tee |
| F | STRATEGY | 4.99 | PZ-SG-G05 | Puppies vs. Zombies |

**Task 3: Use the lookup in a search.**

**NOTE:** For this course, you will be searching across all time using the main index. This is NOT a best practice in a production environment, but needed for these labs due to the nature of the limited dataset.

15. Search the web application data for all events where a user purchased a product successfully. (index=main sourcetype=access_combined_wcookie status=200 file=success.do)
16. Use the `lookup` command and reference the lookup table you just created. Match the `productId` in lookup table to the `productId` field in the event data. Use the OUTPUT function to output the `product_name` lookup table data to a `ProductName` field.

    (index=main sourcetype=access_combined_wcookie status=200 file=success.do | lookup products_lookup productId as productId OUTPUT product_name as ProductName)
17. Notice that there is now a `ProductName` field in the fields list.

*Example:*

a method 1
# other 100+
a productId 16
a ProductName 16
a punct 3
a referer 16

18. Change the search to use a `stats count` function to count events by `ProductName`.

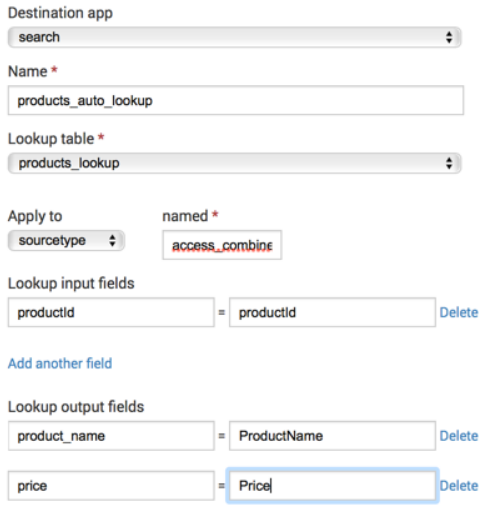
*Example Results:*

| ProductName | count |
|---|---|
| Benign Space Debris | 935 |
| Curling 2014 | 935 |
| Dream Crusher | 1308 |
| Final Sequel | 1155 |
| Fire Resistance Suit of Provolone | 1192 |

**Task 4: Create an automatic lookup definition.**

19. Navigate to **Settings > Lookups > Automatic lookups**
20. Make sure **Search & Reporting** is selected for **App context** and Click **New**.
21. Save the automatic lookup with these values:
    - Destination app: *search*
    - Name: *products_auto_lookup*
    - Lookup table**:** *products_lookup*
    - Apply to: *sourcetype*
    - named: *access_combined_wcookie*
    - Lookup input fields: *productId = productId*
    - Lookup output fields: *product_name = ProductName*
      *price = Price*

*Example:*




**Task 5: Verify your automatic lookup is working.**

22. Return to the Search view.
23. Search the web application data for all events where a user purchased a product successfully. Use the `stats sum` function to sum the `price` field by `ProductName`. Name the resulting field `Revenue`.

    (index=main sourcetype="access_combined_wcookie" file=success.do status=200 | stats sum(Price) as Revenue by ProductName)
24. Use the `sort` command to find the product that has generated the largest revenue. Take note of the `ProductName` as you might be asked to recall it in the module quiz.

    (Dream Crusher)
25. Save the report as a dashboard panel on your `Sales Dashboard`.