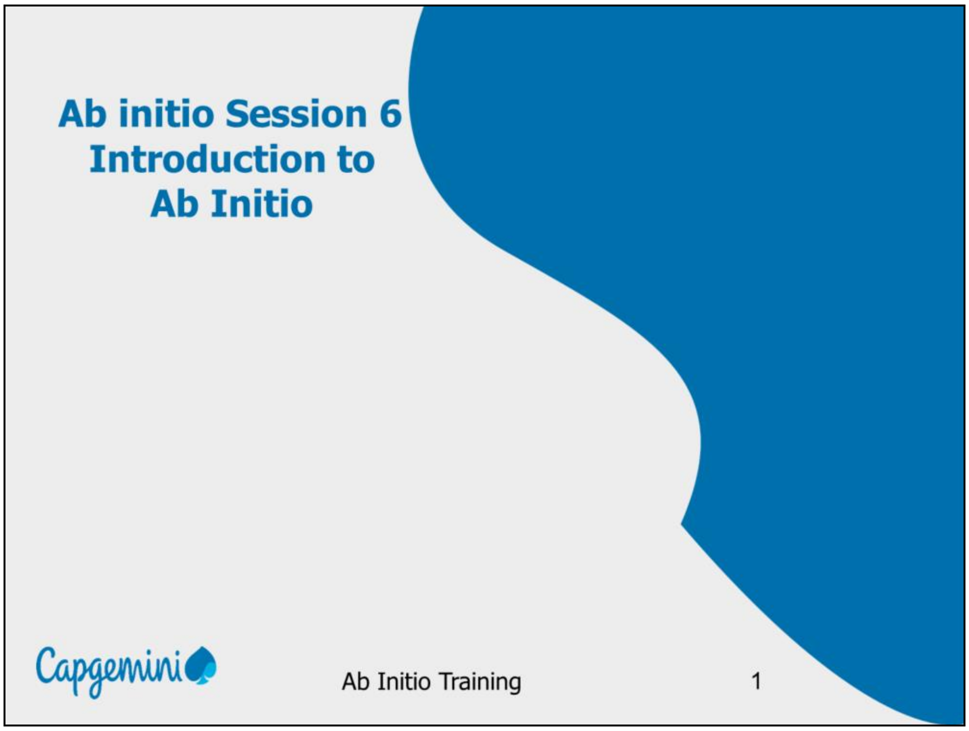
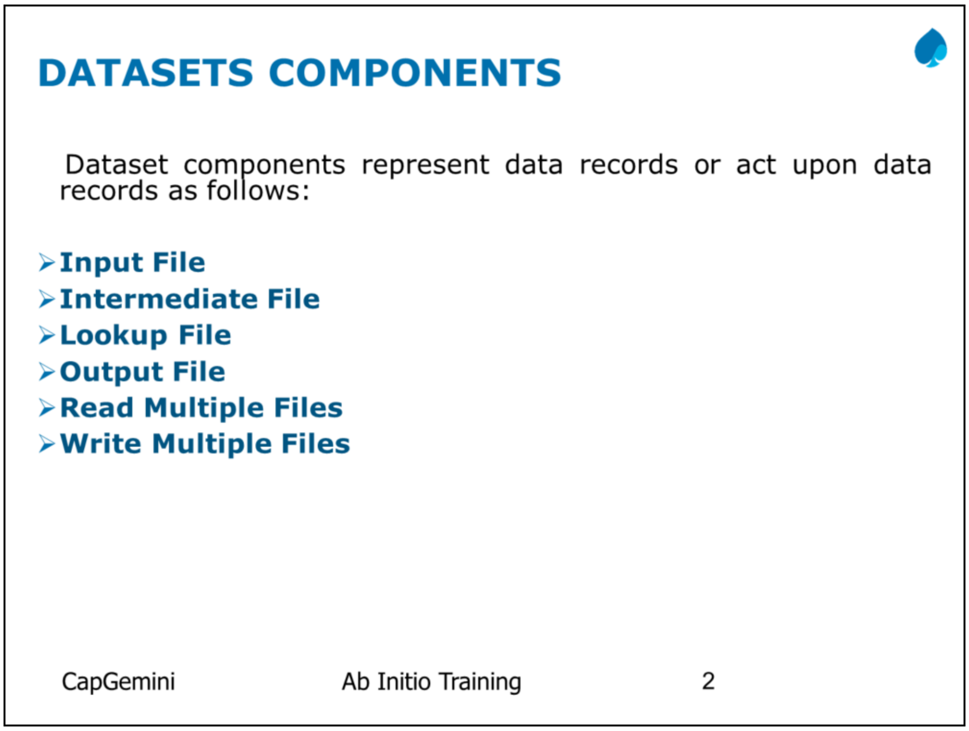


Ab initio Session 6 Introduction to Ab Initio

Ab Initio Training 1

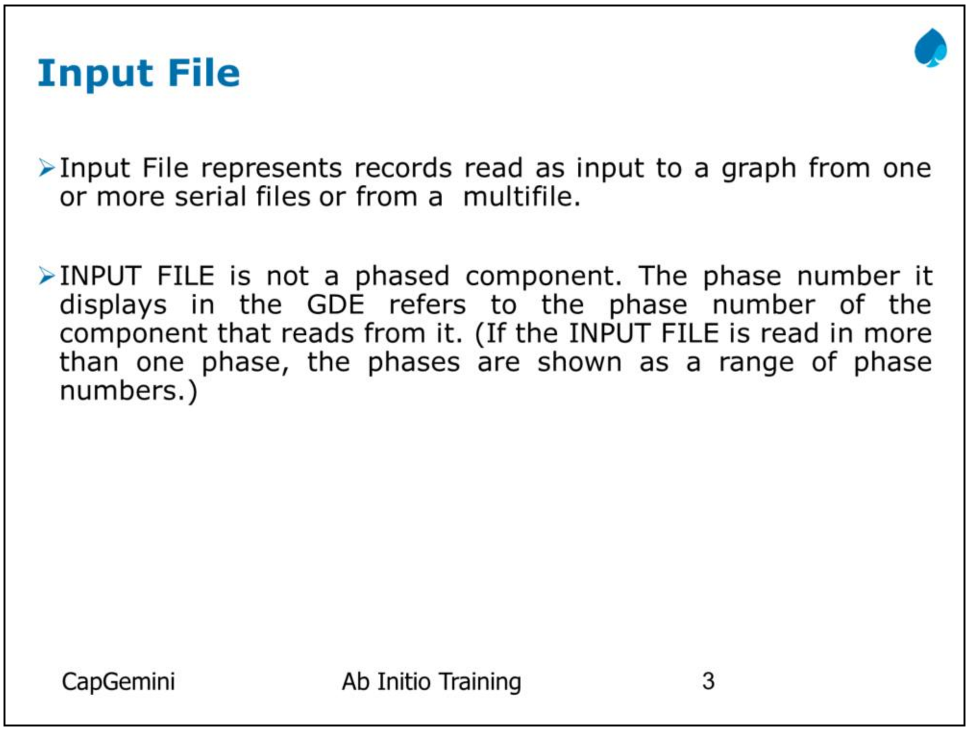


DATASETS COMPONENTS

Dataset components represent data records or act upon data records as follows:

- **Input File**
- **Intermediate File**
- **Lookup File**
- **Output File**
- **Read Multiple Files**
- **Write Multiple Files**

CapGemini Ab Initio Training 2



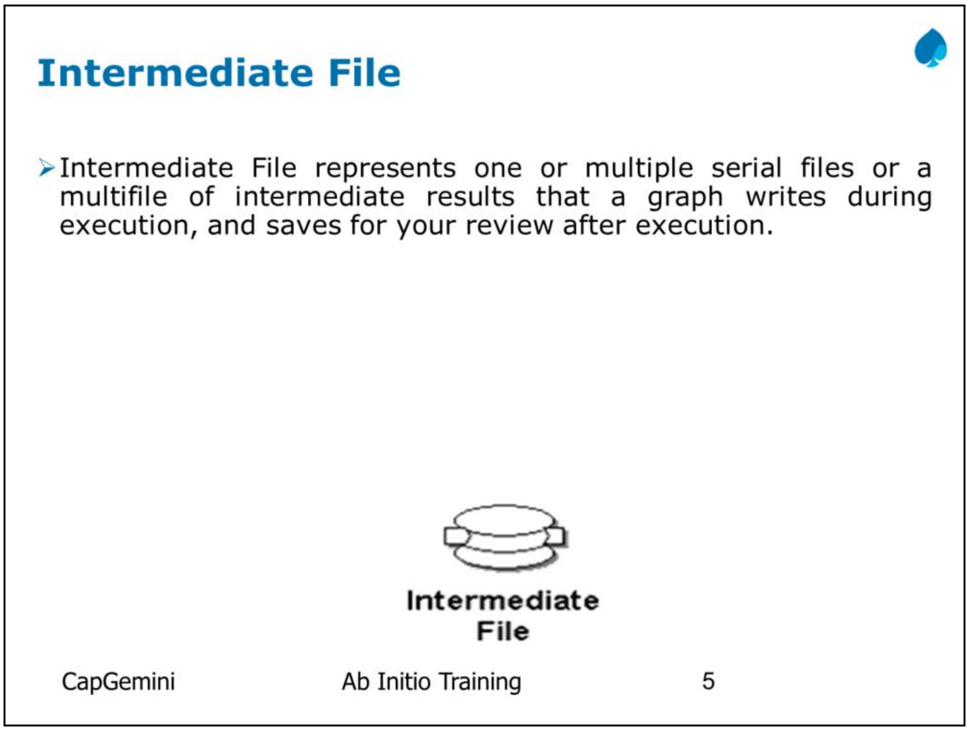
Input File

- Input File represents records read as input to a graph from one or more serial files or from a multifile.
- INPUT FILE is not a phased component. The phase number it displays in the GDE refers to the phase number of the component that reads from it. (If the INPUT FILE is read in more than one phase, the phases are shown as a range of phase numbers.)

CapGemini Ab Initio Training 3

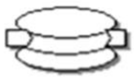
Parameters : Input File

The Input File ***Properties*** dialog does not have a prominent ***Parameters*** tab. However, you can specify values for parameters on the ***Description***, ***Access***, and ***Ports*** tabs of the Input File ***Properties*** dialog. This includes parameters such as input file location, file handling behavior and permissions, and the intermediate record format.



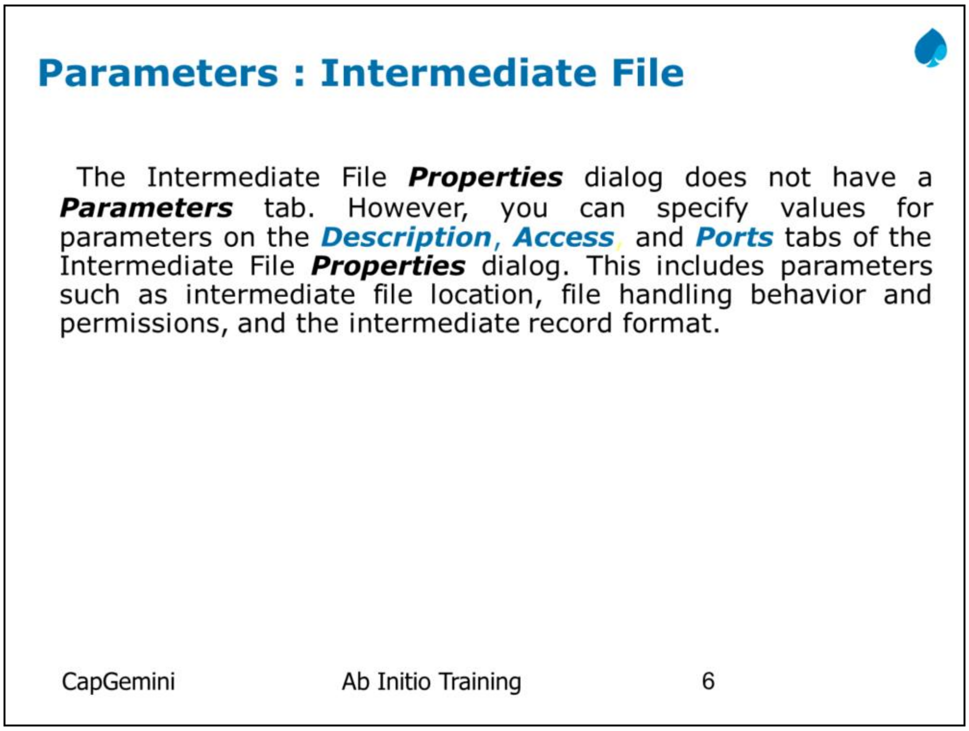
Intermediate File

- Intermediate File represents one or multiple serial files or a multifile of intermediate results that a graph writes during execution, and saves for your review after execution.



Intermediate File

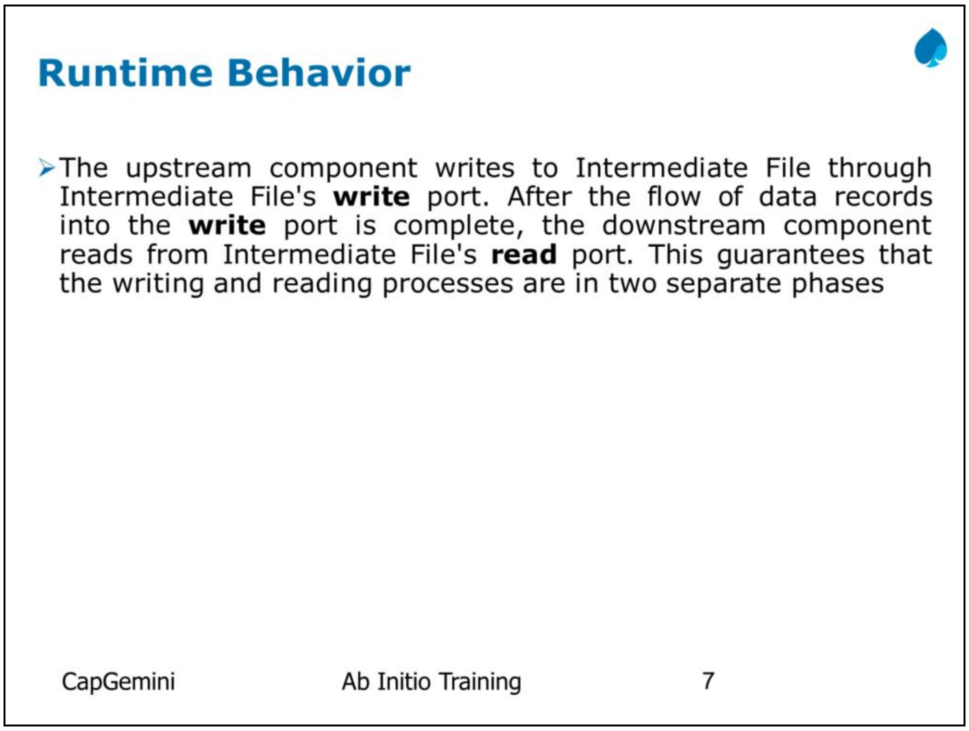
CapGemini Ab Initio Training 5



Parameters : Intermediate File

The Intermediate File ***Properties*** dialog does not have a ***Parameters*** tab. However, you can specify values for parameters on the ***Description***, ***Access***, and ***Ports*** tabs of the Intermediate File ***Properties*** dialog. This includes parameters such as intermediate file location, file handling behavior and permissions, and the intermediate record format.

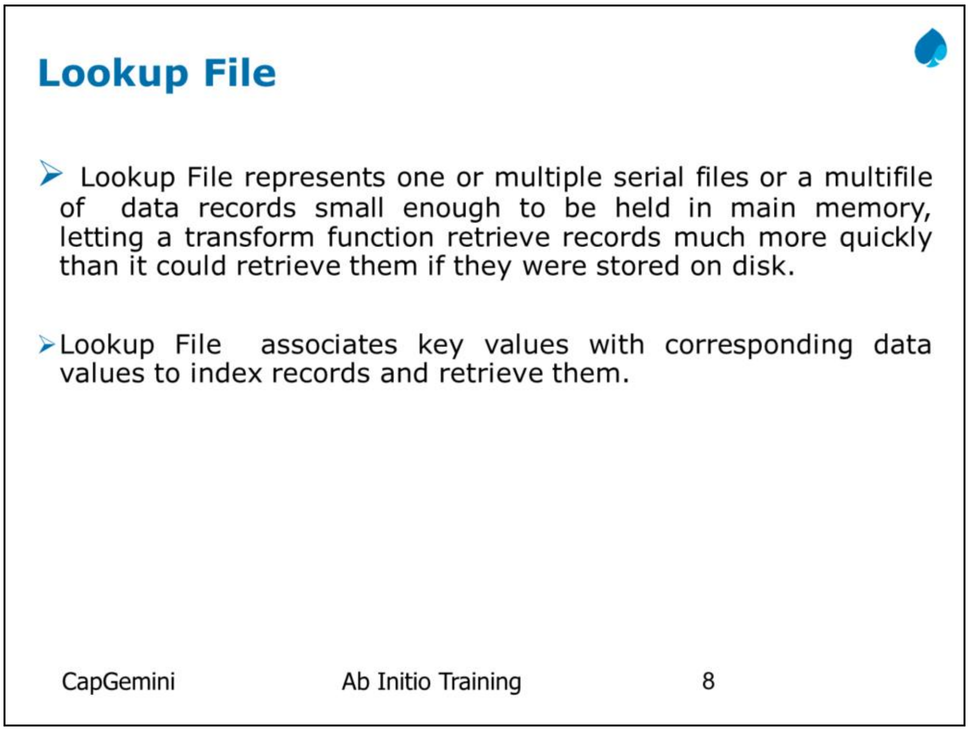
CapGemini Ab Initio Training 6



Runtime Behavior

- The upstream component writes to Intermediate File through Intermediate File's **write** port. After the flow of data records into the **write** port is complete, the downstream component reads from Intermediate File's **read** port. This guarantees that the writing and reading processes are in two separate phases

CapGemini Ab Initio Training 7

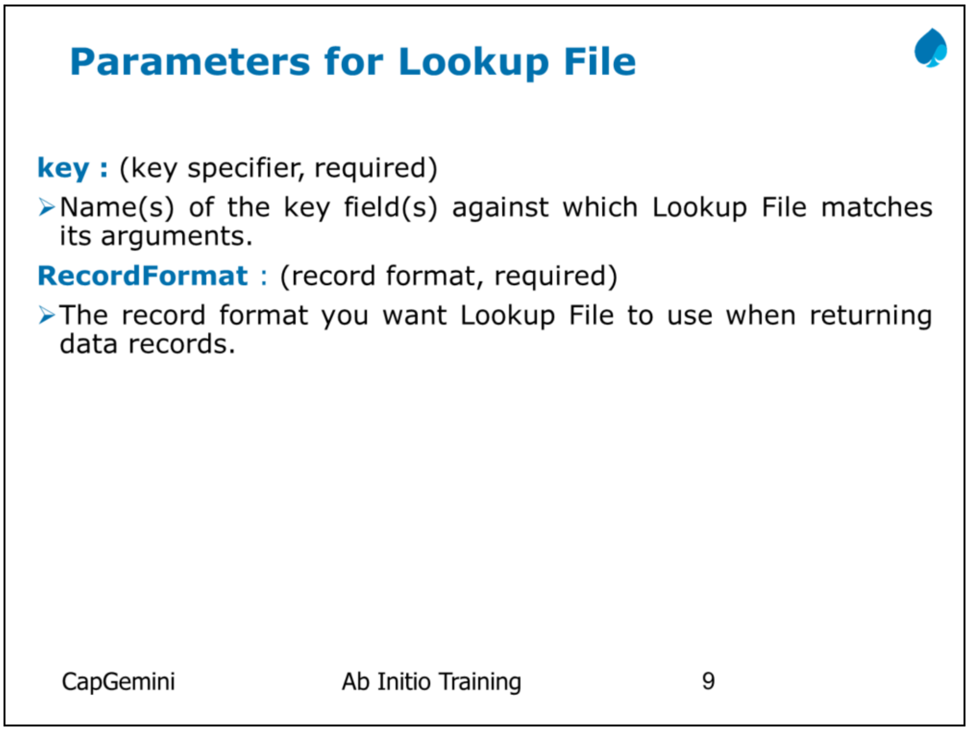


Lookup File

- Lookup File represents one or multiple serial files or a multifile of data records small enough to be held in main memory, letting a transform function retrieve records much more quickly than it could retrieve them if they were stored on disk.

- Lookup File associates key values with corresponding data values to index records and retrieve them.

CapGemini Ab Initio Training 8



Parameters for Lookup File

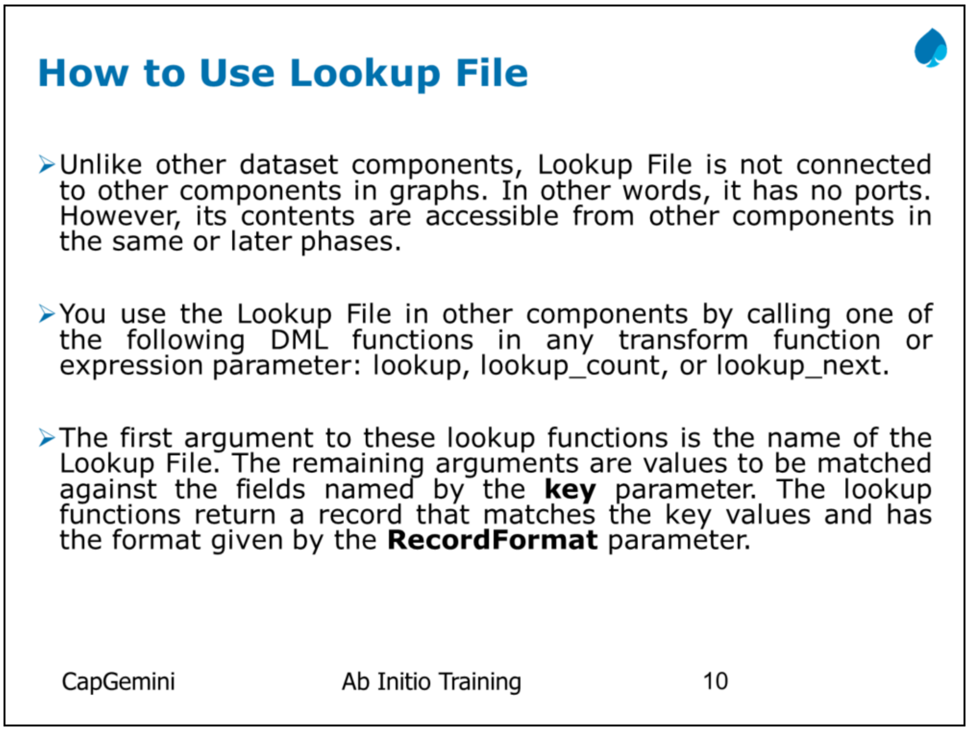
key : (key specifier, required)

- Name(s) of the key field(s) against which Lookup File matches its arguments.

RecordFormat : (record format, required)

- The record format you want Lookup File to use when returning data records.

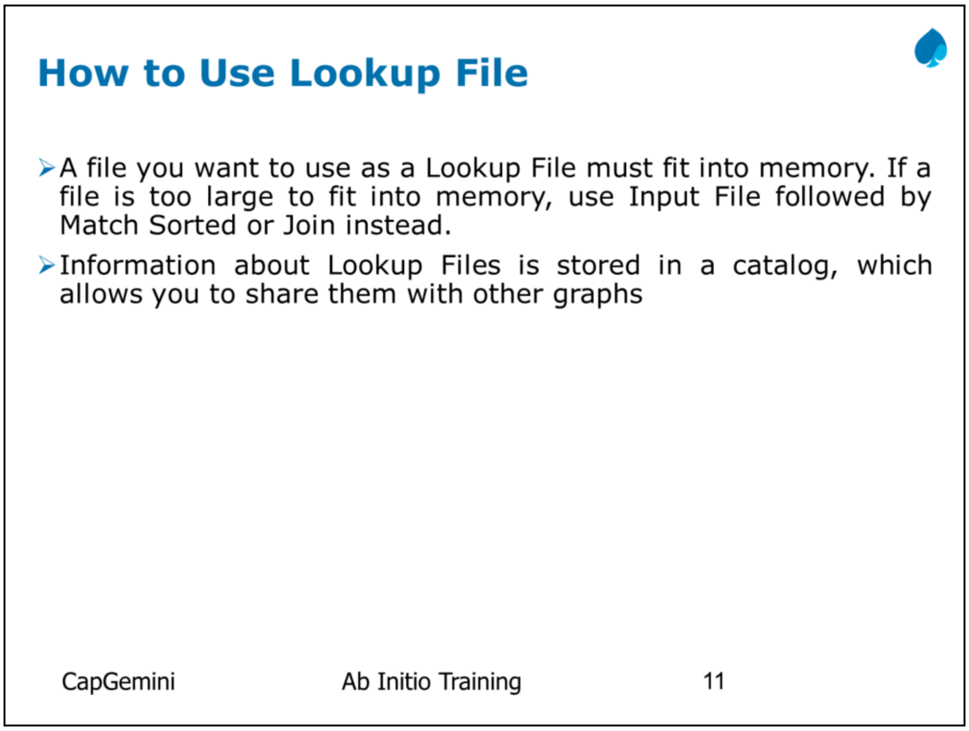
CapGemini Ab Initio Training 9



How to Use Lookup File

- Unlike other dataset components, Lookup File is not connected to other components in graphs. In other words, it has no ports. However, its contents are accessible from other components in the same or later phases.
- You use the Lookup File in other components by calling one of the following DML functions in any transform function or expression parameter: lookup, lookup_count, or lookup_next.
- The first argument to these lookup functions is the name of the Lookup File. The remaining arguments are values to be matched against the fields named by the **key** parameter. The lookup functions return a record that matches the key values and has the format given by the **RecordFormat** parameter.

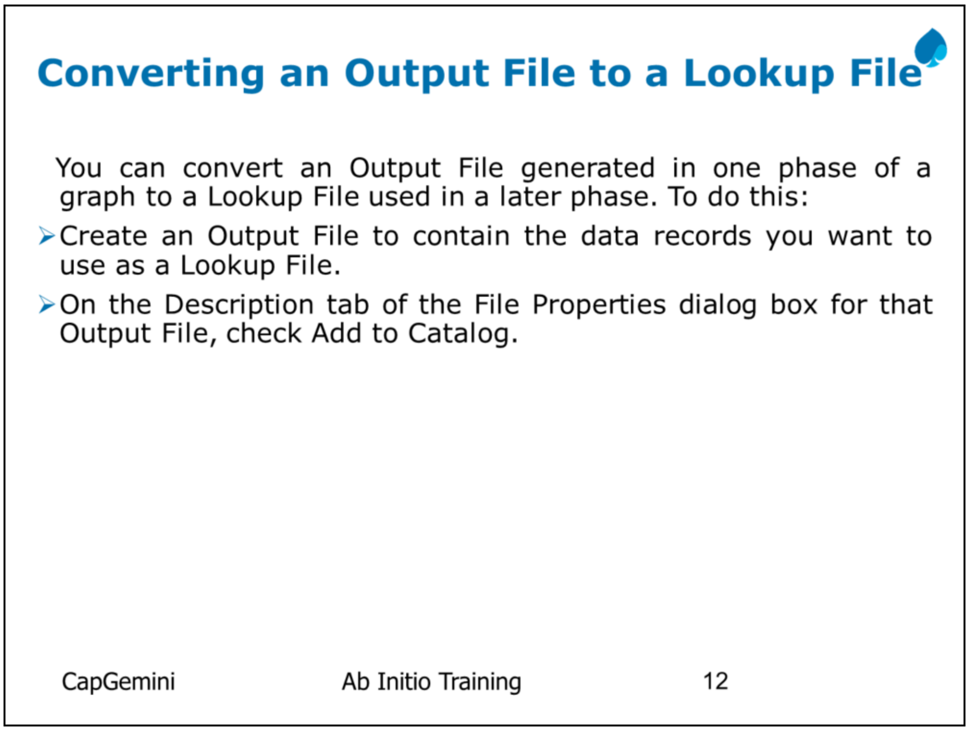
CapGemini Ab Initio Training 10



How to Use Lookup File

- A file you want to use as a Lookup File must fit into memory. If a file is too large to fit into memory, use Input File followed by Match Sorted or Join instead.
- Information about Lookup Files is stored in a catalog, which allows you to share them with other graphs

CapGemini Ab Initio Training 11

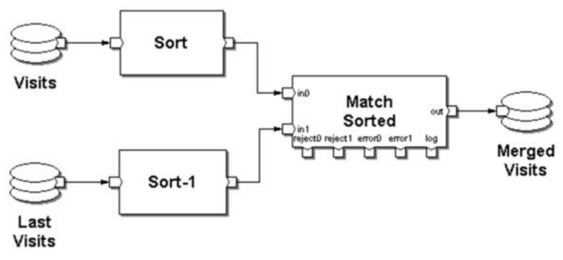


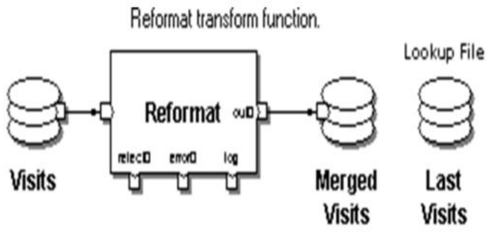
Converting an Output File to a Lookup File

You can convert an Output File generated in one phase of a graph to a Lookup File used in a later phase. To do this:

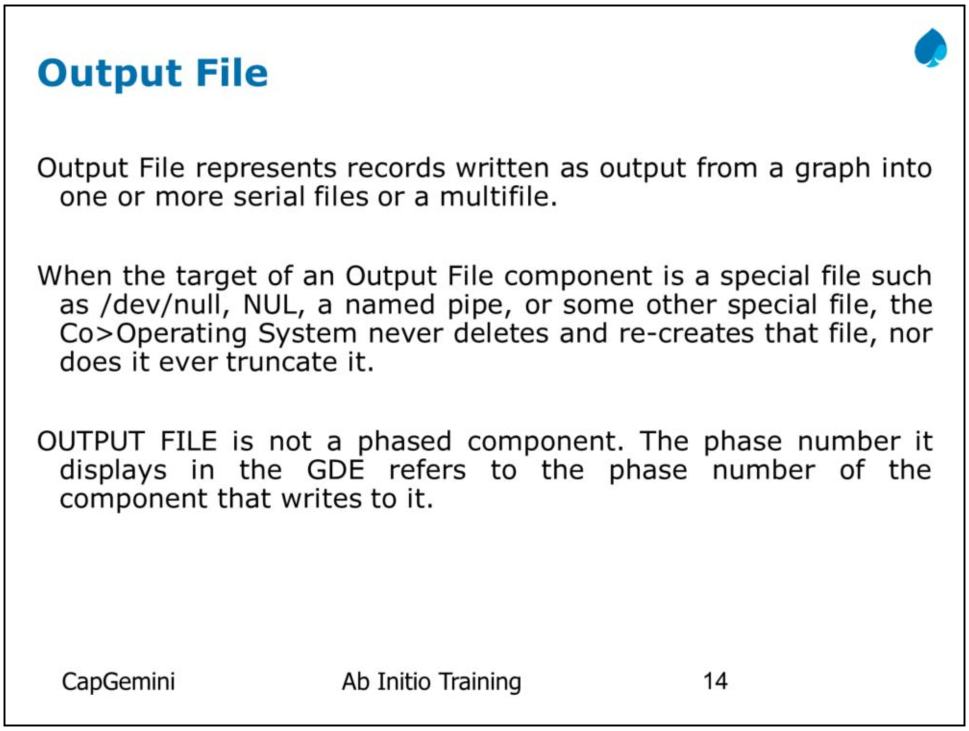
- Create an Output File to contain the data records you want to use as a Lookup File.
- On the Description tab of the File Properties dialog box for that Output File, check Add to Catalog.

CapGemini Ab Initio Training 12





CapGemini Ab Initio Training 13



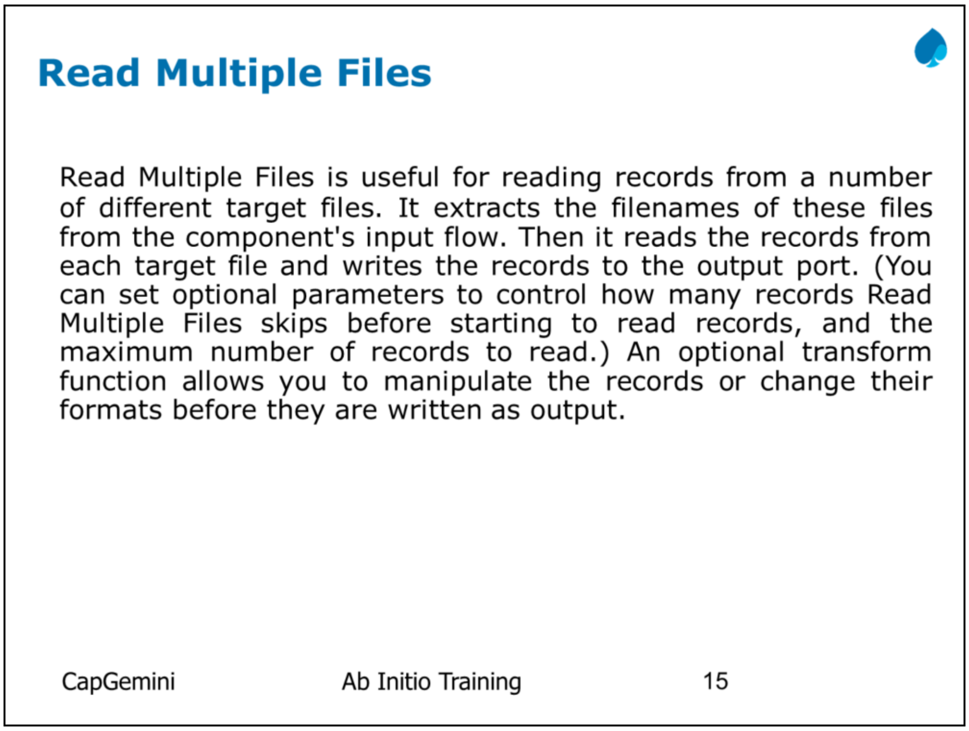
Output File

Output File represents records written as output from a graph into one or more serial files or a multifile.

When the target of an Output File component is a special file such as /dev/null, NUL, a named pipe, or some other special file, the Co>Operating System never deletes and re-creates that file, nor does it ever truncate it.

OUTPUT FILE is not a phased component. The phase number it displays in the GDE refers to the phase number of the component that writes to it.

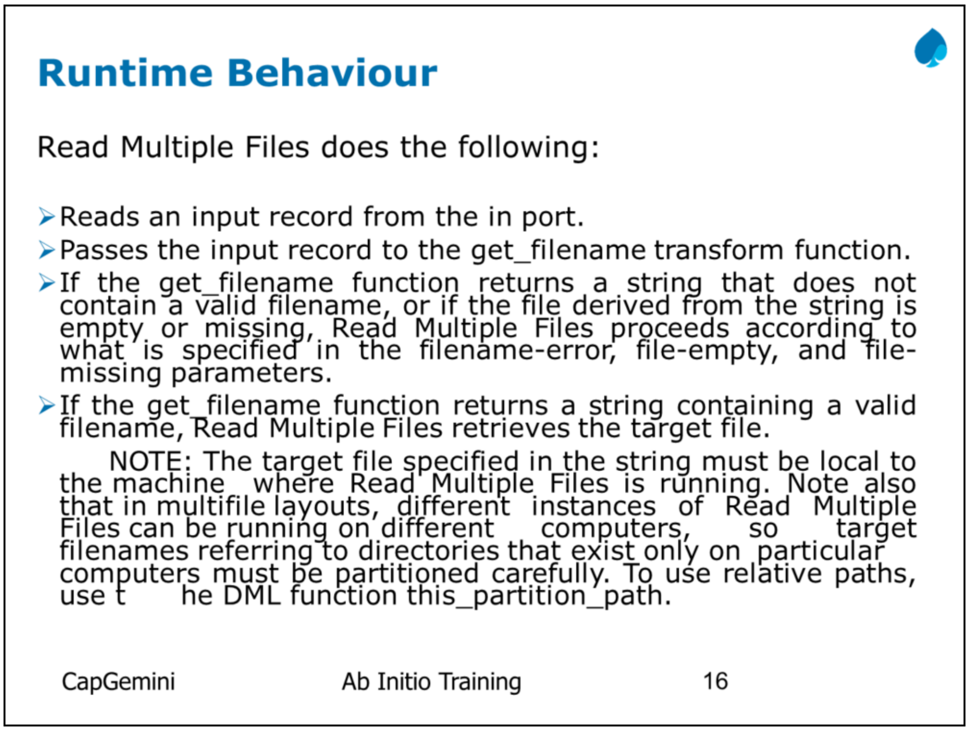
CapGemini Ab Initio Training 14



Read Multiple Files

Read Multiple Files is useful for reading records from a number of different target files. It extracts the filenames of these files from the component's input flow. Then it reads the records from each target file and writes the records to the output port. (You can set optional parameters to control how many records Read Multiple Files skips before starting to read records, and the maximum number of records to read.) An optional transform function allows you to manipulate the records or change their formats before they are written as output.

CapGemini Ab Initio Training 15



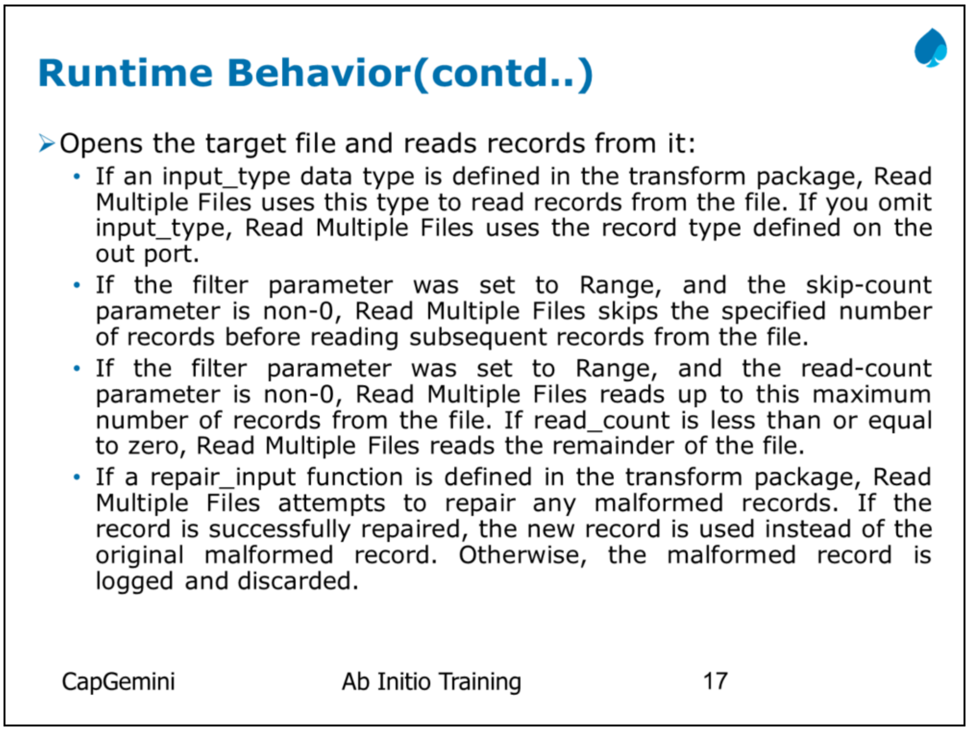
Runtime Behaviour

Read Multiple Files does the following:

- Reads an input record from the in port.
- Passes the input record to the get_filename transform function.
- If the get_filename function returns a string that does not contain a valid filename, or if the file derived from the string is empty or missing, Read Multiple Files proceeds according to what is specified in the filename-error, file-empty, and file-missing parameters.
- If the get_filename function returns a string containing a valid filename, Read Multiple Files retrieves the target file.

  NOTE: The target file specified in the string must be local to the machine where Read Multiple Files is running. Note also that in multifile layouts, different instances of Read Multiple Files can be running on different computers, so target filenames referring to directories that exist only on particular computers must be partitioned carefully. To use relative paths, use t he DML function this_partition_path.

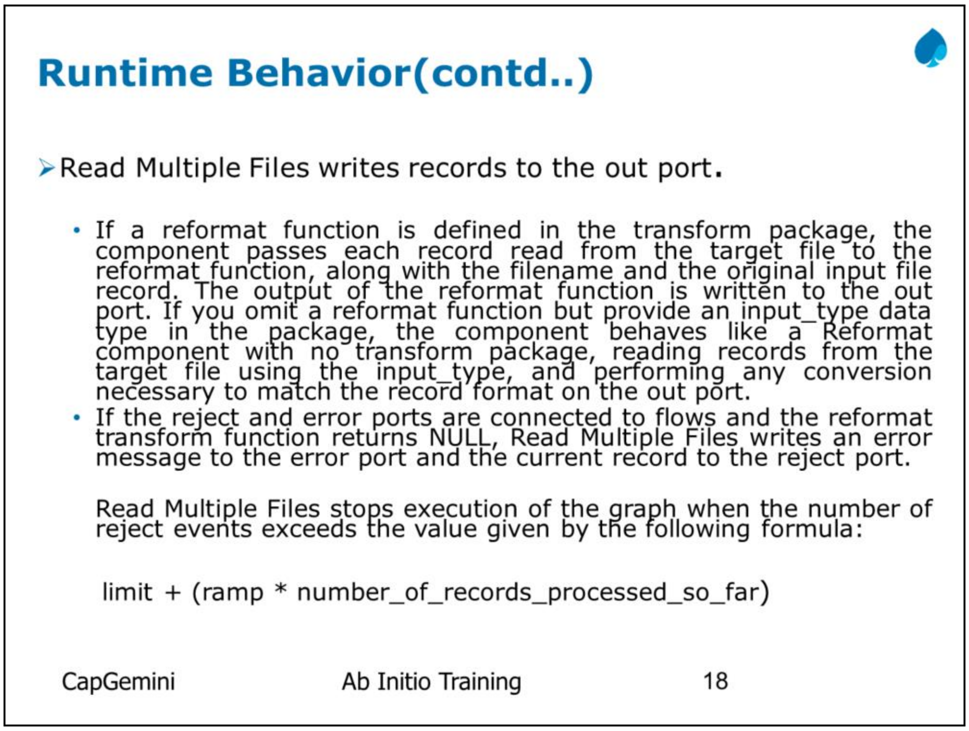
CapGemini Ab Initio Training 16



Runtime Behavior(contd..)

- Opens the target file and reads records from it:
  - If an input_type data type is defined in the transform package, Read Multiple Files uses this type to read records from the file. If you omit input_type, Read Multiple Files uses the record type defined on the out port.
  - If the filter parameter was set to Range, and the skip-count parameter is non-0, Read Multiple Files skips the specified number of records before reading subsequent records from the file.
  - If the filter parameter was set to Range, and the read-count parameter is non-0, Read Multiple Files reads up to this maximum number of records from the file. If read_count is less than or equal to zero, Read Multiple Files reads the remainder of the file.
  - If a repair_input function is defined in the transform package, Read Multiple Files attempts to repair any malformed records. If the record is successfully repaired, the new record is used instead of the original malformed record. Otherwise, the malformed record is logged and discarded.

CapGemini Ab Initio Training 17



Runtime Behavior(contd..)

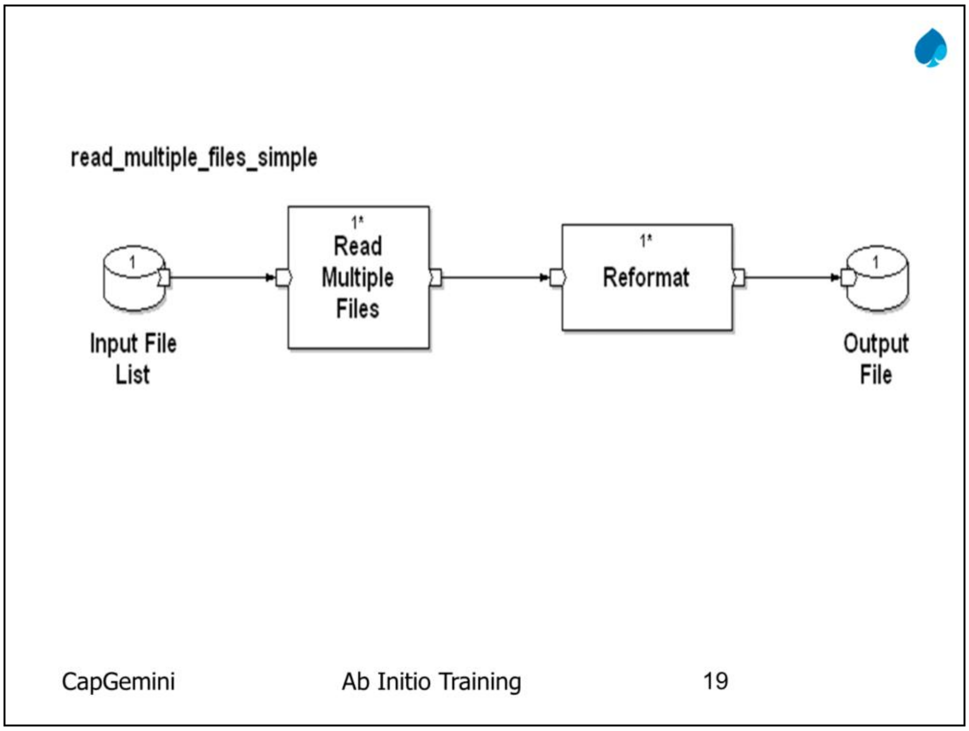
➢ Read Multiple Files writes records to the out port.

- If a reformat function is defined in the transform package, the component passes each record read from the target file to the reformat function, along with the filename and the original input file record. The output of the reformat function is written to the out port. If you omit a reformat function but provide an input_type data type in the package, the component behaves like a Reformat component with no transform package, reading records from the target file using the input_type, and performing any conversion necessary to match the record format on the out port.
- If the reject and error ports are connected to flows and the reformat transform function returns NULL, Read Multiple Files writes an error message to the error port and the current record to the reject port.

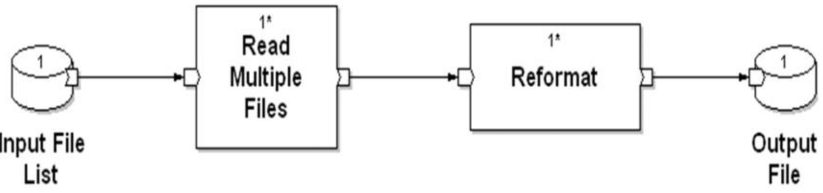
  Read Multiple Files stops execution of the graph when the number of reject events exceeds the value given by the following formula:

  limit + (ramp * number_of_records_processed_so_far)

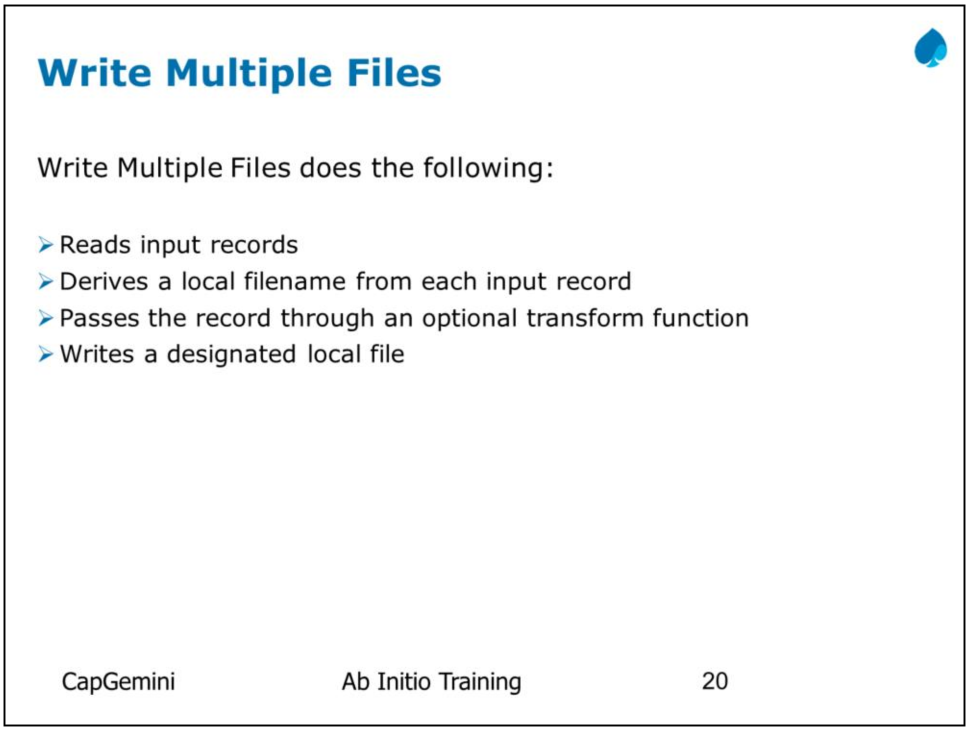
CapGemini Ab Initio Training 18



read_multiple_files_simple



CapGemini Ab Initio Training 19

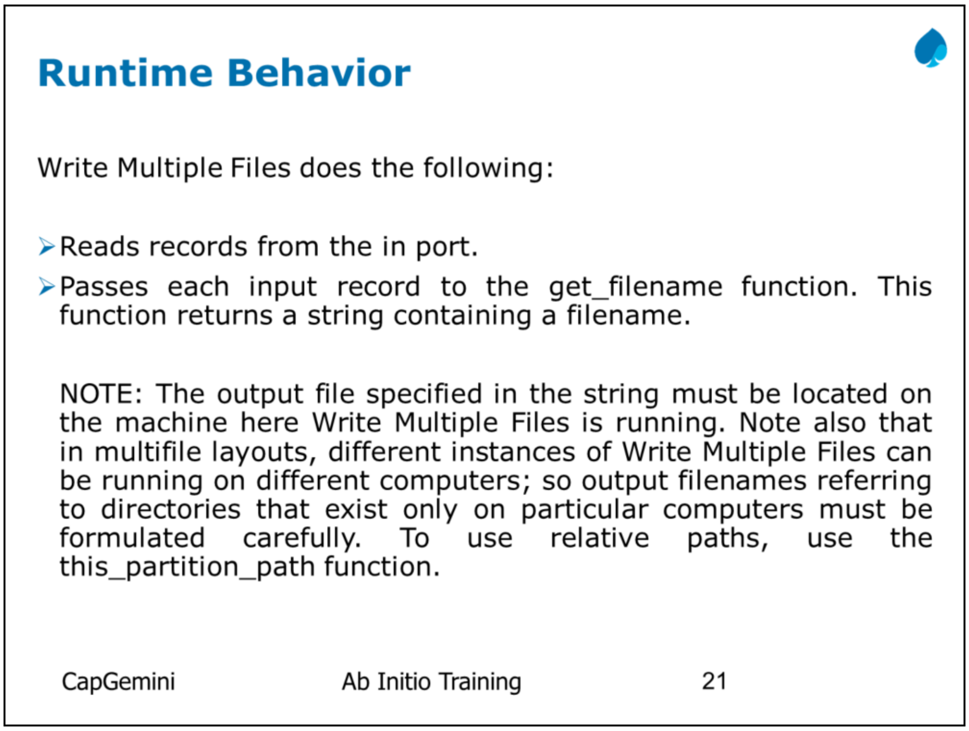


Write Multiple Files

Write Multiple Files does the following:

- Reads input records
- Derives a local filename from each input record
- Passes the record through an optional transform function
- Writes a designated local file

CapGemini Ab Initio Training 20



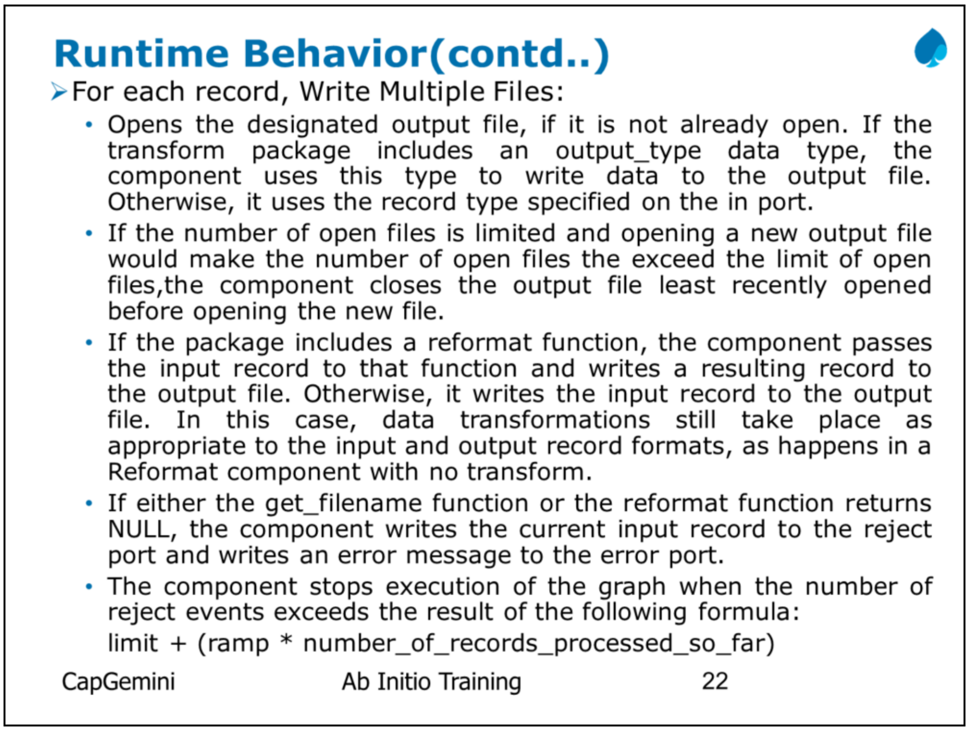
Runtime Behavior

Write Multiple Files does the following:

- Reads records from the in port.
- Passes each input record to the get_filename function. This function returns a string containing a filename.

NOTE: The output file specified in the string must be located on the machine here Write Multiple Files is running. Note also that in multifile layouts, different instances of Write Multiple Files can be running on different computers; so output filenames referring to directories that exist only on particular computers must be formulated carefully. To use relative paths, use the this_partition_path function.

CapGemini Ab Initio Training 21



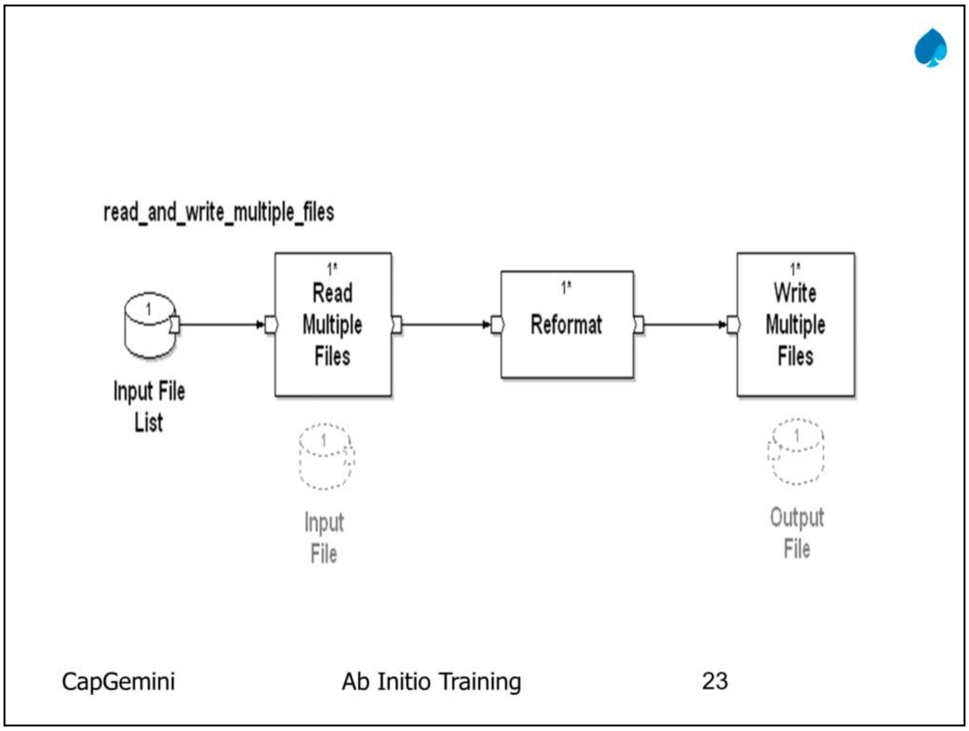
Runtime Behavior(contd..)

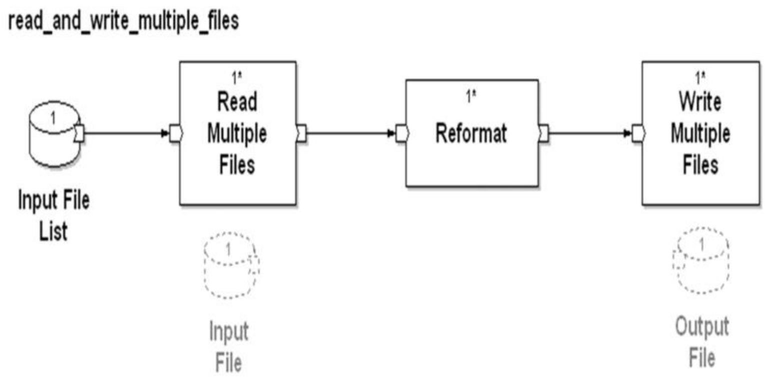
➢ For each record, Write Multiple Files:

- Opens the designated output file, if it is not already open. If the transform package includes an output_type data type, the component uses this type to write data to the output file. Otherwise, it uses the record type specified on the in port.
- If the number of open files is limited and opening a new output file would make the number of open files the exceed the limit of open files,the component closes the output file least recently opened before opening the new file.
- If the package includes a reformat function, the component passes the input record to that function and writes a resulting record to the output file. Otherwise, it writes the input record to the output file. In this case, data transformations still take place as appropriate to the input and output record formats, as happens in a Reformat component with no transform.
- If either the get_filename function or the reformat function returns NULL, the component writes the current input record to the reject port and writes an error message to the error port.
- The component stops execution of the graph when the number of reject events exceeds the result of the following formula:

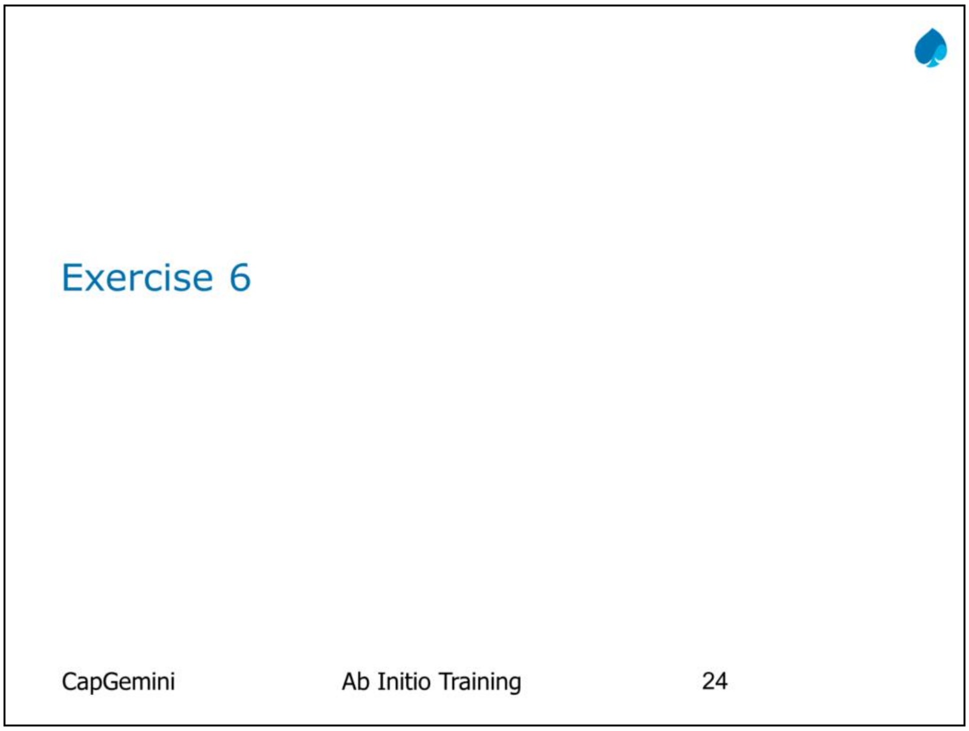
  limit + (ramp * number_of_records_processed_so_far)

CapGemini Ab Initio Training 22



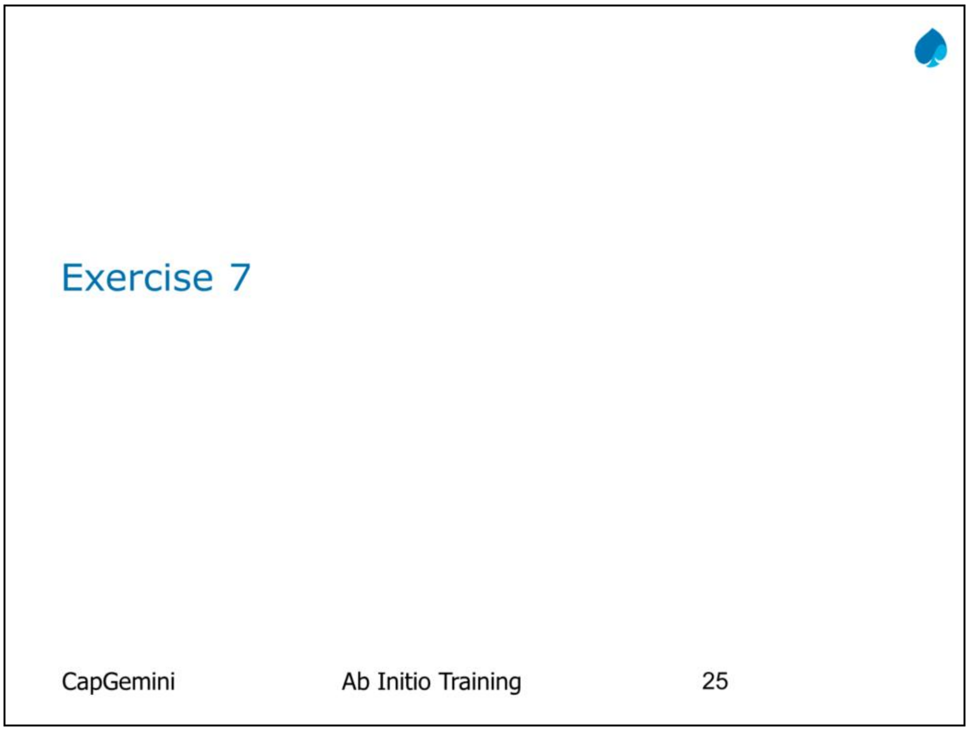


CapGemini Ab Initio Training 23



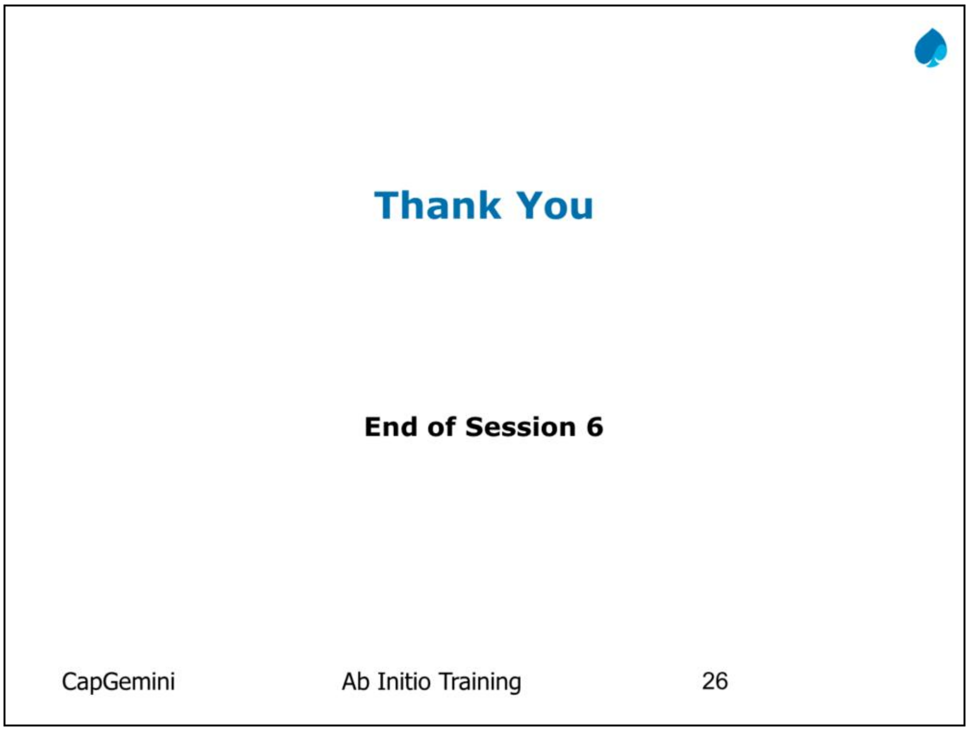
Exercise 6

CapGemini Ab Initio Training 24



Exercise 7

CapGemini Ab Initio Training 25



Thank You

End of Session 6

CapGemini Ab Initio Training 26