MPISum.java > MPISum > main(String[])

```java
import mpi.MPI;

import java.util.Scanner;

import mpi.*;

public class MPISum {
    Run | Debug
    public static void main(String[] args) throws Exception {
        MPI.Init(args);

        int rank = MPI.COMM_WORLD.Rank();
        int size = MPI.COMM_WORLD.Size();

        int unitsize = 5;
        int root = 0;
        int send_buffer[] = null;
        // 1 process is expected to handle 4 elements
        send_buffer = new int[unitsize * size];
        int recieve_buffer[] = new int[unitsize];
        int new_recieve_buffer[] = new int[size];

        // Set data for distribution
        if (rank == root) {
            int total_elements = unitsize * size;
            System.out.println("Enter " + total_elements + " elements");
            for (int i = 0; i < total_elements; i++) {
                System.out.println("Element " + i + "\t = " + i);
                send_buffer[i] = i;
            }
        }

        // Scatter data to processes
        MPI.COMM_WORLD.Scatter(
                send_buffer,
                0,
                unitsize,
                MPI.INT,
                recieve_buffer,
                0,
                unitsize,
                MPI.INT,
                root);

        // Calculate sum at non root processes
        // Store result in first index of array
        for (int i = 1; i < unitsize; i++) {
            recieve_buffer[0] += recieve_buffer[i];
        }
        System.out.println(
                "Intermediate sum at process " + rank + " is " + recieve_buffer[0]);

        // Gather data from processes
        MPI.COMM_WORLD.Gather(
                recieve_buffer,
                0,
                1,
                MPI.INT,
                new_recieve_buffer,
                0,
                1,
                MPI.INT,
                root);

        // Aggregate output from all non root processes
        if (rank == root) {
            int total_sum = 0;
            for (int i = 0; i < size; i++) {
                total_sum += new_recieve_buffer[i];
            }
            System.out.println("Final sum : " + total_sum);
        }
        MPI.Finalize();
    }
}
```