

Todo list

úvod k čemu software je a co umí	3
screenshot okna, číselný popis prvků	3
video / obrázky / config / binary	3
jak se ovládá, jak se chová	3
odkaz na sekci s konfigurací úlohy	3
nic extra, pouze poznámka, že nenastavené parametry se nastaví na výchozí (uživatel se nemusí starat)	3
uložení configu / bin. otisku	3
zobrazení výsledků a případné uložení	3
detailní popis možných položek v konfiguračním souboru	3
velikost facetu, parametr výpočtu prodloužení, velikost okna pro lokální dohledávání atd.	4
spravne zdrojove kody	5





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Manuál k softwaru DIC-Voucher

Ing. Petr Ječmen

Fakulta mechatroniky, informatiky a mezioborových studií
Technická Univerzita v Liberci
Studentská 2
461 17 Liberec



1 Úvod

úvod k čemu softare je a co umí



2 Ovládání aplikace

2.1 Úvodní okno

screenshot okna, číselný popis prvků

2.2 Výběr vstupu

video / obrázky / config / binary

2.3 Zadávání ROI

jak se ovládá, jak se chová

2.4 Expertní nastavení

odkaz na sekci s konfigurací úlohy

2.5 Spuštění výpočtu

nic extra, pouze poznámka, že nenastavené parametry se nastaví na výchozí (uživatel se nemusí starat)

2.6 Uložení

uložení configu / bin. otisku

2.7 Výsledky

zobrazení výsledků a případné uložení

3 Konfigurační soubor úlohy

detailní popis možných položek v konfiguračním souboru



4 Doporučená nastavení

velikost facetu, parametr výpočtu prodloužení, velikost okna pro lokální dohledávání atd.



5 Source code examples

5.1 Instance creation and messaging

Constant declaration

```
final UUID ID = UUID.randomUUID(); // define ID for messaging
```

Client side code

```
public static void main(String[] args) {
    Client client = Client.Client.initNewClient();
    client.getListenerRegistrator().setIdListener(ID, new MessageHandler()
    );
}

class MessageHandler implements Listener<Identifiable> {
    @Override
    public Object receiveData(Identifiable data) {
        if (data instanceof Message) {
            Message m = (Message) data;
            System.out.println(m.getData()); // print content to console
            return m.getHeader(); // send valid response
        } else {
            return "ERROR"; // report error
        }
    }
}
```

5.2 Job Management

Server side code

```
public static void main(String[] args) {
    double start = -100.0, end = 100.0; // define range of values
    double step = 0.01;
    // divide the range in small pieces and create a Set<double[]>, where
    // double[] defines start and end of each sub-range and step size
    for (double[] d : subranges) { // submit all jobs
        s.getJobManager().submitJob(d);
    }
    s.getJobManager().waitForAllJobs(); // wait until all jobs are
    complete
}
```

```
// use s.getJobManager().getAllJobs() to get all submitted jobs and
    find the best result
}

class DataStore implements DataStorage {
    @Override
    public Object requestData(Object o) {
        switch (o.toString()) {
            case "data":
                return dataArray; // return data
            default:
                return "Illegal data request";
        }
    }
}
```

