

# How Things Start

This chapter covers the following topics:

- Starting an Application Designer Session
- Starting Additional Application Designer Subsessions

---

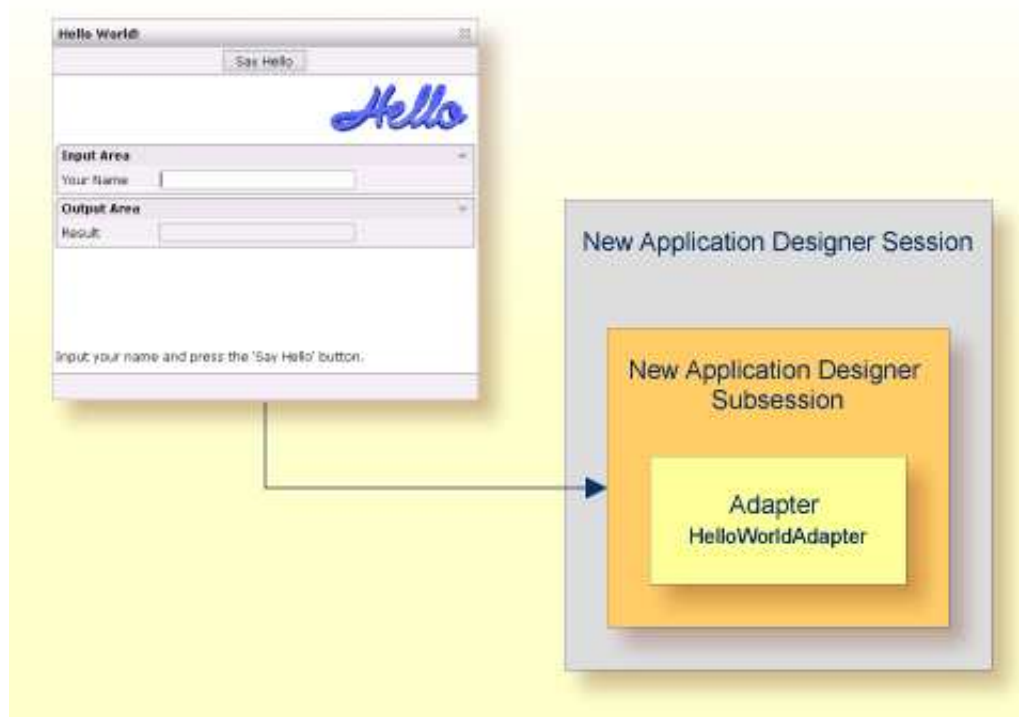
## Starting an Application Designer Session

The proper start of a session is to open an Application Designer page via the StartCISPage servlet.

Example: If you start the "Hello World!" page with the following URL, a new Application Designer session object with a new session ID is automatically created on the server side:

```
http://localhost:51000/cis/servlet/StartCISPage?PAGEURL=/cisyourfirstproject/helloworld.html
```

The logical counter part of the page - the HelloWorldAdapter object - is opened inside a subsession that is automatically created inside the Application Designer session.



You see that inside one Application Designer session, there is always at least one subsession.

# Starting Additional Application Designer Subsessions

You may use your pages in a mode in which you always work inside one Application Designer subsession - the one which was created during the `StartCISPage` procedure. But maybe you want to start additional subsessions.

There are two good reasons for starting additional subsessions:

- **Separate life cycles of activities**

A subsession is keeping all adapter instances which play a role inside the processing of a certain activity. By closing the subsession, all adapter objects belonging to the subsession are released and can be caught by the garbage collector. In other words: a subsession is something like the life cycle manager for its contained adapters.

Consequence: if you have multiple activities running in parallel, then each activity has its own life cycle, e.g. it can be closed individually without any consequence for the life cycle of the other activities.

- **Isolated activities**

The adapter objects are created per subsession. This means: you can run one and the same activity in parallel - represented by two subsessions. In both subsessions, adapters are built up in parallel - completely isolated from one another.

Programming applications inside "multi document interface"-like programs, (e.g. applications inside the Application Designer workplace) is therefore simple: each document (activity) is associated with its own subsession. The workplace just coordinates that the correct page is linked to the correct subsession at the appropriate point of time.

The starting of a new Application Designer subsession is done by opening a page inside a frame or inside an Application Designer subpage via Application Designer APIs.

Application Designer offers APIs (in class `com.softwareag.cis.server.Adapter`) to open Application Designer pages in a certain frame. These APIs always have one "simple" variant and one "complex" variant:

- **Simple Variant**

```
protected void openCISPageInTarget(String pageURL,  
                                   String target)
```

By calling this method, you open a certain page in a certain frame. The page is automatically linked to the subsession of the adapter calling this method.

- **Complex Variant:**

```
protected void openCISPageInTarget(String pageURL,  
                                   String subsessionId,  
                                   String target)
```

By calling this method you open a certain page in a certain frame. But now you can explicitly pass a new `subsessionId` to be used for the page's adapter.

The proper call for a page which should belong to a new subsession is:

```
...  
...
```

```
public void onOpenNewPage()  
{  
    // create new subsession id  
    String newSSID = UniqueIdMgmt.createPseudoGUID();  
    openCISPageInTarget("...URL...",newSSID,"...TARGET...");  
}  
...  
...
```