Dynamic Service Creator (DSC)

The Dynamic Service Creator (DSC) is a proxy service that accepts an input message to a Web service from a workflow engine (GPEL) on behalf of an actual Web service instance, finds an appropriate Web service instance, and invokes the Web service instance. DSC uses XRegistry to look for available Web services instances, and when no Web service instances are available, DSC contacts GFac to create a new Web service instance.

**The architecture of DSC**

Figure 1 shows the flow of execution involving DSC. During a workflow execution, first, a workflow engine (GPEL) invokes a Web service using a WSDL document given along with the workflow script. This WSDL document has the address of DSC as the location of Web service. Thus, the workflow engine sends the input message sends the input message to DSC, believing that DSC is an actual Web service (step 1 in the figure). An important point here is that the local part of the address of DSC contains the QName of the portType of the Web service, enabling DSC to know what Web service the workflow engine believes that DSC is.

When DSC receives a message, DSC binds a Web service interface identified by the QName of the portType to a Web service instance (step 2 in the figure). DSC does this in the following steps. First, DSC looks up the service registry (XRegistry) to check if there are any Web service instances running (step 2.1 in the figure). If there are some Web service instances available, DSC pings the Web service instances to check if they are really alive. If there is no Web service instance available or alive, DSC contacts a service factory (GFac) and creates a new Web service instance (step 2.2 in the figure). After creating the Web service, the service factory registers the created Web service instance to the service registry so that DSC can reuse this Web service instance in the future (step 2.3 in the figure). At this point, DSC knows at least one available Web service instance, either by selecting it from the service registry or by creating a new one using the service factory. Then, finally, DSC forwards the input message received from the workflow engine to the Web service instance (step 3 in the figure).

When the Web service instance finishes the execution, the output message returns directory to the workflow engine (step 4 in the figure). This is done by setting the ReplyTo address specified in the WS-Addressing specification in the SOAP header. This is important to make DSC stateless, and thus scalable.
The API of DSC

Since DSC works as a proxy for any Web service, it does not have its own WSDL. Clients, such as a workflow engine, uses the WSDL document of actual Web service, but the location of the Web service should be the address of DSC.

The local part of the address of DSC contains the QName of the portType of a Web service. The following describes the steps to embed the QName in the DSC address.

   e.g. {http://www.extreme.indiana.edu/lead}Echo
2. Encode the string representation to application/x-www-form-urlencoded using the UTF-8 character encoding.
   e.g. %7Bhttp%3A%2F%2Fwww.extreme.indiana.edu%2Flead%7DEcho
3. Append the encoded string to the address of DSC.
   e.g.

DSC also requires LEAD context header to be set in the SOAP message. The required fields are described below:

- **xregistry-url**
  The URL of XRegistry that DSC uses to search for available Web service instances.
- **gfac-url**
  The URL of GFac that DSC uses to create a Web service instance.