

Using BLOBs

There may be cases where an application requires more control over the Web service request or response payload. For example, the Web service requires a message structure that can't be created or parsed by the native LightWave Server JSON support, or the message payload is larger than the maximum message size allowed by an interprocess message. For these cases LightWave Server supports the BLOB data type.

When using BLOBs (Binary Large Objects), your application provides or receives the Web service payload as raw data in the IPM or in an Enscribe file. When a file is used, the name of the file is supplied in the request and/or response IPMs.

Using IPM BLOBs

When an API definition specifies a request or reply field type as *BLOB Field*, the data transferred to or from the field in the IPM represents exactly the data that is to be sent or that was received. The *sizeIs* attribute may be used to reference a field that contains the actual size of the data to send or the size of the data that was received. The maximum size of an IPM blob is 2MB, minus the size of all other fields in the IPM.

Message Field Mapping			+
Field Name *	Source Type	Required?	
<input type="text" value="blob"/>	HTTP Request Body to BLOB Field ▾	<input type="checkbox"/>	

```
<type name="RqBlob">
  <element name="rqCode" type="short" offset="0" size="2"/>
  <element name="blobSize" type="short" offset="2" size="2" hide="1"/>
  <element name="blob" type="string" offset="4" size="1000000" sizeIs="blobSize"/>
</type>
```

Using File BLOBs

File BLOBs are similar to IPM BLOBs but Enscribe files are used to transfer the data. When an API definition specifies the request or reply field type as *BLOB File*, the source or destination field contains the name of the Enscribe file containing the BLOB data. For web service requests, LightWave Server creates the BLOB file and supplies the file name in the blobFileName field in the request IPM to the application server. For web service responses, the application server supplies the BLOB file name in the reply IPM. The LightWave Server SERVER / SWORKER processes must have read access to the file.

Message Field Mapping			+
Field Name *	Source Type	Required?	
<input type="text" value="blobFileName"/>	HTTP Request Body to BLOB File ▾	<input type="checkbox"/>	

```
<type name="RqBlobFile">
  <element name="rqCode" type="short" offset="0" size="2"/>
  <element name="blobFileName" type="string" offset="2" size="256"/>
</type>
```

Request BLOB files are created by the LightWave Server processes as type 101 EDIT files, when the request content type is "text/plain". For all other content types, type 180 files (binary stream) are created. The files are created using the location and attributes specified in the SERVER process *--blob-files* startup option. See [SERVER](#) for more information on the *--blob-files* startup option. The location and attributes may also be specified in the API definition. Values specified in the API definition are optional, and when specified will override the SERVER process values.

 Specifying the BLOB file location and attributes in the API definition can be problematic if the API definition will be moved to different environments on different NonStop systems. When specifying the attributes in the API definition, consider using DEFINES with unique application-specific names to specify the BLOB file location. The DEFINE values can be set appropriately in each environment when starting the SERVER process. Use SAFEGUARD to configure appropriate subvolume access permissions to allow access to the BLOB files by users and applications.

Note that the application is responsible for the disposal of request BLOB files. Response BLOB files may be automatically purged by LightWave Server by selecting the *Purge BLOB file on request completion* option in the API reply definition. When this option is selected, the BLOB file is deleted on completion of the request, regardless of the outcome.