

# Composing IDocs

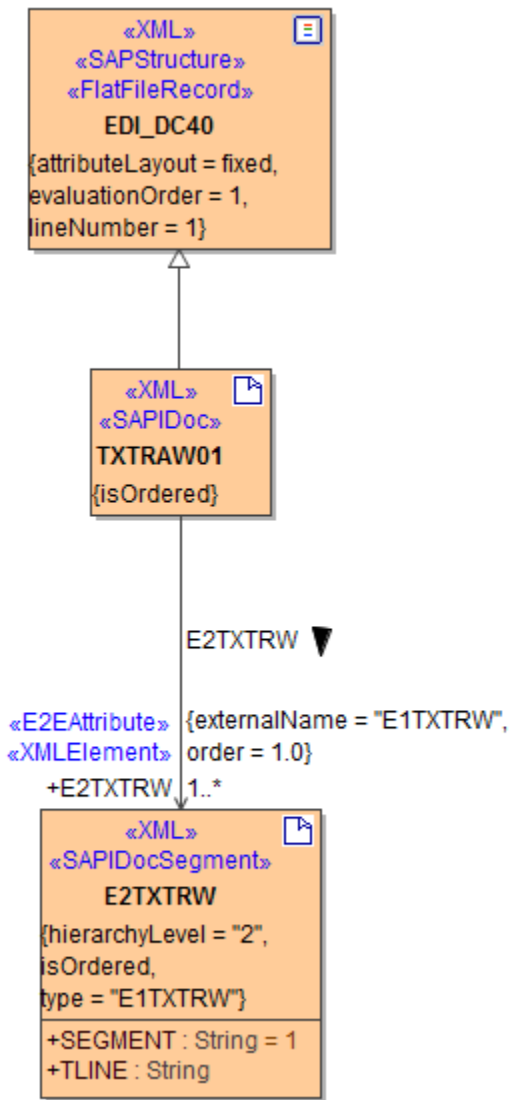
IDocs (Interface Documents) are modeled as UML classes (cf. **TXTRAW01**). Such classes can be serialized into SAP IDoc tables (arrays) by the `<<SAPIDocRecordComposer>>`.

There are two situations in which you may want to compose an IDoc data buffer:

- You want to map an IDoc object (e.g. **TXTRAW01**, see figure below) to a **String**, keeping the structure of the data (mostly done, if you want to write the IDoc to a file). This is done by an action having the stereotype `<<SAPIDocComposer>>` (see [Composing to String](#)).
- You want to map an IDoc object (e.g. **TXTRAW01**, see figure below) to a **SAP tables** structure (mostly done, if you want to perform an SAP RFC). This is done by an action having the stereotype `<<SAPIDocRecordComposer>>` (see [Composing to SAP Tables](#)).

After having composed an IDoc object, you can send it to an SAP backend using IDOC\_INBOUND\_ASYNCHRONOUS. The necessary interface is already part of the base components. Refer to [RFC Service](#) for more information on how to implement the call.

Figure: UML Class Model of IDoc TXTRAW01



## On this Page:

- [Composing to String: <<SAPIDocComposer>>](#)
  - [Composing a Single IDoc to an IDoc String](#)
  - [Composing Multiple IDocs to an IDoc String](#)
- [Composing to SAP Tables: <<SAPIDocRecordComposer>>](#)
  - [Composing a Single IDoc to SAP Tables](#)
  - [Composing Multiple IDocs to SAP Tables](#)
- [Composing to XML: <<SAPXMLIDocComposer>>](#)
  - [Composing Multiple IDocs to an XML Blob](#)
- [IDoc Composition and SAP R/3](#)

## Related Pages:

- [Parsing IDocs](#)
- [tRFC Service](#)
- [SAP Adapter Reference](#)

#### Example File (Builder project Add-ons/SAP):



<your example path>\Add-ons\SAP\uml\sapIdoc.xml

## Composing to String: <<SAPIDocComposer>>

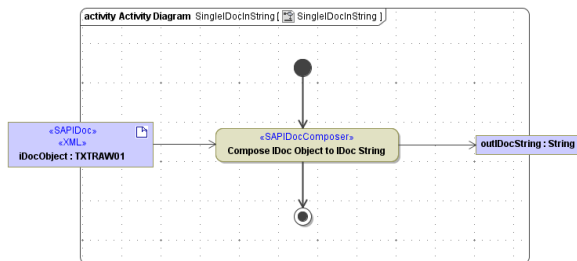
The <<SAPIDocComposer> takes one or more IDoc objects as input and composes a **String** containing the object data. The structure of the data is kept (padding).

Name	Type	Direction	Description
anyObjectFlow	Any	in	Parsed IDoc object(s) The class specifying the type of this parameter must have stereotype <<SAPIDoc>>.
idocString	String	out	String containing IDoc data (e.g. the content of an IDoc file)

The following examples are demonstrating some applications of the <<SAPIDocComposer>>.

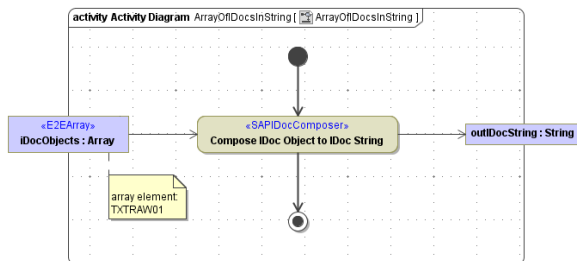
### Composing a Single IDoc to an IDoc String

Figure: SAPIDocParser, Single IDoc to a String



### Composing Multiple IDocs to an IDoc String

Figure: SAPIDocComposer, Multiple IDocs to a String



## Composing to SAP Tables: <<SAPIDocRecordComposer>>

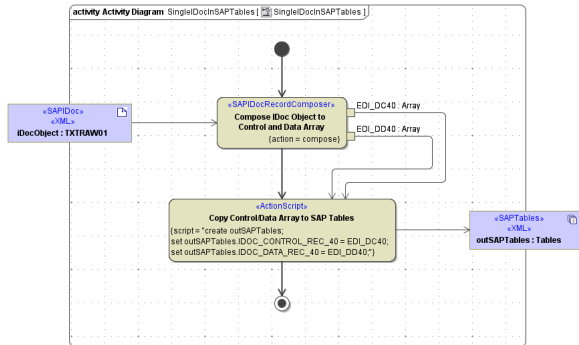
The <<SAPIDocRecordComposer> takes one or more IDoc objects as input and composes an array of structured control records and an array of structured data records (as expected for SAP tables).

Name	Type	Direction	Description
------	------	-----------	-------------

anyObjectFlow	Any	in	Parsed IDoc object(s) The class specifying the type of this parameter must have stereotype <<SAPIDoc>>.
EDI_DC40	Array	out	Array of IDoc structured control records
EDI_DD40	Array	out	Array of IDoc structured data records

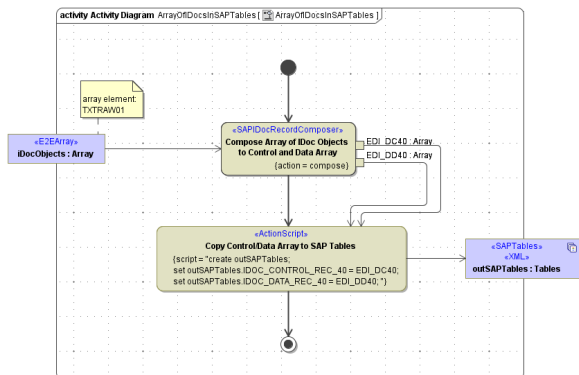
## Composing a Single IDoc to SAP Tables

Figure: IDoc Composer with a IDoc-array to Tables



## Composing Multiple IDocs to SAP Tables

Figure: IDoc Composer with a IDoc-array to string



## Composing to XML: <<SAPXMLIDocComposer>>

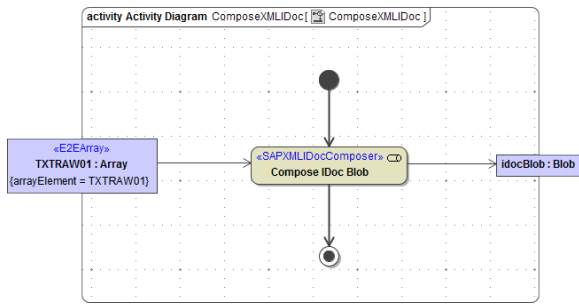
The <<SAPXMLIDocComposer>> takes one or more IDoc objects as input and composes a **Blob** containing the object data. The structure of the data is kept (padding).

Name	Type	Direction	Description
anyObjectFlow	Any	in	Parsed IDoc object(s) The class specifying the type of this parameter must have stereotype <<SAPIDoc>> and <<XML>>.
idocBlob	Blob	out	Blob containing IDoc data (e.g. the content of an IDoc file)

The following example is demonstrating an application of the <<SAPXMLIDocComposer>>.

## Composing Multiple IDocs to an XML Blob

Figure: SAPIDocParser, Multiple IDocs to XML



## IDoc Composition and SAP R/3

The above examples show how to compose IDocs version of 4.x. However, SAP R/3 3.x systems use IDocs, too. These IDocs are handled similar as in version 4.x. The differences between composing version 4 and version 3 IDocs are as follows :

- IDoc3 Composer Stereotype: `<<SAPIDoc3Composer>>`
- IDoc3 Stereotype: `<<SAPIDoc3>>`, this is taken into account if the import was done in version 3 mode (see [Importing SAP IDoc Meta Data](#)).