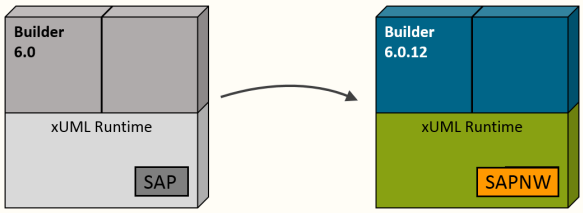
Adapting the Services

The E2E Builder – the tool for developing xUML services – comes with an embedded xUML Runtime. As of Builder 6.0.12 this Runtime contains the SAP NetWeaver firmware package. This means, that you cannot test services on the embedded Runtime anymore, that have been designed for the old SAP runtime package.



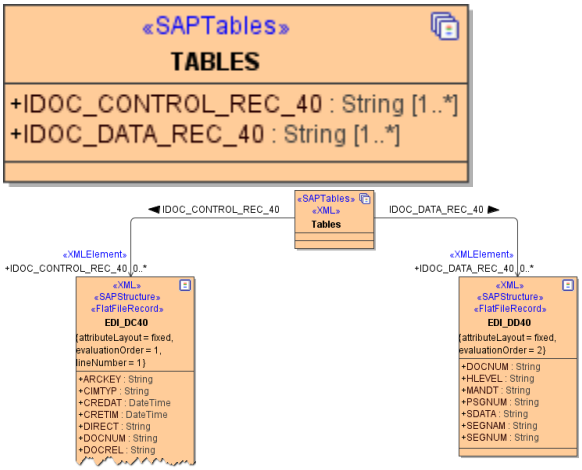
Changes in Data Handling

Changes from the old SAP runtime package to SAP NetWeaver mostly concern data handling.

SAP Tables Parameter

Formerly, the SAP library expected the SAP parameters to be given as ordered strings. Now, the order of SAP parameters no longer matters but names became significant: SAP parameters are given as ordered structures.

Figure: Old And New SAP Table Parameter



As SAP requires a structure now, you must use structures in your service model.

Using structured parameters has two advantages:

- You now can nest no-trivial structures to any level.
- You can use incomplete structure definition. That is, if you know that you need only 5 fields from a 30-field structure, go ahead and define only those 5. The rest will be omitted.
  Remember however, that this may not work for RFC Server. For RFC Client, however, it's perfectly safe to do so.

Type Conversion

The SAP libraries will not throw a conversion error anymore, if `CHAR` data is too long, but the data will be truncated.

Activity Diagrams

On this Page:

Related Pages:

- SAP Adapter
- SAP Service
- Service Migration Checklist

As discussed in Changes in Data Handling before, SAP parameters have changed. In addition to the old SAP parameter **TABLES**, the Builder now provides a new parameter **Tables** in the Add Ons section of the Base Components and an IDoc operation using the new parameter format.

Old Adapter		New Adapter	
Element	**Description**	**Element**	**Description**
TABLES	unstructured class for SAP Tables parameter	**Tables**	structured class for SAP Tables parameter
		EDI_DC40	class representing the structure of an IDoc control record
		EDI_DD40	class representing the structure of an IDoc data record
RFCPort_IDOC_INBOUND_ASYNCHRONOUS	IDoc RFC operation using **TABLES**	**RFCPort_IDOC_INBOUND_ASYNCHRONOUS_NEW**	IDoc RFC operation using **Tables**

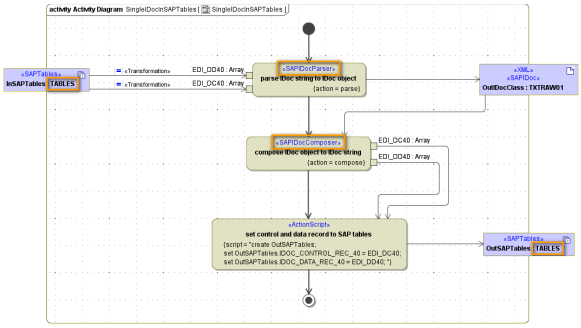
All <<SAPRFCAdapters>> and <<SAPTRFCAdapters>> that are expecting a **tables** parameter must now be provided with a **tables** parameter of type **Tables** instead of **TABLES**.

Before calling an adapter operation that is expecting an IDoc, you can convert the IDoc into a **Tables** parameter using the new <<SAPIDocRecordComposer>> action.
After having called an adapter that comes back with an IDoc, you can convert the **Tables** parameter into an IDoc using the new <<SAPIDocRecordParser>> action.

In services using the **RFCPort_IDOC_INBOUND_ASYNCHRONOUS** operation, this one has to be replaced by **RFCPort_IDOC_INBOUND_ASYNCHRONOUS_NEW**.

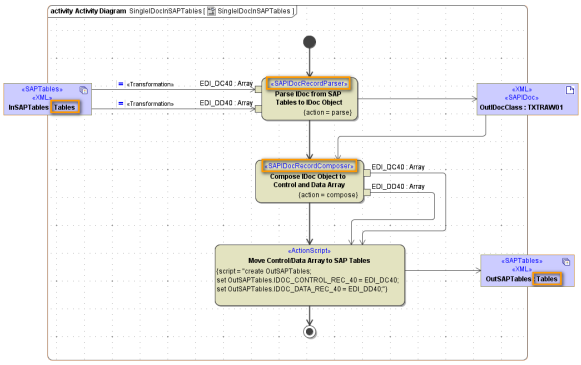
Example: IDocs in SAP Tables

Figure: OLD Implementation of Single IDoc in SAP Tables



The old <<SAPIDocParser>> and <<SAPIDocComposer>> have to be used, because the IDoc control and data records come as unstructured objects of type **String**.

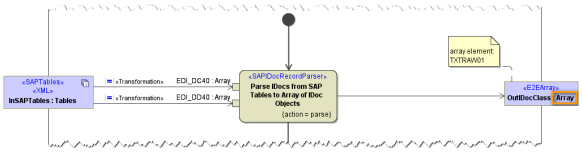
Figure: NEW Implementation of Single IDoc in SAP Tables



Now, the **InSAPTables** come as structured types, so the <<SAPIDocRecordParser>> and <<SAPIDocRecordComposer>> have to be used.

If object **OutDocClass** was to be an array, the <<SAPIDocRecordParser>> would directly parse multiple IDocs from the input tables to an array of parsed IDocs.

Figure: NEW Implementation of Multiple IDoc in SAP Tables



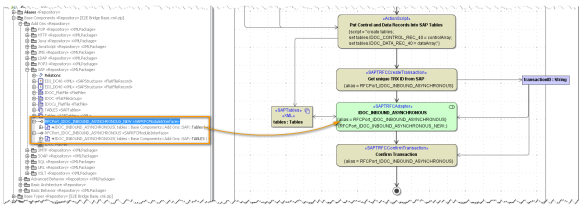
Example: IDocs from File

Concerning processing IDocs from file, there is no change (see example Add-ons/SAP/sapIDoc.xml). As before, you can

1. read the IDoc file using the <<FileSystemAdapter>>
2. transcode the data blob to an IDoc string
3. the parse this string to an IDoc object using the old <<SAPIDocParser>>

Example: Sending an IDoc to SAP

Operation **IDOC_INBOUND_ASYNCHRONOUS** now also needs to get the new SAP tables. Therefore, we provide a new operation in the Base Components.



In your model, replace **IDOC_INBOUND_ASYNCHRONOUS** coming from port RFCPort_INDOC_INBOUND_ASYNCHRONOUS with the operation provided by **RFCPort_INDOC_INBOUND_ASYNCHRONOUS_NEW** as depicted in the figure above.

Service Components

SAP Padding

- Bridge 6.0.52.0

sapPadding

- Old services have no padding specified. For these services, the padding remains **mixed** and does not change its behavior, as long as the service is not re-compiled.
- If a service is re-compiled, the **sapPadding** will get the new default **never** applied. If you run into problems with padding set to never, you can set the padding back to mixed.

For more information on this tagged value, refer to Frontend Components.

SAP Gateway

The **SAP gateway** discovery does not work anymore, if the gateway ID is specified. There are two possible solutions to this problem:

- Replace the SAP gateway ID (e.g. sapgw00) in the component diagram by the port number of the SAP system (e.g. 3300). In the default configuration, the port number will be 3300 + <system number>.
- Edit the service definition file (e.g on Linux systems: /etc/services) and add a mapping from sapgw<system number> to the appropriate port.

SAP Connection String

The SAP connection string in the component diagram **must** be provided in the following format:

<optionName>="<optionValue>"<space><optionName>="<optionValue>"...

Failure to conform with the pattern will lead unrecognized options. Those errors won't be reported, but affect SAP behavior (e.g. you'll get a SAP connection error with `CALL_FUNCTION_SIGNON_INCOMPL`).

Imported RFC WSDLs

Due to a former issue with the RFC-WSDL generator some imports of RFC WSDLs may be wrong. If a subfield e.g. was called `/BEV1/SRFUND`, tagged value **externalName** was set to `-_BEV1-_SRFUND`. Also unnecessary classes may have been generated.

This was no problem with the old sap library**,** as the importer created simple structures that were inlined into parent containers, thus keeping the binary alignment. The Bridge treated those as structures, the SAP system did not, but the binary layout was exactly the same.
Now there is no longer a binary buffer being composed and name and type need always be set correctly. So it's impossible to set/read a structure, where metadata indicates some other type.

To solve the problem, we recommend generate the affected WSDLs once again and to re-import them into your model. Alternatively, you could manually rectify the imported WSDLs.