



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

IRD to Composer Migration

Data and Services Object Migration

12/18/2025

Contents

- 1 Data and Services Object Migration
 - 1.1 External Service Object
 - 1.2 Web Service Object
 - 1.3 Database Wizard

Data and Services Object Migration

External Service Object

The External Service Object in IRD is used to exchange data with third-party (non-Genesys) processes/applications that use the Genesys Interaction SDK or any other server/application that complies with the Interaction Server communication protocol.

Composer migrates this object to **External Service block** which is very similar to the IRD object and enables calling ESP APIs. It supports all properties exposed by the IRD object except for a behavior difference regarding user data input to the ESP API. The IRD object has a checkbox to disable sending userdata in the ESP call whereas Composer, by default, does not send userdata. Instead, userdata keys to be included in the ESP call need to be specified in the Composer block.

What needs to be done manually?

1. Specify UserData to be passed in the block as the entire userdata will no longer be passed in the ESP request.

IRD Source Property	Composer Block Property	Migration Transformation	Comments
Application type	None	No need for migration	IRD uses it as a UI filter to narrow down the list of applications. Composer displays application in a tree organized by application type.
Application name	Application	Property value is migrated without change	Both properties have the same semantics and intent. They point to an application defined in configuration server
Service	Service Name	Property value is migrated without change	Both properties have the same semantics and intent.
Method	Method Name	Property value is migrated without change	Both properties have the same semantics and intent.
Timeout Property	Service Timeout Property	Property value is migrated without change	Both properties have the same semantics and intent.
Parameters	Method Parameters	Property value is migrated without change	Both properties have the same semantics and intent.
Don't send user data	User Data	Not migrated	To optimize the ESP request, Composer

IRD Source Property	Composer Block Property	Migration Transformation	Comments
			requires relevant user data keys to be specified.
Result Tab	Result Property	Not migrated unless IRD stored output to a variable	Composer uses other blocks to attach data and mapping results to variables.

Web Service Object

In IRD this object is used to invoke SOAP WebServices and get results that are then used in other parts of the strategy.

Composer migrates this object to the **Web Service block**. This block is very similar to the equivalent IRD object. It uses a WSDL file (specified as part of the project or a URL) to determine details of the SOAP WebService like available services, bindings, end points etc and exposes properties to pass in parameter values and retrieve results back into the application. In addition, this block also offers offline usage where the SOAP call is not made at runtime and instead user provided values are used for output parameters. It also provides access to the Web Services Explorer that can be used to test SOAP WebServices at design time without the need for a test call or interaction.

What needs to be done manually?

IRD does not store a WSDL service URL which is used by Composer to populate all the block properties. Therefore no properties are set automatically. Specify the WSDL URL in the Composer block and select other properties that are populated based on the WSDL URL.

IRD Source Property	Composer Target Property	Migration Transformation	Comments
WSDL Location	Service URL	None. The WSDL URL has to be specified manually in the Composer block. The IRD object does not retain the WSDL URL therefore the original URL will have to be entered again in the Composer block.	Both properties have the same semantics and intent.
Service name	Available Services	None. Both source and target properties are strings.	
Method name	Operations	None. Both source and target properties are strings.	
Method namespace	Target Name Space Uri (Hidden property)	None. Both source and target properties are strings.	In Composer Web Service block, Namespace gets

IRD Source Property	Composer Target Property	Migration Transformation	Comments
			automatically set from the parsed WSDL file.
SOAPAction	Soap Action (Hidden property)	None. Both source and target properties are strings.	In Composer's Web Service block, SOAPAction gets automatically set from the parsed WSDL file.
Request Parameters	Input Parameters	None. Both source and target properties are a list of either Variable or String.	
HTTP Authentication (Anonymous / Basic)	Authentication Type	None. Both source and target properties are strings.	Digest Authentication is not supported in Composer
Name	Login Name	None. Both source and target properties are strings / / Variable names.	Authentication User name
Password	Password	None. Both source and target properties are strings / Variable names.	Authentication password
Assign output values to variables by mapping SOAP response values	Map Output Values to Variables.	String Value "AssignByKey" in IRD will be considered as True (Boolean) in Composer	
Output Values	Output Result	None. Both source and target properties are a list of either Variable or String.	Output Params mapping can be done only when the "AssignByKey" option is chosen on the IRD side. Composer doesn't support "AttachByKey" option.

Note: As IRD doesn't have any option to choose HTTP methods, the Use Protocol property of the Composer Web Service block will always be set to "SOAP".

Database Wizard

In IRD, this object is used to query a database and the queried information can then be attached to the call or assigned to a variable.

Composer migrates this object to an instance of the **DBData Block**. The DBData block does not utilize DBServer unlike the equivalent object in IRD. Instead, it uses a set of server side pages (Java/JSP or ASP.NET/C#) that execute on the application server as part of the Composer generated application. This block uses a database connection defined in the Composer project that can be configured to use connection pooling transparently. It includes a visual query builder and a stored

procedure call helper to visually design a query or to invoke a stored procedure and test it from within the block. If situation where the query is too complex to be created visually or is already available, it supports specifying a file containing a query to be used instead of a query framed using the query builder.

What is created automatically?

These significant differences in paradigm mean that the connection information is not migrated over. Instead migrated creates a DBData block, creates a text file containing the query from the IRD object and sets the DBData block to use this file. Stored Procedures calls are not migrated over automatically and should be specified using Composer's Stored Procedure Helper UI.

What needs to be done manually?

1. Check the SQL query written to the .sql file that the DBData block points to.
2. Create a Database connection using the Database Connection Manager. Set the DBData block to use this connection.

To see a list of supported databases, please consult online help in Composer.

IRD Source Property	Composer Block Property	Migration Transformation	Comments
SQL	SQL File Property set to a file containing the SQL statement	The SQL will be extracted and written to a .sql file. The DB Data block will point to this .sql file.	
Access Point	Database connection.	Connection information is not migrated.	DBServer is no longer used. See post-migration manual steps.