IRD to Composer Migration

IRD and Function Migration

12/15/2025

Contents

IRD and Function Migration

In IRD strategies, functions may be used:

- Within an explicit IRD object such as the Function object or
- As a part of the setting of various IRD object parameters.

IRD Objects Supporting Function Expressions

The table below shows how migration handles IRD Objects that support using Functions in Expressions.

Function Migration Types

IRD Object	Composer Block Used with Valid Expression	Failed/Not Supported (In the ECMAScript block, the failed expression will be commented out.)
If Object	Branching Block	SCXML State Block
Assign Object	Assign Block	ECMAScript Block
MultiAssign	Assign Block	ECMAScript Block
Function Object	ECMAScript Block	ECMAScript Block
MultiFunction Object	ECMAScript Block	ECMAScript Block
Selection Object	Target Block	Target Block

Function Migration Types

Migration attempts to parse function expressions, break them down into individual function calls, and then form an equivalent expression. However, this is a complex operation. Some IRD functions are not supported in Orchestration while other functions are implemented as either Events or Services in Orchestration Server, and their Orchestration equivalent is now a Composer block. Given the different approaches used in Orchestration, migration classifies functions into the following types and handles each type differently.

Function Migration Types

Type	Description	Comment
Direct Mapping	A simple mapping exists. Migration will replace with an equivalent function call.	
Tag Mapping Direct	Function now requires an SCXML tag. For example, GetPriority()	

Type	Description	Comment
	now needs a <queue:query> and then pick up the property from the Done event.	
Tag Mapping Indirect	Function needs to be converted to a property of a block, which was migrated as a counterpart of another block. Effectively, it will merge into another block. For example, Priority() now will become a block property but will not create a new block.	
Composer block	Function requires a new block to be implemented in Composer.	
Composer Implementation	Function for which Composer implements an ECMAScript function.	
Not Supported	Not supported in IRD to Composer migration. You will need to migrate these functions manually. Migration will add a placeholder Generic State block with a SCXML comment inside it. Comment will include IRD block properties. You can add arbitrary SCXML code in this block to achieve the desired functionality.	

Migration Based on IRD Categories

The *Universal Routing 8.1 Reference Manual* organizes functions into categories. The IRD Function object dialog box also uses these categories. The table below summarizes migration handling for each category of IRD functions. For migration detail, click a function category name.

IRD Function Category	General Migration Comment
CallInfo	Most functions map directly to Composer properties, with a few exceptions
Configuration Options	Primarily mapped to Genesys Functional Modules, with a few exceptions
Data Manipulation	Migration through either Database Wizard or External Service block
Date/Time	Primarily replaced with standard ECMAScript Date() manipulation, with a few specialized functions
Force	Implemented as a service
List Manipulation	Replaced with ECMAScript utility script
Miscellaneous	This category is so wide that functions in this category fall within all migration classification

IRD Function Category	General Migration Comment
	types
Reporting	Supported via inbuilt functions
Statistical	Primarily all supported functions, with one exception implemented as a service
String Manipulation	Standard ECMASCript functions
Target Manipulation	A mixture of functions and services is used