Composer Help

Composer Versus IRD

5/3/2025

Composer Versus IRD

Contents

Integrated Development Environment

- Composer is a single Integrated Development Environment (IDE) for creating applications to orchestrate the entire customer experience. Composer-created voice and routing applications can command and control the customer experience through all channels (IVR, voice, e-services, and so on).
- Composer's open framework enables widely-available, existing competencies to be used to create reusable components that manage the customer experience. The IDE allows both customers and integrators to utilize existing code sets (HTML, VXML, java, perl, REST and others) to control the customer experience.
- The open framework also allows simplified integration into all Enterprise applications to harness the information within the Enterprise to drive and personalize the customer experience.

Differences from IRD

In the past, Interaction Routing Designer was the only Genesys tool to create routing applications. Genesys Composer is now the tool of choice for creating both routing and voice self-service applications.

A few of the differences between Composer and Interaction Routing Designer are listed below.

- Composer is integrated with Orchestration Server allowing you to manage customer conversations spread out over time using the ORS session-based functionality and persistent storage as well as Orchestration Extensions.
- Composer encompasses IRD's functionality and much more routing functionality in general.
- Composer lets you create routing applications using an open language (SCXML) and ECMAScript for decision-making. In contrast, IRD uses a closed Genesys proprietary language (IRL) and you are limited to IRD's objects and functions.
- Composer gives the option of writing your own SCXML code and/or using predefined blocks.
- Unlike IRD, you can also use Composer to create voice self-service applications for Genesys Voice Platform, including VoiceXML and CCXML-based applications. You can also create integrated voice and routing applications.

Composer Routing Application Types

You can use Composer's predefined blocks and/or write your own code to create routing applications that route based on various criteria such as:

-
  - **Agent, Agent Group, ACD Queue, Place, Place Group, Route Point, Skill, or Variable**
  - **last called agent**

- **date and time**
- **the value of a statistic**,
- **dialed number (DNIS)**
- **originating number (ANI)**
- **percent and conditional routing**

The above list is by no means complete. It represents only a few types of routing applications that can be created in Composer. Since Composer uses open languages (SCXML and ECMAScript), you are not limited to its pre-defined blocks, but are free to create many types of routing applications.

Migrating IRD Strategies into Composer

Note: You can migrate routing strategies created in IRD into Composer. For more information, see the IRD to Composer Migration Guide.

Composer Blocks Mapped to IRD Objects

Composer refers to the fundamental element of a workflow as a "block" whereas in IRD documentation, this element is referred to as an "object." The tables below group IRD objects based on their IRD toolbar category name and point to the corresponding functionality in this release of Composer. Summary information is presented below.

Data & Services

IRD Object Name	Composer Block Name	Description
Database Wizard	DB Data	DB Data retrieves information from the database. Uses a the Query Builder Query Builder.
Web Service	Web Service	Invokes Web Services. GET, POST and SOAP over HTTPS are supported.
	Web Request	Invoke any supported HTTP web request or REST-style web Service. See sample: Routing Based on Web Request.

Miscellaneous

IRD Object Name	Composer Block Name	Description
Assign Multi-Assign	Assign	Assigns a computed value/ expression or a literal value to a variable. Variables are defined in the Entry block. Capable of

		multiple assignments.
Call Subroutine	Subroutine	Creates reusable sub-modules.
Entry	Entry	Sets global error (exception) handlers. Defines global variables (see Variables section below).. All routing strategy diagrams must start with an Entry block.
Exit	Exit	Terminates the strategy and returns control back to calling workflow in case of a subroutine.
Error Segmentation	Multiple error output ports can be created in Composer blocks based on each block's Exception property.	
Function Multi-Function	ECMAScript	Builds an ECMAScript expression using the Expression Builder. Many URS functions are available as Genesys Functional Modules described the Orchestration Server Documentation Wiki Can invoke multiple functions.
If	Assign, Branching, ECMAScriptBlock blocks all open Expression Builder	Expression Builder can be used to create IF expressions.
Multi-Attach	ECMAScript	Can be used for attaching data to an interaction.

Routing

IRD Object Name	Composer Block Name	Description
Selection	Target	Routes an interaction to a target, which can be Agent, AgentGroup, ACDQueue, Place, PlaceGroup, RoutePoint, Skill, or Variable. Skill target uses Skill Expression Builder.
Percentage	Target	Order Property Statistics Order property in Target block, lets you perform percentage allocation. Also see sample: Routing Based on Percent Allocation.
Default	Default Route	Routes the interaction to the default destination.
Routing Rule		Orchestration Server 8.1 does not support service level routing rules.
Switch to Strategy		Orchestration Server 8.1 does not support switch to strategy

		routing rules.
Force Route	Force Route	Not exposed as a routing rule in Composer.
Statistics	Target	Although statistical routing rules are not yet supported as in IRD's Statistics routing object, users can use the Target object Property Statistic property to route based on the value of a statistic. A Statistics Manager and Builder let you create your own statistics from URS predefined statistics.

Segmentation

IRD Object Name	Composer Block Name	Description
ANI	Branching	See Your First Application: DNIS Routing for an example.
DNIS	Branching	See Your First Application: DNIS Routing for an example.
Date	Branching	See the sample: Routing Based on Date and Time.
Day of Week	Branching	See the sample: Routing Based on Date and Time.
Time	Branching	See the sample: Routing Based on Date and Time.
Classification Segmentation	Branching	For classification segmentation, an ECMAScript function determines if a particular category name or ID exists in the array of category objects represented by an application variable.
Generic	Branching	Use as a decision point in a workflow. It enables you to specify multiple application routes based on a branching condition.

Also see Context Services Blocks.

Voice Treatment

See Composer Equivalent to IRD Treatment.

eServices Multimedia

See Composer Equivalent to IRD Multimedia.

Outbound

See Outbound Common Blocks

Context Services

See Context Services Blocks

Business Process

See Interaction Processing Diagrams Overview and Interaction Process Diagram Blocks.

Reusable Objects

- IRD List Object: See Composer's List Object Manager.
- IRD Variable List Dialog Box: See Entry block Variables property.

In contrast to IRD, which defines variables in a special dialog box outside of the strategy, Composer defines both workflow and Project variables.