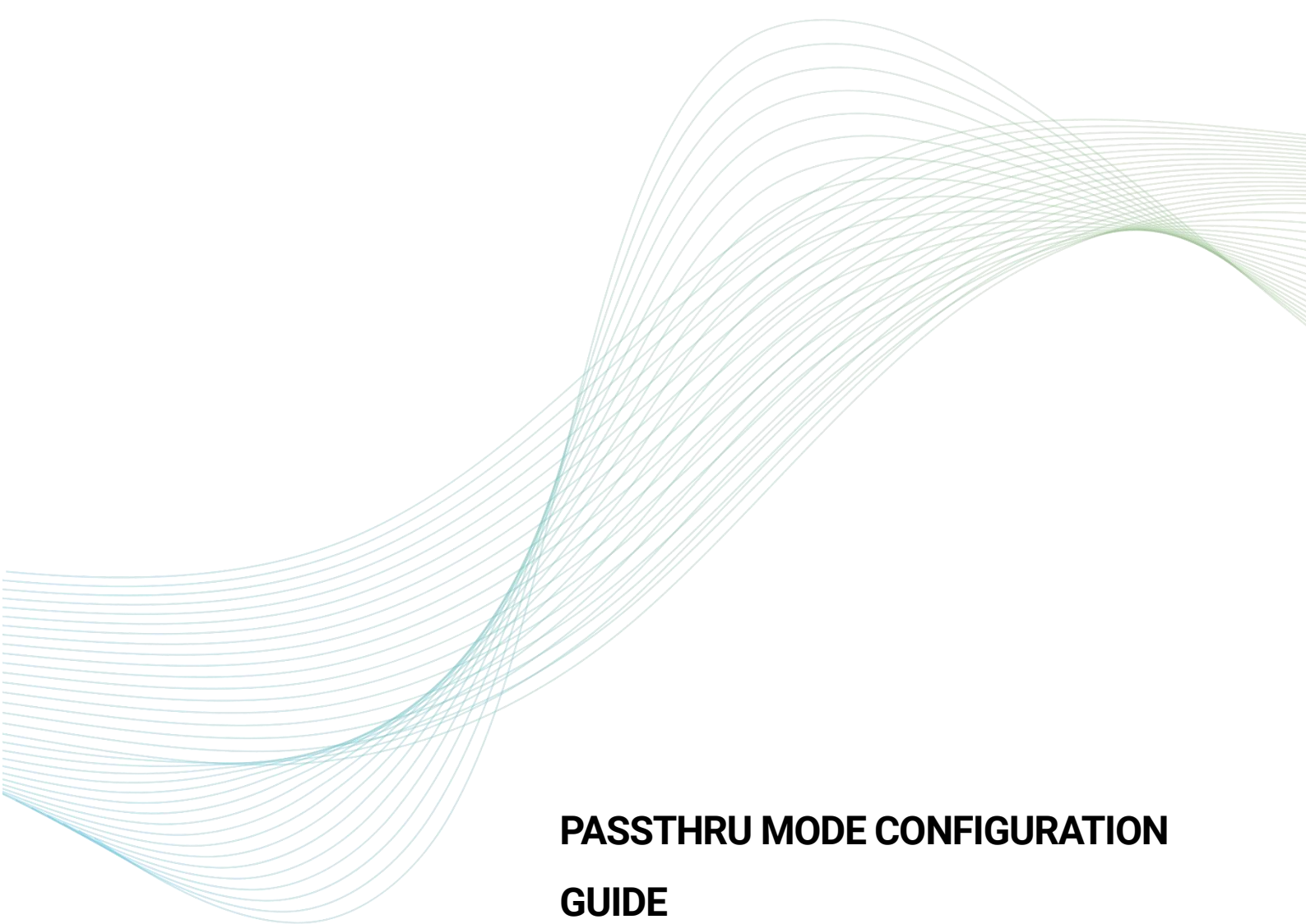


A series of thin, overlapping, wavy lines in shades of light blue and green flow across the middle of the page, creating a sense of motion and depth.

PASSTHRU MODE CONFIGURATION GUIDE

Internal Technical Documentation

Table of Contents

Table of Contents	2
1 Introduction.....	3
2 Passthru Mode Configuration	4
3 How Does Passthru Mode Work	6

1 Introduction

A **Passthru** mode is a service mode in which a service processes and forwards third-party web service requests directly to their intended destination without significant alteration or processing. This mode essentially acts as an intermediary, ensuring that the requests and responses pass through seamlessly.

The **Passthru** mode is used either as a service configuration or as a parameter in an API request to the service. This means that all service requests, or specific requests, will be handled as passthru based on the service configuration for passthru in the **CodiacSDK.Core.config** file and the presence of the passthru parameter in a particular request.

By minimizing processing, passthru mode can improve performance and reduce latency. This is beneficial in scenarios where the speed of data transmission is critical.

The main goal of the **Passthru** mode is to override the security checks specified in the **c_SpecialAccessRestriction** table.

If the passthru parameter is enabled in the service configuration or included as a request parameter, any alias marked as eligible for passthru (with the **Y** value in the **PassthruEnabled** field) will bypass the security specifications from the **c_SpecialAccessRestriction** table.

Setting up passthru mode typically involves configuring the service to forward requests based on specified rules or criteria, ensuring that the routing logic is correct to prevent misdirection or data loss.

By using passthru mode, services can efficiently handle third-party web service requests, providing a transparent and efficient data transmission path between different systems or components.

2 Passthru Mode Configuration

To launch the Codiack service in passthru mode, you need to add a passthru record to the **CodiackSDK.Core.config** file and set the **is_passthru** record to **true**.

```
<config>
  <datapool libname = "CodiackSDK.PostgreSQL.dll" connstring = "hostaddr = 192.168.100.215 dbname = CODIACK_DB_20181108"
  <logs store_success = "true" store_failed = "true" store_db = "false" plugins_logs = "true" />
  <security encryption = "false" hashpassword="nohash" />
  <passthru is_passthru = "true" acceptXML="true" />
</config>
```

Each record in the **AliasTable** table contains a **PassthruEnabled** field. If this field has a value of **Y**, the database field specified in the **AliasDatabaseField** and the table specified in the **AliasDatabaseTable** are accessible for passthru requests; otherwise, they are not accessible.

215 - CODIACK_DB_20181108 - public.AliasTable

```
1 SELECT * FROM public."AliasTable"
2 ORDER BY "AliasCode" ASC
```

Data Output	AliasSQLStatement	AliasModule	AliasModuleType	DefaultGroupUserIsNoAccess	DefaultValuesIsNoAccess	LockedBy	LockTime	PassthruEnabled
baseField varying (50)	text	character varying (1)	character varying (3)	character varying (1)	character varying (1)	character varying (40)	character varying (16)	character varying (1)
int	[null]	F		F	F	[null]	[null]	Y
any	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
ser	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
late	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
ype	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
late	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y
	[null]	[null]	[null]	[null]	[null]	[null]	[null]	Y

The **SpecialAccessRestriction** table verifies the rights for user groups. This table is inner-joined with the **AliasTable** using the **AliasCode** value. Rights for each field are checked in the **queryStringFieldsRights** function within the **SQLUtility.cpp** file for each communication library (PostgreSQL, MicrosoftSQL, Oracle, and MySQL).

215 - CODIAC_DB_20181108 - public.SpecialAccessRestriction

1

SELECT * FROM public."SpecialAccessRestriction"

2

ORDER BY "Id" ASC

Data Output

Explain

Messages

Query History

	Id [PK] integer	AliasCode character varying (50)	Entity character varying (50)	UserGroupValue character varying (1)	Rights character varying (1)
1	5	Custom.TestModule.Rang...	Admin	U	U
2	6	Alias.Plan.InsuranceType	PO	V	N
3	7	Alias.Plan.PlanDesc	Standard_Group	G	R
4	8	Alias.workflow.user_id	Standard_Group	G	U
5	9	Alias.workflow.user_id	Admin	U	U
6	10	Alias.workflow.screen_id	Standard_Group	G	U

3 How Does Passthru Mode Work

The **CoreSDK.cpp** file checks the configuration settings in both the Custom and Common controllers. If the **is_passthru** option is set to true, the Core plugin executes the passthru record for the request. For example:

```
if (m_CodiacAPIConfig.m_isPassthru) {  
    jsonFuncParam_Upd[TEXT("passthru")] = web::json::value(true);  
}
```

This record is verified within the plugins and passed to the communication libraries, as seen in the **SQLUtility.cpp** file below, where the SQL request is then executed. For example:

```
std::string queryStringFieldsRights(KeyInfo& ki, std::map<std::string, std::string>  
tableFiledList)
```