DynamoDB support for Shell source code

CAST supports **DynamoDB** via its Shell extension. Details about how this support is provided for **Shell source code** is discussed below.

Objects

Objects created specific for DynamoDB AWS service interpretation.

Icon	Metamodel Name
	SHELL DynamoDB Database
	SHELL DynamoDB Table
	SHELL Unknown DynamoDB Table

Supported versions

Versions	Supported
V1	✅
V2	✅

Supported commands

DynamoDB

All commands listed below have as **caller** a *SHELL program or Function* and as **callee** a *SHELL (Unknown) DynamoDB Table.*

API (with CRUD links)	Link Type	Remark
aws dynamodb **get-item**	useSelectLink	
aws dynamodb **scan**	useSelectLink	
aws dynamodb **query**	useSelectLink	
aws dynamodb **create-backup**	useSelectLink	
aws dynamodb **batch-get-item**	useSelectLink	Only inline json input data supported
aws dynamodb **batch-write-item**	useInsertLink, useDeleteLink	Only inline json input data supported
aws dynamodb **restore-table-to-point-in-time**	useSelectLink, useInsertLink	
aws dynamodb **restore-table-from-backup**	useUpdateLink	
aws dynamodb **delete-table**	useDeleteLink	

aws dynamodb **delete-item**	useDeleteLink	
aws dynamodb **transact-write-items**	useUpdateLink, useDeleteLink	Only inline json input data supported
aws dynamodb **update-item**	useUpdateLink	
aws dynamodb **update-table**	useUpdateLink	
aws dynamodb **put-item**	useUpdateLink, useInsertLink	

API	Effect
aws dynamodb **create-table**	A DynamoDB table is created
aws dynamodb **create-global-table**	A DynamoDB table is created

What results can you expect?

A DynamoDB Table object can be created by two different situations. In the first one, explicit creation of tables is detected by the presence of **create-table** or **create-global-table** commands and therefore bookmarked to the corresponding command string. In the second situation, there is no explicit table creation, but references to tables are present in different commands. In this case, a DynamoDB Table object is created by reference and all references are bookmarked into the object.

When the name of a table is not resolved (either because of absence of information or technical limitations) a *Shell Unknown DynamoDB Table* is created instead. A single unknown table is created per project.

Below we illustrate the expected results with various examples. The code samples used are adaptations from available examples in official AWS documentation pages.

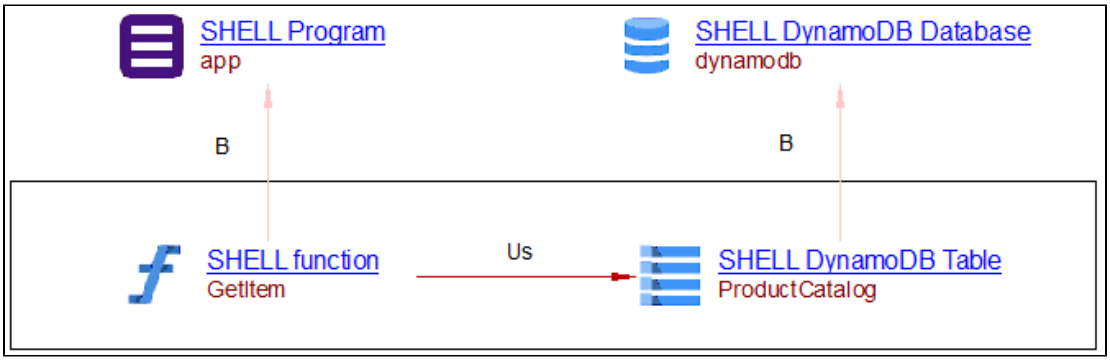
Example 1

The following code in a file *app.sh* contains a single *aws-dynamodb* command enclosed in a SHELL function (*GetItem*) requesting an item from a DynamoDB table of name ProductCatalog:

```
#!/bin/sh


GetItem () {
   aws dynamodb get-item --table-name ProductCatalog --key '{"Id": {"N": "102"}}' --projection-expression "#T,
#C, #P" --expression-attribute-names file://names.json
}
```

As a result, a *SHELL DynamobDB Table* is created after the reference to the table, together with a parent SHELL DynamoDB Database objet. A link is also created of type *useSelectLink* between the caller GetItem function and the table.



Note that a single SHELL DynamoDB Database objet is created per project (if SHELL DynamoDB Tables are present).

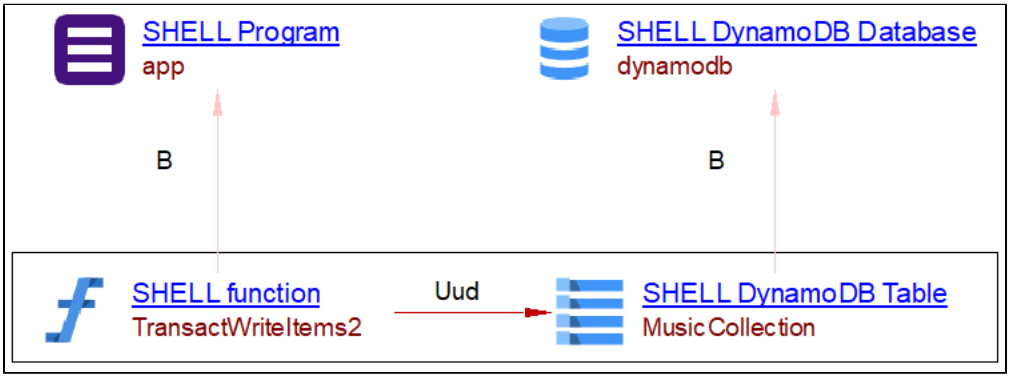
Example 2

The following code in a file *app.sh* contains a single *aws-dynamodb* command enclosed in a SHELL function (TransactWriteItems2) writing and deleting items (details are omitted for simplicity) from a DynamoDB table of name MusicCollection:

```
#!/bin/sh


TransactWriteItems2() {
   aws dynamodb transact-write-items --transact-items '[{"Update": {"TableName": "MusicCollection"}},{"Delete":
{"TableName": "MusicCollection"}}]' --return-consumed-capacity TOTAL --return-item-collection-metrics SIZE
}
```

The references to the table are inside the --transact-items argument given as a json string together with the actions (Update, Delete). The analyzer will parse the json input string and create the corresponding objects and links (useUpdateLink and useDeleteLink):



Limitations

Using external .json files as inputs is not yet supported (ex. --transact-items file://transact-items.json)