

# Schema Calls

## Introduction

Schema calls describe the signature used for a schema function call. The schema framework uses this to validate the arguments when the call is made in schema script and also to show the correct signature in the symbol picker in the user interface.

## Creating and using

Register the call with [SchemaClassRegisterCall](#)

Add parameters to the call with [SchemaCallRegisterParam](#)

Set the call as the signature for a schema member when you register the member with [SchemaClassRegisterMember](#)

## Example

```
void RegisterSchema()
{
    HINSTANCE instance = AfxGetResourceHandle();
    SchemaHandle client = SchemaOpenRegistry(L"MySchemaClientName");

    // Type is undefined and handler is null because we never instantiate this class
    SchemaHandle schemaClass = SchemaRegisterClass(client, LocString(L"MyClassName",
instance, IDS_CLASS_NAME, IDS_CLASS_DESCRIPTION), SystemType::Undefined, NULL);

    // Register a signature called "FileExistsParams" that takes a string
    SchemaHandle fileExistsParams = SchemaClassRegisterCall(client, schemaClass,
L"FileExistsParams");
    SchemaCallRegisterParam(fileExistsParams, LocString(L"ParamIdName", instance,
IDS_PARAM_FRIENDLY_NAME, IDS_PARAM_DESCRIPTION), SystemType::AString, false);

    // Register a class member that has a signature that is the previously registered
    "FileExistsParams" call.
    SchemaHandle fileExists = SchemaClassRegisterMember(client, schemaClass,
LocString(L"FileExists", instance, IDS_FUNC_NAME, IDS_FUNC_DESCRIPTION),
eSchemaReadOnly, SystemType::Boolean, eSchemaVarDispCall, FileExistsFunc, -1,
L"FileExistsParams");
}
```