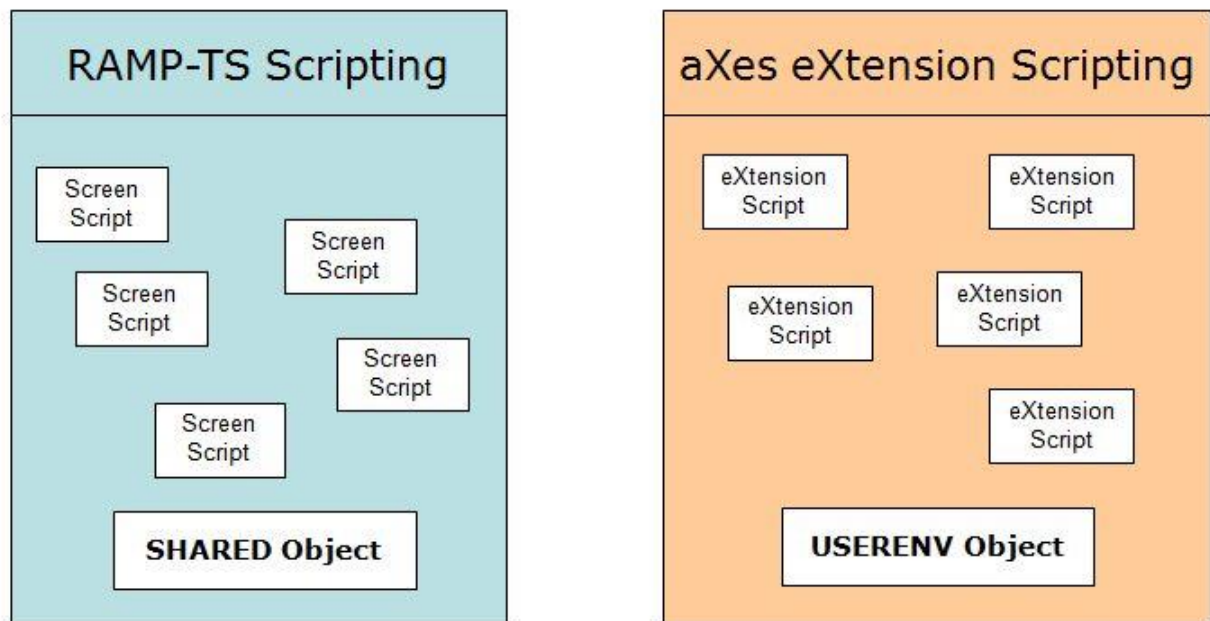


Cross-Scripting: Using RAMP-TS functions and properties in your aXes scripts and vice versa

(For RAMP-TS developers.)

In the RAMP-TS environment, functions can be shared by scripts by using the SHARED object.

In the aXes environment, functions can be shared by scripts by using the USERENV object.



aXes scripts can use RAMP-TS functions and properties, including the SHARED object, *if* the aXes script is being executed within a RAMP-TS application at runtime. But they must prefix the function name with "RAMP." , and they must be coded to handle the absence of the RAMP runtime environment.

```
if (RAMP != null)
{
    RAMP.ALERT_MESSAGE("my alert message");
}
```

or

```
if (RAMP != null)
{
    RAMP.SHARED.Test ;
}
```

Conversely, RAMP-TS scripts can always refer to aXes functions in USERENV

```
USERENV.asString(26);
```

Key points to remember:

- ❖ RAMP-TS screen navigation scripts and aXes eXtension scripts execute in different name spaces.
- ❖ The aXes eXtensions scripting environment is always available.
- ❖ The RAMP-TS scripting environment exists only when the RAMP-TS execution environment is running.
- ❖ The RAMP-TS scripting environment is not available when you execute an eXtension's script in Axes all by itself - or when executing it from RAMP-Tools.

Example: To execute RAMP functions in an Axes Script

Suppose you are writing an onClick script for a push button extension, and you want to be able to use the RAMP-TS function `ALERT_MESSAGE("My message")`.

To use the function in an aXes script you must specify that it is a RAMP script. You do this by using the RAMP name space.

```
RAMP.ALERT_MESSAGE("My message");
```

This will work provided that the aXes script always runs in aXes within a RAMP-TS application. But what if it is sometimes run in a standalone aXes environment, or in the RAMP Tools window?

You handle this by checking for the existence of RAMP, before trying to use `ALERT_MESSAGE`.

```
if (RAMP != null)
{
    RAMP.ALERT_MESSAGE("My message");
}
else
{
    /* use some other method, or do nothing */
}
```

Similarly, if your aXes onClick routine wants to use a function that the RAMP-TS scripts share through the SHARED object, (held in `UF_SY420_RTS.js`),

```
if (RAMP != null)
{
```

```

    RAMP.SHARED.myFunction1("x", "y", "z");
}
else
{
    /* use some other method, or do nothing */
}

```

Example: To execute an aXes USERENV function in a RAMP Script

Suppose you are writing a RAMP-TS vHandle_ARRIVE routine, and want to use function asString that the aXes eXtension scripts all share in USERENV.

If using RAMP-TS, then the aXes scripting environment is always available, so you just write:

```
USERENV.asString(26);
```

A table of scripting options

| Situation | In a RAMP-TS Script | In an eXtension script |
|--|--|--|
| Use RAMP-TS function ALERT_MESSAGE("Hello") | Use ALERT_MESSAGE("Hello") Or RAMP.ALERT_MESSAGE("Hello") | Must use RAMP.ALERT_MESSAGE("Hello") and cater for situations where RAMP is null. |
| Access SHARED property named Test. | Use SHARED.Test Or RAMP.SHARED.Test | Must use RAMP.SHARED.Test and cater for situations where RAMP is null. |
| Access USERENV property named Test | Use USERENV.Test | Use USERENV.Test |

