

RTSC Codec & Server Package Wizards

Generated Content

Texas Instruments
December 2007

Technology for Innovators™

 TEXAS INSTRUMENTS

Table Of Contents

- Codec Package Structure
 - “Base” Package
 - Codec Engine Package
- Server Package Structure

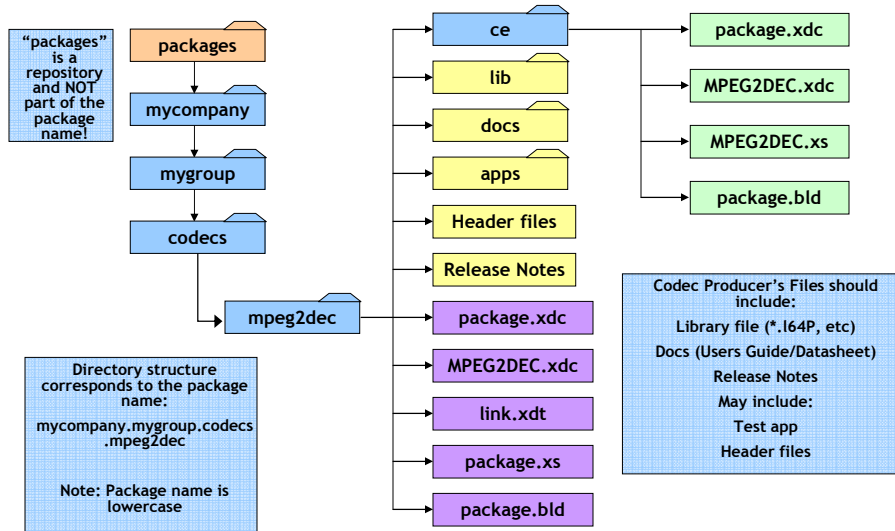
Technology for Innovators™

 TEXAS INSTRUMENTS

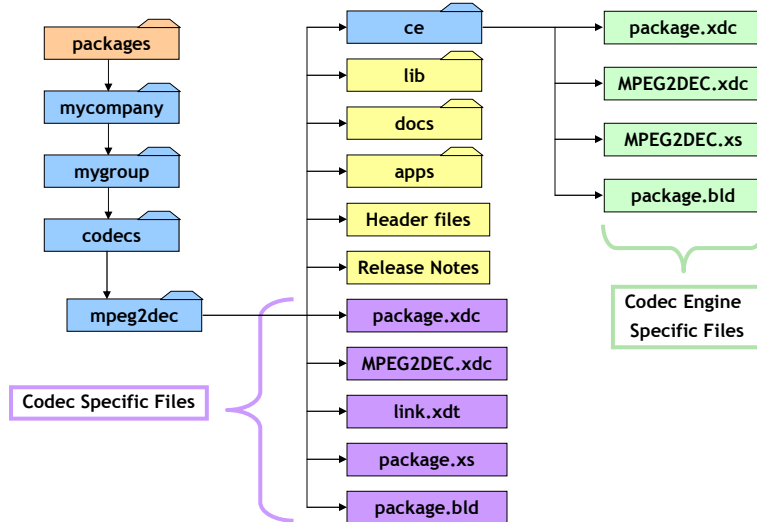
Table Of Contents

- Codec Package Structure
 - “Base” Package
 - Codec Engine Package
- Server Package Structure

Codec Package Structure



Codec Package Structure



Technology for Innovators™

TEXAS INSTRUMENTS

Package Specification File

```

/*!
 * ===== package.xdc =====
 */
package mycompany.mygroup.codecs.mpeg2dec [1, 0, 0] {
    module MPEG2DEC;
}
    
```

- Role: defines that this is an XDC package.
- Package name imposes a directory structure
- Module declared as "MPEG2DEC"
 - XDC looks for a file named MPEG2DEC.xdc in the same directory
- ⚠ Convention: Packages are named in lower case and modules start with a capital letter
- ⚠ Common Mistake: Do NOT add "requires ti.sdo.ce.video" here—this belongs in the CE directory! We want the base codec package to be framework independent

Technology for Innovators™

TEXAS INSTRUMENTS

Compatibility Key

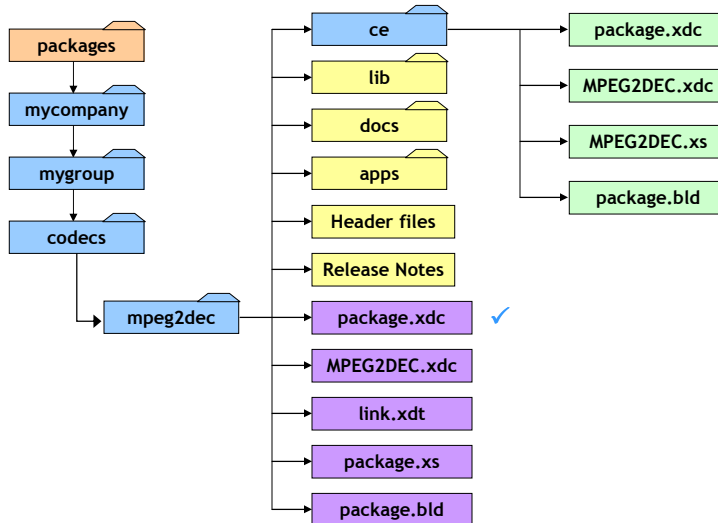
```
/*!  
 * ===== package.xdc =====  
 */  
  
package mycompany.mygroup.codecs.mpeg2dec [1, 0, 0] {  
    module MPEG2DEC;  
}
```

Compatibility Key
NOT a version
number!

Role: captures the binary and source level compatibility of this version of the package with its earlier releases

- > 3 Digit Key: [M, S, R]
 - > Major: Changes in M denote incompatibility
 - > Source: Changes in S guarantee source backwards compatibility
 - > Radix: Changes in R guarantee binary backwards compatibility
- > It allows packages depending on this package to specify which releases of the package are acceptable and which are not
- > A detected incompatibility tells the user to relink the application, recompile their source code, or (worst case) rewrite their source code - or get a different version of the package

Codec Package



Module Specification File

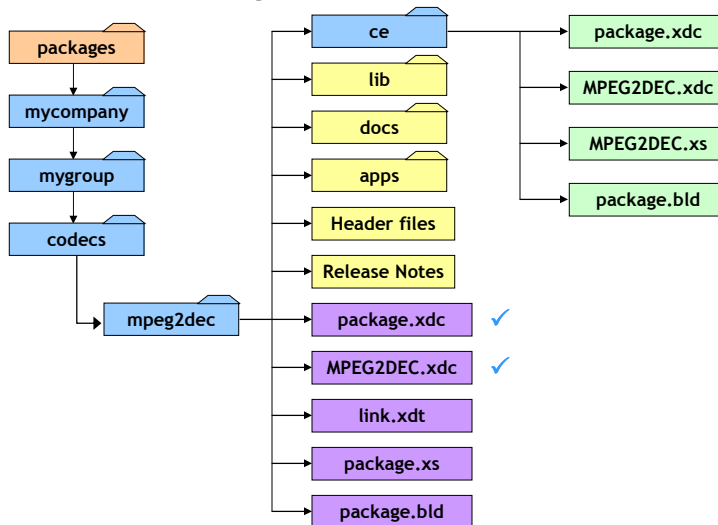
```
/* ===== MPEG2DEC.xdc =====*/  
  
metaonly module MPEG2DEC  
{  
    config Bool watermark = true;  
  
    config String codeSection;  
    config String udataSection;  
    config String dataSection;  
}
```

- > Role: exposes configurable package parameters
- > One use of configurable parameters is to determine which library to return (more on this to come)
- > Other examples of configurable Boolean: “useDMA” or “useVICP”
- > Section Name variables should be left uninitialized. The server configuration file will take care of the placement (e.g. Don’t have codeSection=“DDR”, etc.)

Technology for Innovators™

TEXAS INSTRUMENTS

Codec Package



Technology for Innovators™

TEXAS INSTRUMENTS

Linker Template File

```
/*!
 * ===== link.xdt =====
 */
SECTIONS
{
% if (this.MPEG2DEC.dataSection) {
    .const:MPEG2VDEC_TI_dSect1 >
    `this.MPEG2DEC.dataSection`
%}

% if (this.MPEG2DEC.codeSection) {
    .text:MPEG2VDEC_TI_cSect1 > `this.MPEG2DEC.codeSection`
%}

% if (this.MPEG2DEC.udataSection) {
    .far:MPEG2VDEC_TI_uSect1 > `this.MPEG2DEC.udataSection`
%}
}
```

- **Role:** Specify the groupings of the sections so that the server integrator can optimally place them
- Section alignment can be specified here by:
 - `.const:MPEG2VDEC_TI_dSect1 > `this.MPEG2DEC.dataSection`, align = 0x10000`
- Recall that the variables `codeSection`, `dataSection`, and `udataSection` are defined in `MPEG2DEC.xdc`

Technology for Innovators™

TEXAS INSTRUMENTS

Linker Template File

```
SECTIONS
{
% if (this.G711DEC.dataSection) {
    .g711dec_const > `this.G711DEC.dataSection`
%}

% if (this.G711DEC.codeSection) {
    .text:algNumAlloc > `this.G711DEC.codeSection`
    .text:algInit > `this.G711DEC.codeSection`
    .text:algAlloc > `this.G711DEC.codeSection`
    .text:algFree > `this.G711DEC.codeSection`
    .text:exit > `this.G711DEC.codeSection`
%}
}
```

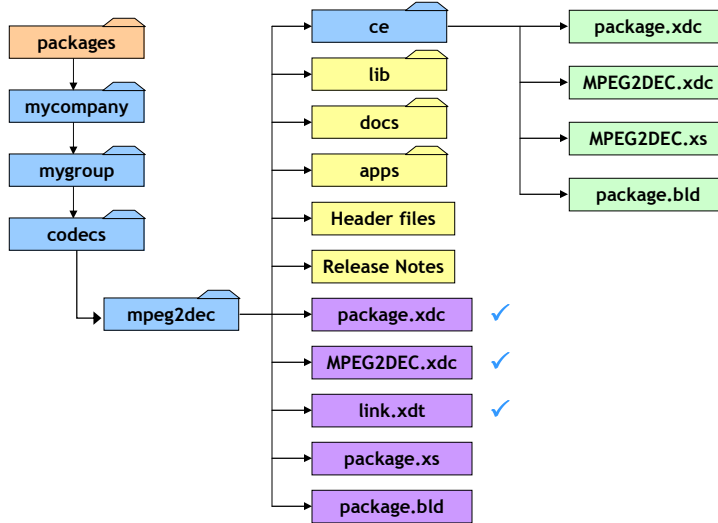
Poorly named section!

- **Note:** `.g711dec_const` is not prefixed with `.far` or `.const` subsection! The linker can arbitrarily place it (e.g. L1PSRAM) and crash the whole system!
- View the sections contained within a library using the `sectti.pl` script
 - See `CG_XML` at https://www-a.ti.com/downloads/sds_support/applications_packages/cg_xml/index.htm

Technology for Innovators™

TEXAS INSTRUMENTS

Codec Package



Technology for Innovators™

TEXAS INSTRUMENTS

Package Configuration Script

```

/*
 * ===== package.xs =====
 */

function getLibs(prog)
{
    var lib = null;
    if (prog.build.target.isa == "64P") {
        if ( this.MPEG2DEC.watermark == false ) {
            lib = lib/mpeg2vdec_ti_prod.l64P;
        }
        else {
            lib = "lib/mpeg2vdec_ti_eval.l64P";
        }
        print(" will link with " + this.$name + ":" + lib);
    }
    return (lib);
}

```

```

function getSects()
{
    var template = null;

    if (Program.build.target.isa == "64P") {
        template =
        "mycompany/mygroup/codecs/mpeg2dec/link.xdt";
    }

    return (template);
}

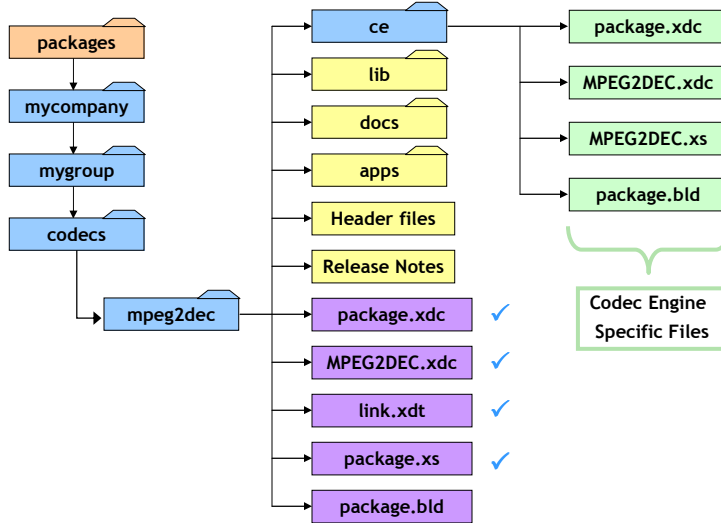
```

- > Role: Must implement getLibs() to indicate which library to return
- > Also implements getSects() to return the linker template, link.xdt
- > May optionally implement other functions written in JavaScript, such as a program integrity check

Technology for Innovators™

TEXAS INSTRUMENTS

Codec Package



Technology for Innovators™

TEXAS INSTRUMENTS

Package Build Script

```

/*
 * ===== package.bld =====
 * Export all files in this directory
 */
Pkg.attrs.exportAll = true;
    
```

← The usual method

```

/*
 * ===== package.bld =====
 * Selectively export files and directories
 */
Pkg.otherFiles = [
    'lib',
    'docs',
    'mpeg2dec_1_00_001_ReleaseNotes.pdf',
];
    
```

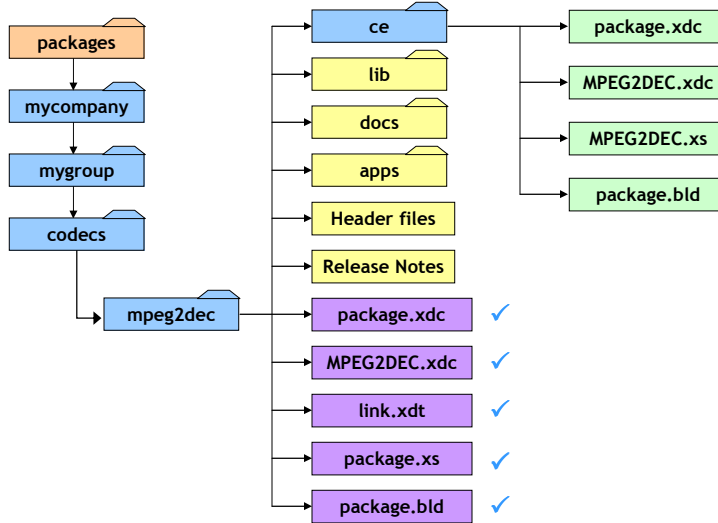
← But if you must...

- Role: Script indicates how to build - but we're primarily interested in including "extra files"
- Written in JavaScript, following the XDC Build Object Model (which defines a set of objects and functions that let the user describe what to build in a portable, OS-independent way)

Technology for Innovators™

TEXAS INSTRUMENTS

Codec Package



Technology for Innovators™

TEXAS INSTRUMENTS

Package Specification File

```
requires mycompany.mygroup.codecs.mpeg2dec;

/!
* ===== package.xdc =====
* Provides ICodec interface adapter for
* mycompany.mygroup.codecs.mpeg2dec codec.
*/

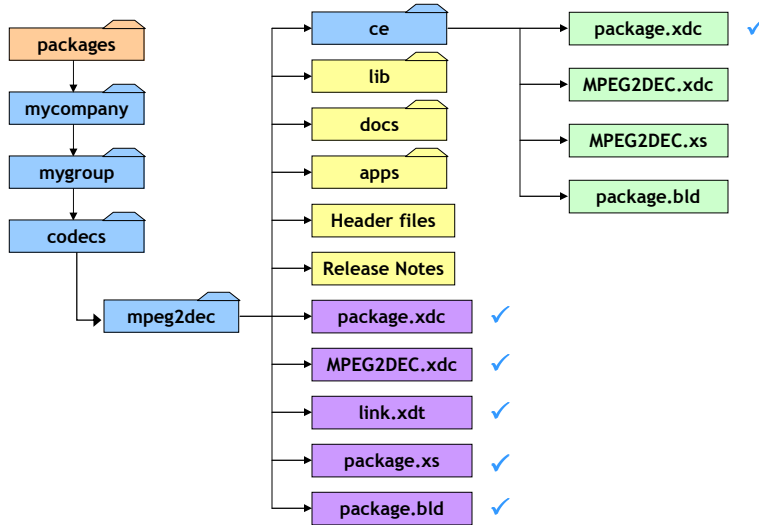
package mycompany.mygroup.codecs.mpeg2dec.ce [1, 0, 0] {
    module MPEG2DEC;
}
```

- > Role: declares that the /ce directory is itself a package that interfaces with ICodec
- > Requires statement needed for the “base” codec package

Technology for Innovators™

TEXAS INSTRUMENTS

Codec Package



Technology for Innovators™

TEXAS INSTRUMENTS

Module Specification File

```

/* ===== MPEG2DEC.xdc ===== */

metaonly module MPEG2DEC inherits ti.sdo.ce.video.IVIDDEC
{
  readonly config mycompany.mygroup.codecs.mpeg2dec.MPEG2DEC.Module alg =
    mycompany.mygroup.codecs.mpeg2dec.MPEG2DEC;

  override readonly config String ialgFxns = "MPEG2VDEC_TI_IALG";
  override readonly config String idma3Fxns = "MPEG2VDEC_TI_IDMA3";
}

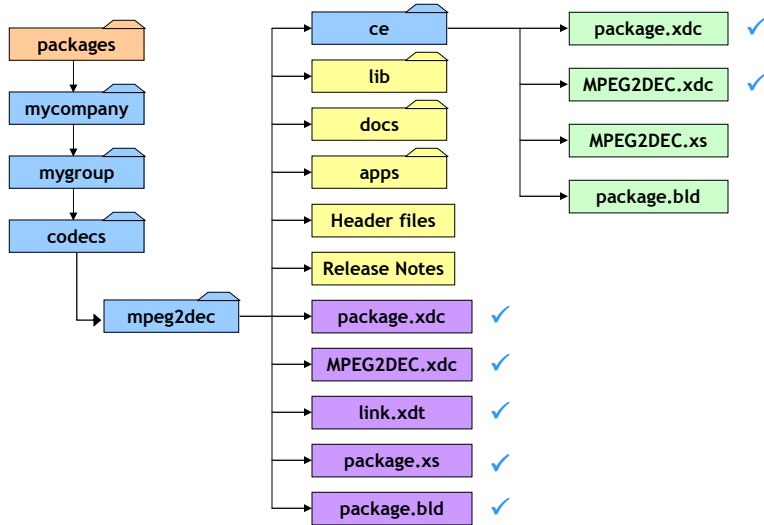
```

- > Role: Specifies information needed to integrate with the Codec Engine
- > By inheriting it.sdo.ce.video.IVIDDEC, MPEG2DEC declares that it "is a" video decoder algorithm which allows the codec engine to automatically supply the stubs and skeletons needed
- > Also declared are the external symbols required by xDAIS that identify the algorithms function-table entry points

Technology for Innovators™

TEXAS INSTRUMENTS

Codec Package



Technology for Innovators™

TEXAS INSTRUMENTS

Module Configuration Script

```

/* ===== MPEG2DEC.xs ===== */
var verbose = false;
function getDaramScratchSize(prog)
{
    if (verbose) {
        print("getting DARAM scratch size for " + this.$name
            + " built for the target " + prog.build.target.$name
            + ", running onplatform " + prog.platformName);
    }

    return (39936);
}
function getSaramScratchSize(prog)
{
    if (verbose) {
        print("getting SARAM scratch size for " + this.$name
            + " built for the target " + prog.build.target.$name
            + ", running onplatform " + prog.platformName);
    }
    return (39936);
}

```

```

function getStackSize(prog)
{
    if (verbose) {
        print("getting stack size for " + this.$name
            + " built for the target " + prog.build.target.$name
            + ", running on platform " + prog.platformName);
    }
    return (4096);
}

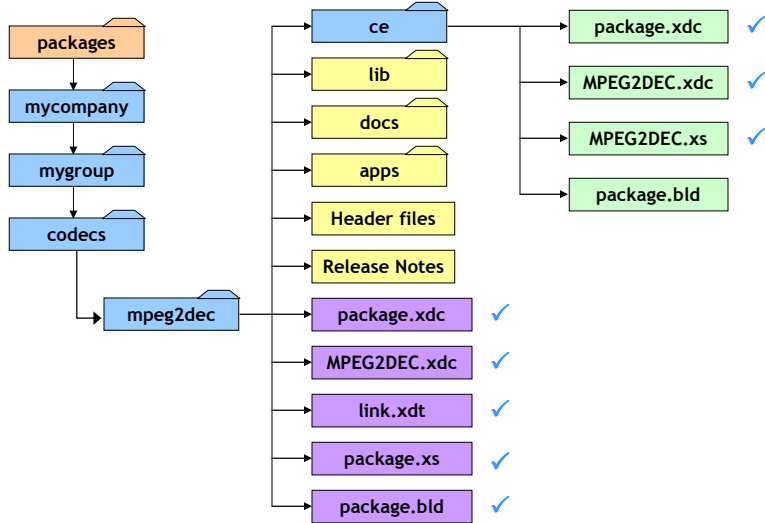
```

- Role: Provides the return values for functions defined within the ICodec interface
- Values can be taken from the corresponding codec datasheet
- Build-time warnings will be generated if these functions are not implemented
- If the return values are 0, the server integrator may not know if insufficient memory has been allocated

Technology for Innovators™

TEXAS INSTRUMENTS

Codec Package



Technology for Innovators™

TEXAS INSTRUMENTS

Package Build Script

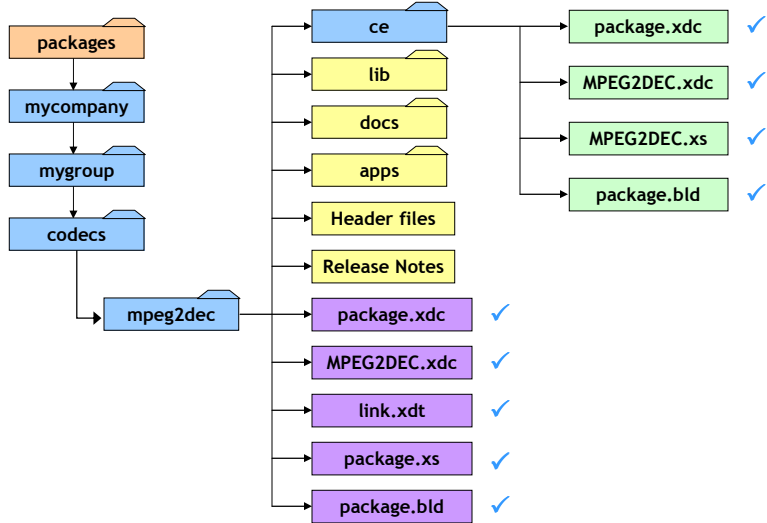
```
/*  
 * ===== package.bld =====  
 */  
  
Pkg.attrs.exportAll = true;
```

- Role: Analogous to the package.bld script for the base package
- Indicate which files to should be included in the package
- Recall: All packages must have a package.xdc and package.bld

Technology for Innovators™

TEXAS INSTRUMENTS

Codec Package



Technology for Innovators™

TEXAS INSTRUMENTS

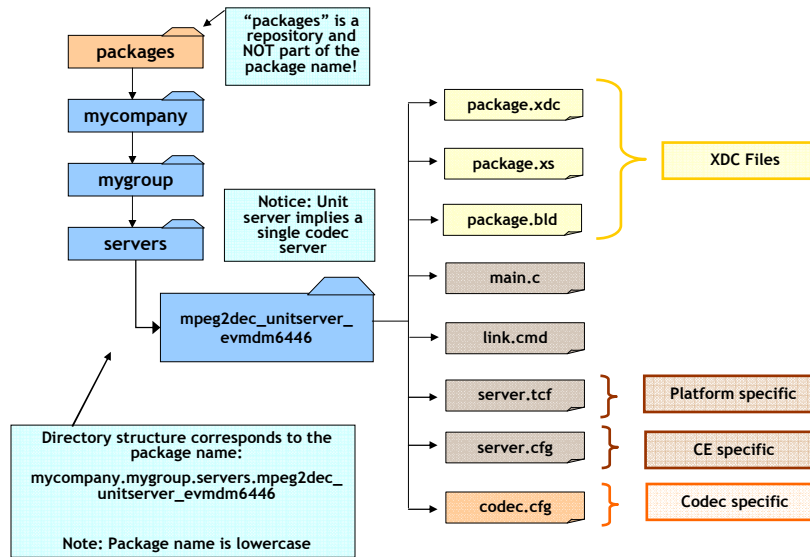
Table Of Contents

- Codec Package Structure
 - “Base” Package
 - Codec Engine Package
- Server Package Structure

Technology for Innovators™

TEXAS INSTRUMENTS

Server Package Structure



Technology for Innovators™

TEXAS INSTRUMENTS

Package Specification File

```

/*!
 * ===== package.xdc =====
 */

package mycompany.mygroup.servers.mpeg2dec_unitserver_evmdm6446 [1,0,0] {
}
    
```

- > Role: defines that this is an XDC package.
- > Package name imposes a directory structure
- ⚠ Convention: Packages are named in lower case and modules start with a capital letter
- ⚠ Common Mistake: Do NOT add requires statements here!

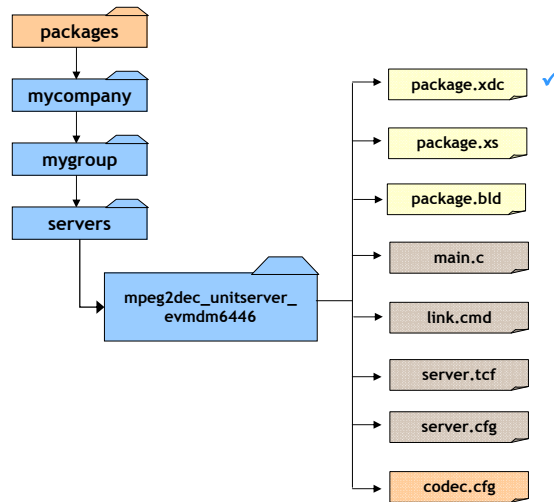
```

requires ti.sdo.ce;
requires ti.sdo.ce.osal;
requires ti.sdo.fc.dskt2;
requires ti.sdo.fc.dman3;
    
```

Technology for Innovators™

TEXAS INSTRUMENTS

Server Package



Technology for Innovators™

TEXAS INSTRUMENTS

Package Configuration Script

```
/*
 * ===== package.xs =====
 */

function validate() {

  /* Code to issue warning if sections are not placed in codec.cfg */
  validate_one_codec( "mycompany.mygroup.codecs.mpeg2dec", "MPEG2DEC" );

}

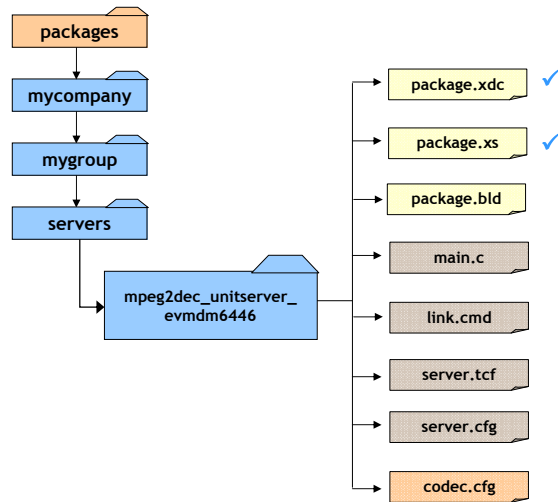
function validate_one_codec( packageName, moduleName ) { ... }
```

- Role: The validate function throws an error if it finds illegal assignments in codec.cfg
- Users may optionally implement other functions written in JavaScript, such as a program integrity check (as in validate) in package.xs

Technology for Innovators™

TEXAS INSTRUMENTS

Server Package



Technology for Innovators™

TEXAS INSTRUMENTS

Package Build Script

```

/* ===== package.bld ===== */

Pkg.attrs.exportAll = true;
Pkg.attrs.exportExe = true;

if (arguments[1]) { var serverName = arguments[1]; }
else {
  var fullName = Pkg.name;
  var serverName =
  fullName.substring(fullName.lastIndexOf('.')+ 1);
}

Pkg.uses = ["ti/bios/include"];

for (var i = 0; i < Build.targets.length; i++) {
  var targ = Build.targets[i];

  /* only build for DSP targets */
  if (targ.os == "Linux") {
    continue;
  }
  print("building for target " + targ.name + "...");
}
  
```

```

Pkg.addExecutable( serverName, targ, targ.platform,
{
  tcopts: "-Dxdc.cfg.check.fatal=false",
  cfgScript: "server.tcf",
  cfgArgs: '{profile: "' + arguments[0] + '"}',
  lopts: "-l link.cmd",
});
addObjects([
  "main.c",
]);
}
  
```

- > Role: Script indicates how to build - but we're primarily interested in adding the DSP executable!
- > Written in JavaScript, following the XDC Build Object Model (which defines a set of objects and functions that let the user describe what to build in a portable, OS-independent way)

Technology for Innovators™

TEXAS INSTRUMENTS

Package Build Script

```

/* ===== package.bld ===== */

Pkg.attrs.exportAll = true;
Pkg.attrs.exportExe = true; ← 1

if (arguments[1]) { var serverName = arguments[1]; }
else {
    var fullName = Pkg.name;
    var serverName =
    fullName.substring(fullName.lastIndexOf('.')+ 1);
}

Pkg.uses = ["ti/bios/include"];

for (var i = 0; i < Build.targets.length; i++) {
    var targ = Build.targets[i];

    /* only build for DSP targets */
    if (targ.os == "Linux") {
        continue;
    }
    print("building for target " + targ.name + "...");

```

```

Pkg.addExecutable( serverName, targ, targ.platform,
{
    tcopts: "-Dxdc.cfg.check.fatal=false", ← 2
    cfgScript: "server.tcf",
    cfgArgs: '{profile: "' + arguments[0] + '"}',
    lopts: "-l link.cmd",
});
addObjects( [
    "main.c",
]);
}

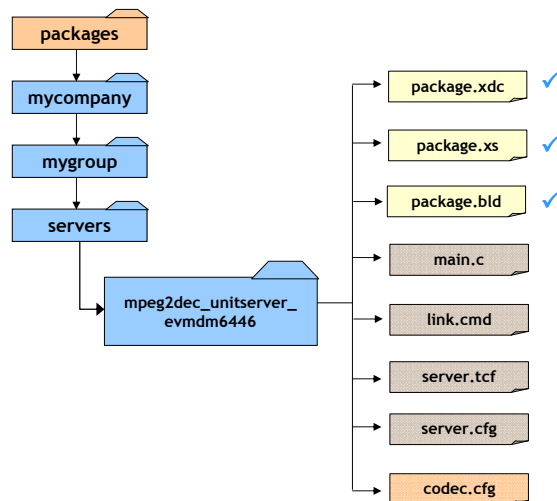
```

- > Command: xdc release
XDCARGS="release myservername -PD ." generates a package with myservername.x64P executable
- > Notice:
 - > 1: needed to include executable
 - > 2: for packages built with < 2.95.01

Technology for Innovators™

TEXAS INSTRUMENTS

Server Package



Technology for Innovators™

TEXAS INSTRUMENTS

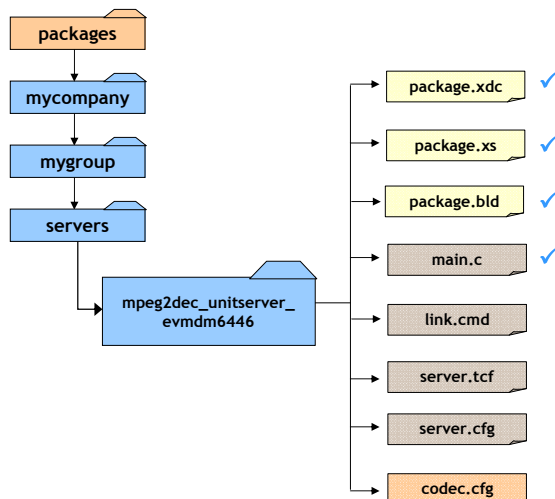
main.c

```
/* ===== main.c ===== */
#include <xdc/std.h>
#include <ti/sdo/ce/CERuntime.h>
#include <ti/sdo/ce/trace/gt.h>
static GT_Mask gtMask = {0,0};

Void main(Int argc, Char *argv[])
{
    CERuntime_init(); ← 1
    GT_init(); ← 2
    GT_create(&gtMask, "codec_unitserver");
    GT_set("codec_unitserver=01234567");
    GT_0trace(gtMask, GT_4CLASS, "main> Welcome to DSP server's main().\n");
}
```

- Role: Contains the applications main() function
- Notice:
 - 1. Initialize Codec Engine
 - 2. Initialize Trace
- DSP/BIOS runs its threads AFTER main() completes, so main() should only contain initialization statements

Server Package



Linker Command File

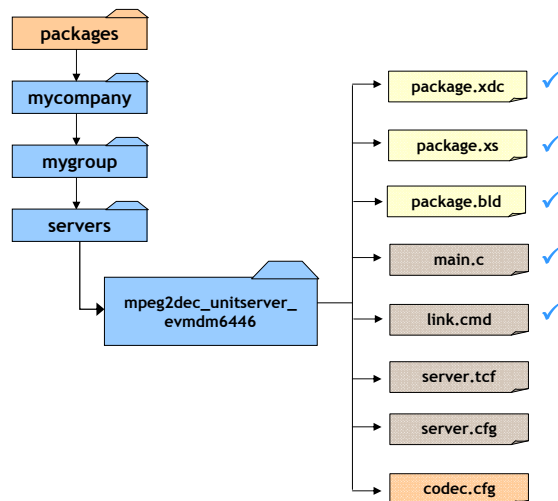
```
/*!  
 * ===== link.cmd =====  
 */  
  
/*  
 * Any additions to the generated linker-command files should be placed here  
 */
```

- > Role: Allows for the specification of any other DSP linker commands
- > Recall that the placement of codec sections is specified in link.xdt in the codec package

Technology for Innovators™

TEXAS INSTRUMENTS

Server Package



Technology for Innovators™

TEXAS INSTRUMENTS

DSP/BIOS Configuration File

```

/*!
 * ===== server.tcf =====
 */
... CODE ...
var mem_ext = [
{
  comment: "DDRALGHEAP: off-chip memory for dynamic
  alghem allocation",
  name: "DDRALGHEAP",
  base: 0x88000000, // 128MB
  len: 0x07A00000, // 122MB
  space: "code/data"
},
{
  comment: "DDR2: off-chip memory for application code and
  data",
  name: "DDR2",
  base: 0x8FA00000, // 250MB
  len: 0x00400000, // 4MB
  space: "code/data"
},
... CODE ...

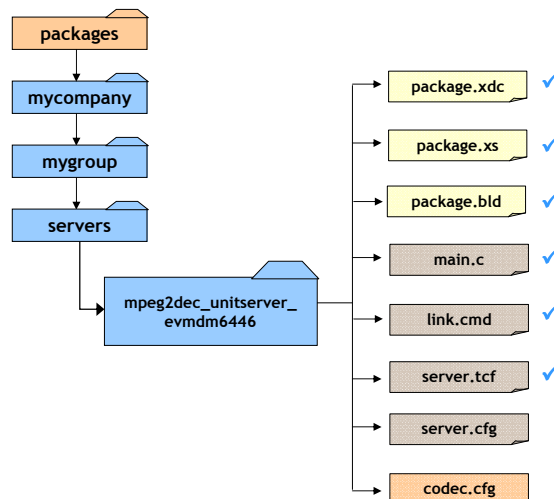
```

- Role: Set up platform specific memory map and attributes (such as clock rate)
- The sections defined in the codec package (such as codeSection, dataSection, udataSection) will be placed in a section defined in server.tcf, such as DDR2
- Default memory map provided with XDC Server Wizard is for the DM6446 SDK
- See Codec Engine Server Integrator's Guide and C6000 DSP/BIOS API Reference Guide for further information on the DSP/BIOS configuration file

Technology for Innovators™

TEXAS INSTRUMENTS

Server Package



Technology for Innovators™

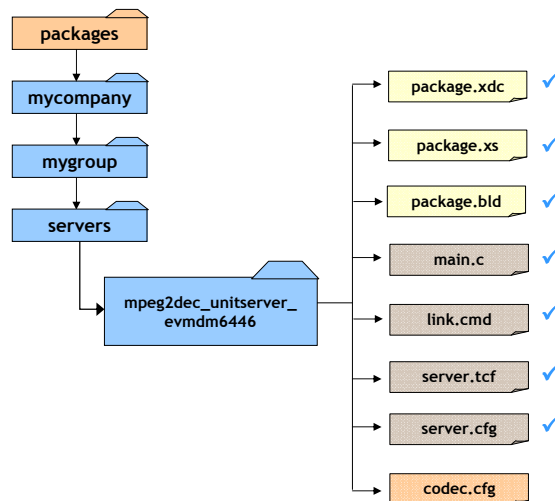
TEXAS INSTRUMENTS

Codec Engine Configuration File

```
/*!
 * ===== server.cfg =====
 */
/* Configures:
   Codec Engine's OSAL
   Server Configuration
   DSKT2 (xDAIS Algorithm memory allocation)
   DMAN3 (DMA manager)
 */
... CODE ...
```

- Role: Configures resources for the server
- See the Framework Components documentation for details on configuring these modules

Server Package



Codec-specific Configuration File

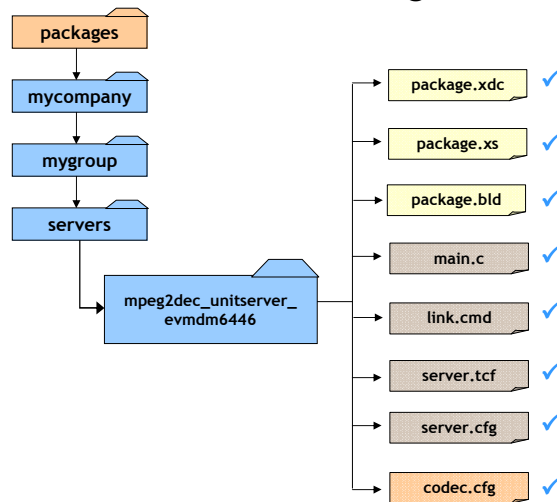
```
/* ===== codec.cfg ===== */  
  
var MPEG2DEC = xdc.useModule('mycompany.mygroup.codecs.mpeg2dec.ce.MPEG2DEC');  
  
// Package Config  
MPEG2DEC.alg.watermark = false;  
MPEG2DEC.alg.codeSection = "DDR2";  
MPEG2DEC.alg.udataSection = "DDR2";  
MPEG2DEC.alg.dataSection = "DDR2";  
  
Server.algs = [  
  {name: "mpeg2dec", mod: MPEG2DEC, threadAttrs: {  
    stackMemId: 0, priority: Server.MINPRI + 2}, groupId: 0,  
  }  
];
```

- Role: Specify codec package to use and its configurable parameters
- These config params are specified in MOD.xdc (in both the base and ce packages) in the codec package
- The sections must be placed in a memory section defined in server.tcf!

Technology for Innovators™

TEXAS INSTRUMENTS

Server Package



Technology for Innovators™

TEXAS INSTRUMENTS

Table Of Contents

- Codec Package Structure
 - “Base” Package
 - Codec Engine Package
- Server Package Structure