

XDC for Multimedia Codecs

Creating Packages and Servers

Katie Roberts-Hoffman
Software Applications
Texas Instruments
katierh@ti.com

Loc Truong
DSP Catalog/EEE
Texas Instruments
l-truong@ti.com

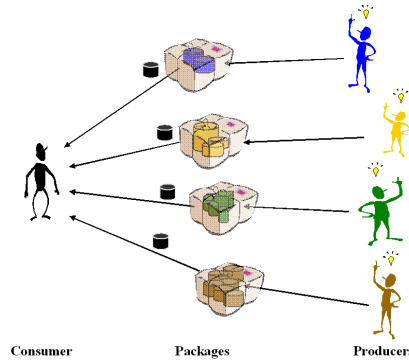
Alan Campbell
Software Applications
Texas Instruments
a-campbell@ti.com

Agenda

- XDC Basics & Terminology
- Creating Codec Packages
- Creating and testing Codec Servers
- Introduction to the RTSC Package Wizards
- Demo: From Codec library to Digital Video Test Bench

What is XDC?

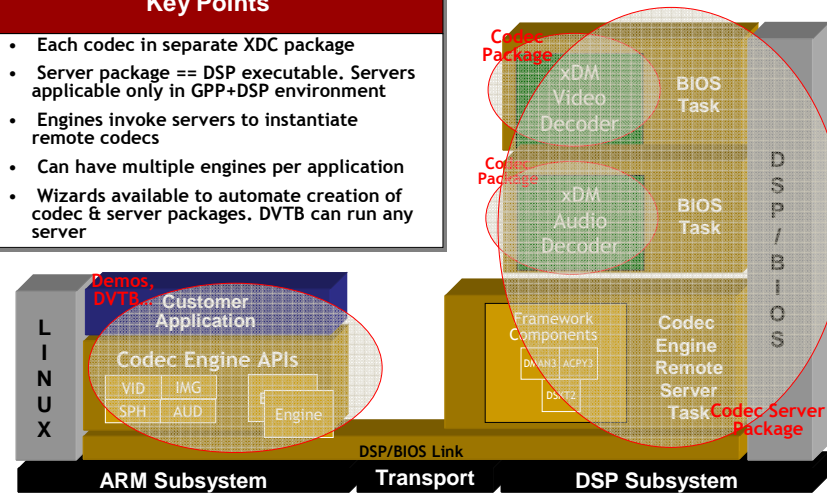
- XDC
 - eXpanDed C
 - Formerly known as "eXpress DSP Components"
- RTSC
 - Real Time Software Components
 - Standardizes the delivery and use of target content
 - Optimized for real-time embedded systems
- Why use RTSC?
 - Configurability and Optimization
 - The consumer can configure content to optimize it for their application
 - Integrity Verification
 - XDC tools verifies that the system is composed of compatible packages
 - Uniformity of management



Codec Packages, Servers, Engines...

Key Points

- Each codec in separate XDC package
- Server package == DSP executable. Servers applicable only in GPP+DSP environment
- Engines invoke servers to instantiate remote codecs
- Can have multiple engines per application
- Wizards available to automate creation of codec & server packages. DVTB can run any server



Who Creates a...

Codec Package?

- Codec Producers
 - Should deliver their codecs as Codec Packages
- Codec consumers who did not receive a Codec Package

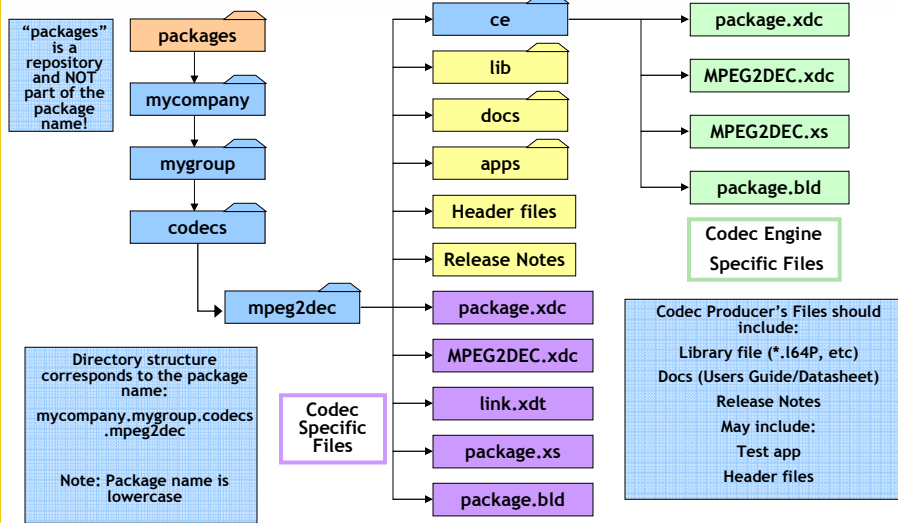
Server Package?

- Codec Producers
- Codec consumers wanting to quickly evaluate a codec via DVTB
- System Integrators wanting to build a *server combo*

Agenda

- XDC Basics & Terminology
- **Creating Codec Packages**
- Creating and testing Codec Servers
- Introduction to the RTSC Package Wizards
- Demo: From Codec library to Digital Video Test Bench

Codec Package Structure



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

Compatibility Key

```

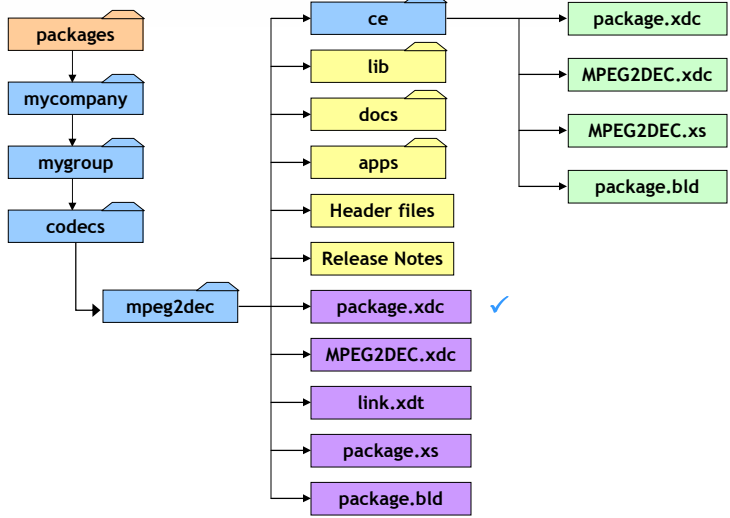
/*!
 * ===== package.xdc =====
 */

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

```

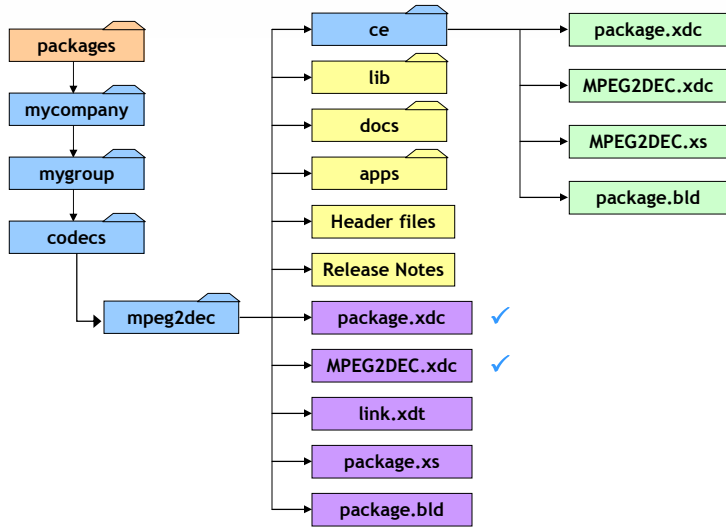
/* ===== MPEG2DEC.xdc =====*/

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.)

Codec Package



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:** Represents the package's optional contributions to the linker command file, including section placement
- > 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`

Linker Template File

```


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

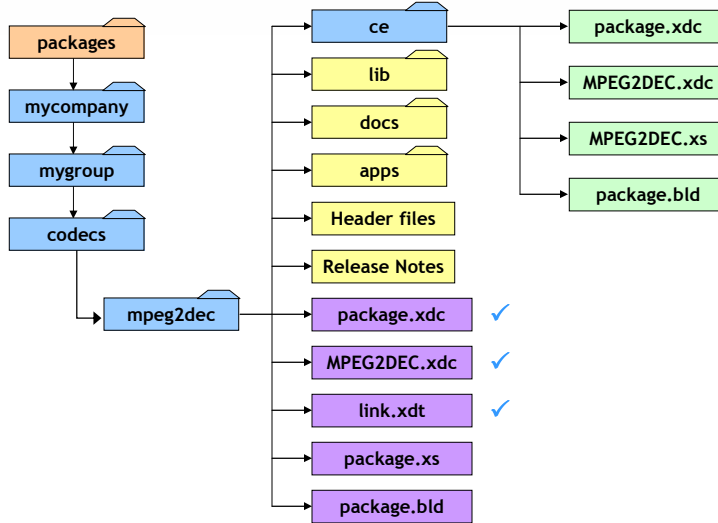
% if (this.G711DEC.codeSection) {
    .text:algNumAlloc > `this.G711DEC.codeSection`
    .text:algInit > `this.G711DEC.codeSection`
    .text:algAlloc > `this.G711DEC.codeSection`
    .text:algFree > `this.G711DEC.codeSection`
    .text:exit > `this.G711DEC.codeSection`
%}
}


```

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

Codec Package



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);
}
    
```

```

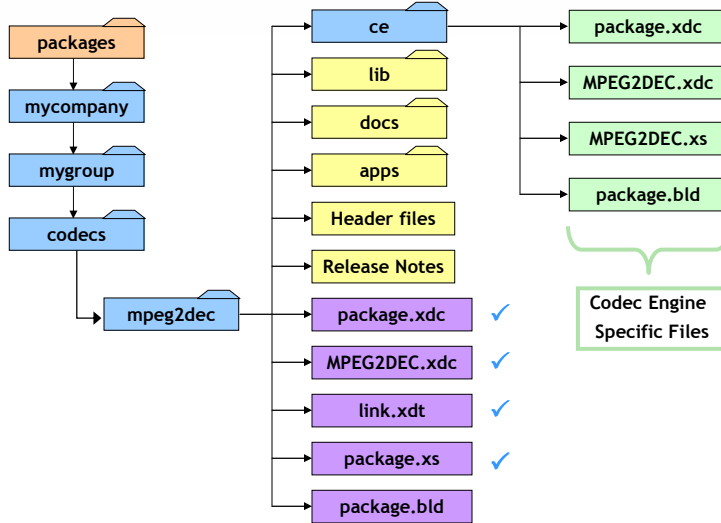
function getSects()
{
    var template = null;

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

    return (template);
}
    
```

- > Role: getLibs() names the actual codec libraries to link with
- > Also implements getSects() to return the linker template, link.xdt
- > May optionally implement other functions written in JavaScript, such as a program integrity check

Codec Package



Package Build Script

```

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

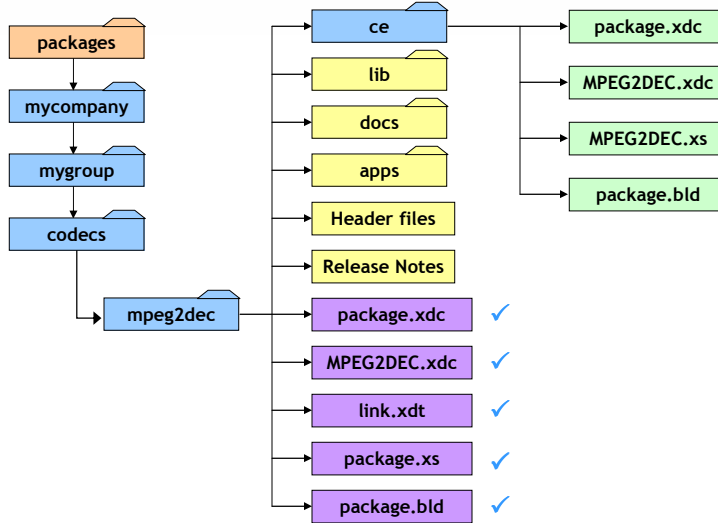
/*
 * ===== package.bld =====
 * Selectively export files and directories
 */
Pkg.otherFiles = [
    'lib',
    'docs',
    'mpeg2dec_1_00_001_ReleaseNotes.pdf',
];
    
```

The usual method

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)

Codec Package



Package Specification File

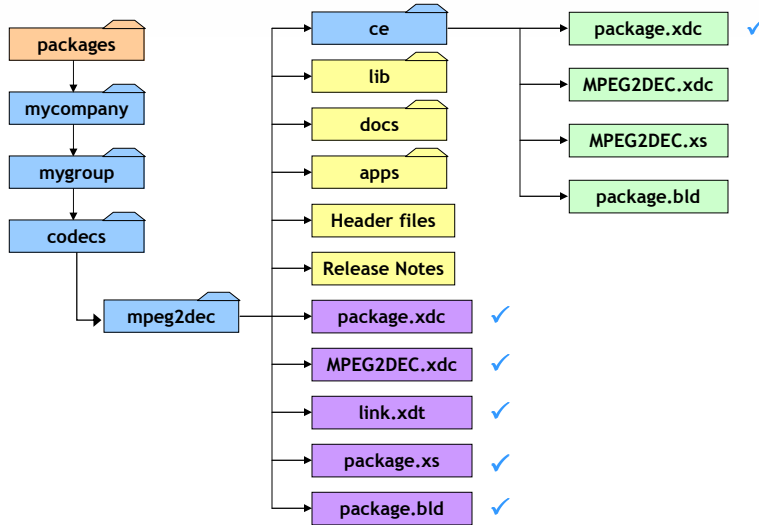
```
requires mycompany.mygroup.codecs.mpeg2dec;

/!
 * ===== 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

Codec Package



Module Specification File

```

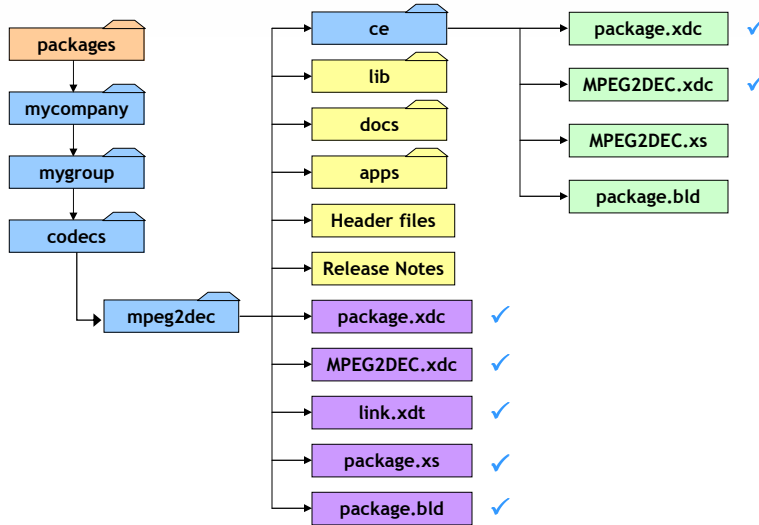
/* ===== MPEG2DEC.xdc ===== */

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 ti.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

Codec Package



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);
}

```

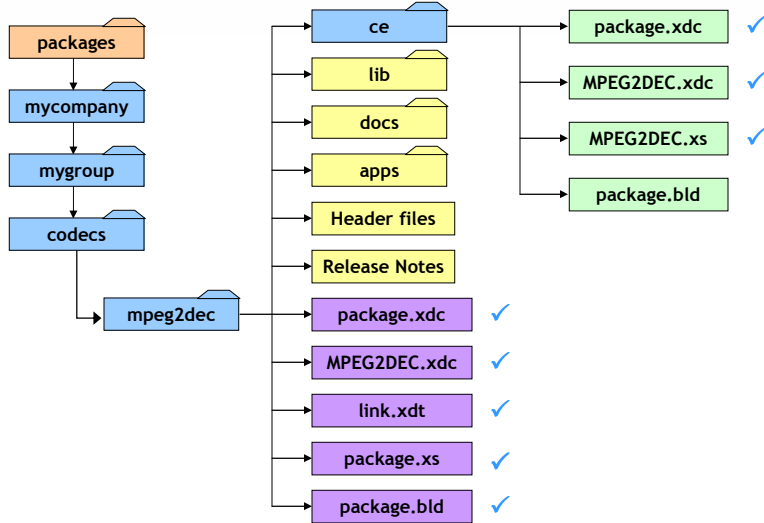
```

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

Codec Package



Package Build Script

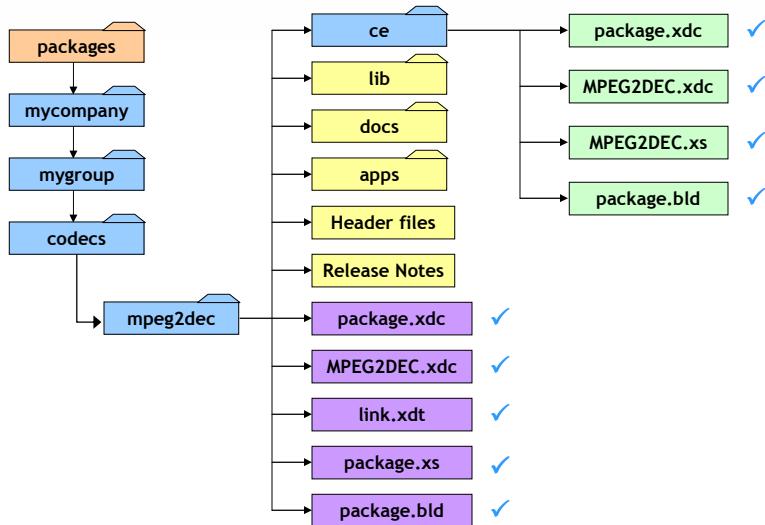
```

/*
 * ===== 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

Codec Package

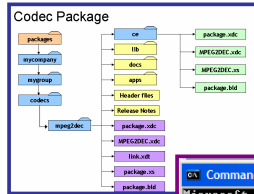


Agenda

- XDC Basics & Terminology
- Creating Codec Packages
- **Creating and testing Codec Servers**
- Introduction to the RTSC Package Wizards
- Demo: From Codec library to Digital Video Test Bench

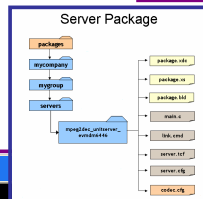
Overview of Server Package Creation Flow

1. Create the files needed for a codec package
2. Create the codec package via "xdc"
3. Create the files needed for a server package
4. Create the DSP executable and server package via "xdc"



```

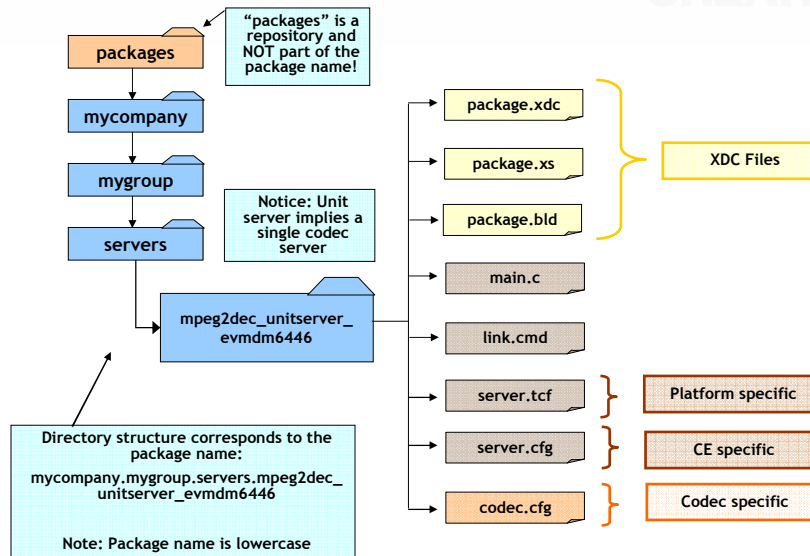
Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
c:\demo\mycompany\mygroup\codecs\mpeg2dec>xdc release
    
```



```

Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
c:\demo\mycompany\mygroup\servers\mpeg2dec_unitserver_evmdm6446>xdc release
    
```

Server Package Structure



Differences between Codec Packages and Server Packages

- Codec Packages are platform *independent* – Server Packages are platform *dependent*
- Server Packages can contain multiple codecs (also known as *Combos*) – Codec Packages only contain a single codec
- Building a Server Package creates a DSP executable
- A codec library is required to create a Codec Package and a Codec Package is needed to create a Server Package

What is DVTB?

- Digital Video Test Bench (DVTB)
 - Provides an easy interface to rapidly test codec servers
 - Available in the DVSDK
 - Supports encode/decode of all VISA classes
 - Supports configuration of audio & video peripherals
 - Scriptable

```
# Specify the codec combo to be used
setp engine name g711dec_engine

# Specify the audio decoder to be used
setp sphdec codec g711dec

# set compandingLaw to aLaw for TI G711DEC
setp sphdec compandingLaw 1

# Trigger the scenario
func sphdec -s /opt/dvevm/data/sounds/davincieffect.g711
```


Using DVTB to test codec servers

- One application, many servers
 - Engine.createFromServer (new in CE 2.00)
 - Configure entire ARM application's engine with a single line of code
 - No need to load codec modules
 - No need to set up DSP memory map info
 - Each server has an auto-generated "Server info file" used by this API

```
#!/
 * ===== server.cfg =====
 */
...
var Engine = xdc.useModule('ti.sdo.ce.Engine');
Engine.createFromServer(
  "g711dec_engine",           // engine name
  "./g711dec_unitserver_evmdm6446.x64P", // server executable
  "ti.sdo.servers.g711dec_unitserver_evmdm6446" // server path
);

Engine.createFromServer(
  "mpeg2dec_engine",
  "./mpeg2dec_unitserver_evmdm6446.x64P",
  "ti.sdo.servers.mpeg2dec_unitserver_evmdm6446"
);
```

DVTB Sample Results

```
root@<ipaddr>:~/opt/ce2x_xdc3x_wizard_fs# ./dvtb-configuro -s dvtb-scripts/mpeg2-play.dvs
setp engine name mpeg2dec_engine
PASS: setp

# Specify the video decoder to be used
setp viddec codec mpeg2dec
PASS: setp

# Trigger the MPEG2 decode+display scenario
func viddec -s /opt/dvevm/data/videos/davincieffect_ntsc.m2v
[DVEVM-ST-LOG]: dvevmStFuncDispatcher.c: Starting Thread for command <viddec>
[DVEVM-ST-LOG]: dvevmStVidPlay.c: Video [Decode][Play] started
[DVEVM-ST-LOG]: dvevmStVidPlay.c: VPBE device </dev/fb/3> initialized for display
[DVEVM-ST-LOG]: dvevmStEngine.c: Engine <mpeg2dec_engine> opened
[DVEVM-ST-LOG]: dvevmStVidPlay.c: Video Decoder <mpeg2dec> initialized
[DVEVM-ST-LOG]: dvevmStVidPlay.c: Frame#, 1, Decoded Frame Size(bytes), 7517, DecTime(us), 10208
[DVEVM-ST-LOG]: dvevmStVidPlay.c: Frame#, 2, Decoded Frame Size(bytes), 288, DecTime(us), 8245
[DVEVM-ST-LOG]: dvevmStVidPlay.c: Frame#, 3, Decoded Frame Size(bytes), 24580, DecTime(us), 8701
[DVEVM-ST-LOG]: dvevmStVidPlay.c: Frame#, 4, Decoded Frame Size(bytes), 25025, DecTime(us), 8839
...
```

- Video output on monitor
- Modify codec parameters via 'getp' as necessary
- For CE 2.0 users: See more debug info via
 - `CE_DEBUG=2 ./dvtb_configuro -s dvtb-scripts/mpeg2-play.dvs`

Agenda

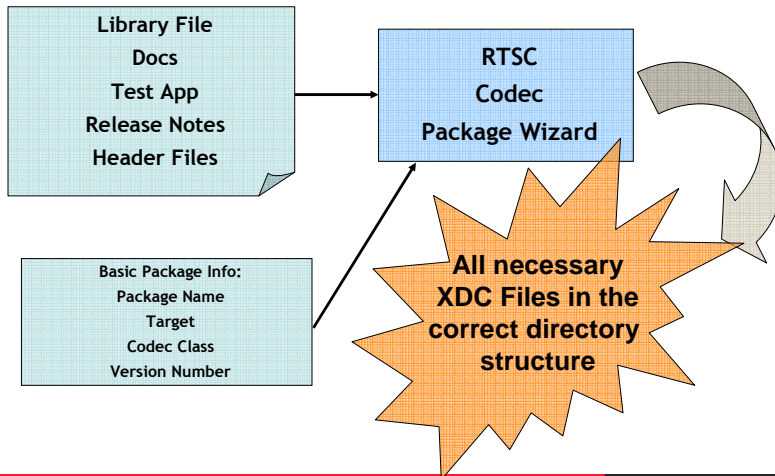
- XDC Basics & Terminology
- Creating Codec Packages
- Creating and testing Codec Servers
- Introduction to the RTSC Package Wizards
- Demo: From Codec library to Digital Video Test Bench

RTSC Package Wizards

https://www-a.ti.com/downloads/sds_support/applications_packages/ceutils/index.htm

- Contains:
 - RTSC Codec Package Wizard
 - RTSC Server Package Wizard
 - Documentation
 - Installation and Use Instructions, FAQ
- Command line and GUI options for both Windows and Linux

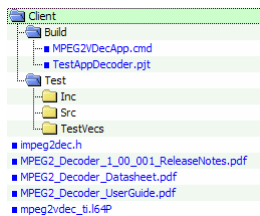
RTSC Codec Package Wizard



Before & After:

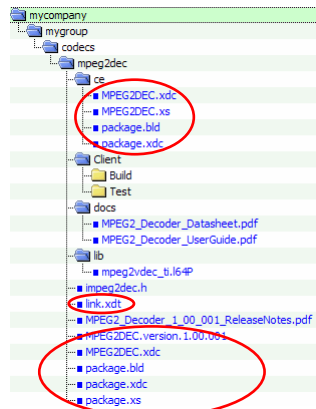
Before:

- library: mpeg2vdec_ti.i64P\
- docs: user guide, datasheet
- release notes
- header file
- test app



After:

- All necessary XDC files
- correct directory structure

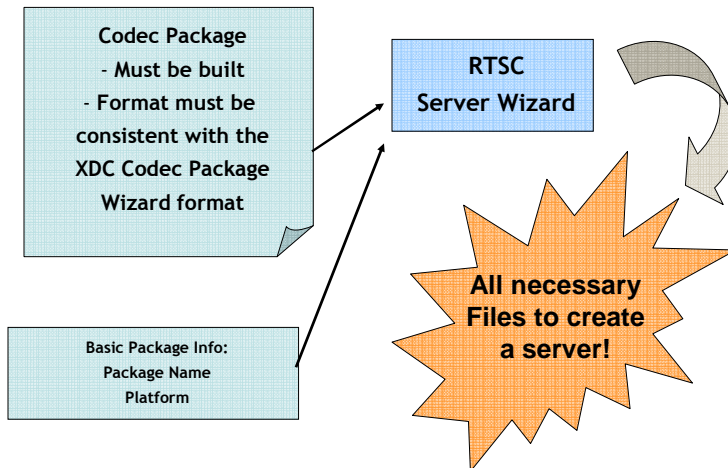


RTSC Codec Package Wizard: IS / IS NOT

- IS
 - Only for codecs with interfaces inheriting from ICodec
 - E.g. IVIDDEC, IAUDENC, IVIDANALYTICS
- DOES
 - Generate basic XDC files for codecs
- CAN
 - Determine stack size requirements*
 - Use information contained within the codec library to determine global variable and section information*
- IS NOT
 - A check for xDAIS compliance
 - A substitute for learning about XDC and packaging
- DOES NOT
 - Execute "xdc release" or generate release tar files
- CANNOT
 - Determine all inputs needed based on the library
 - E.g. scratch size

*For libraries created with TI Code Generation Tools

RTSC Server Package Wizard

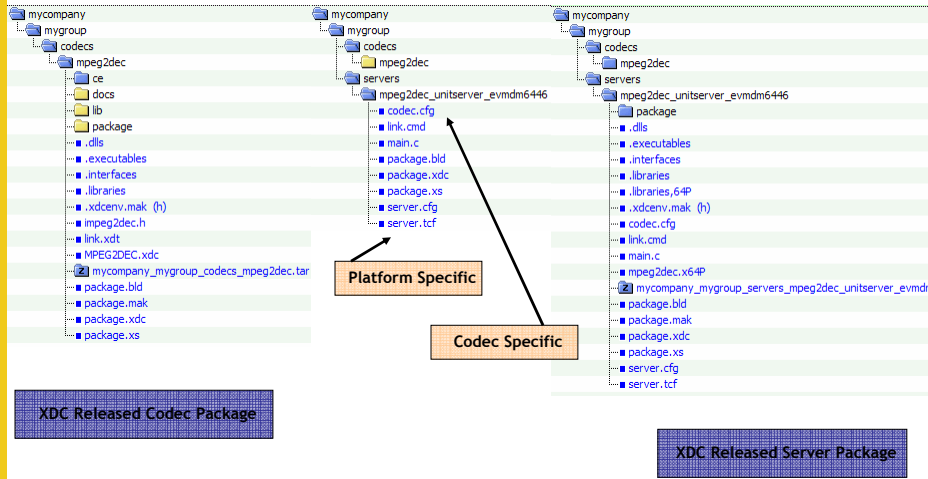


Before & After:

Before:

After Server Wizard:

After XDC release:



RTSC Server Package Wizard: IS / IS NOT

- IS
 - Only for codecs with interfaces inheriting from ICodec
 - E.g. IVIDDEC, IAUDENC, IVIDANALYTICS
- DOES
 - Generate server package files for a single codec server
 - E.g. mpeg2dec_unittest_server.x64P
- CAN
 - Determine config params from the codec package
 - E.g. if the watermark Boolean's default value is set to false in the codec package, then the server configuration file (codeec.cfg) will show the same value
- IS NOT
 - A check for XDAIS compliance
 - A substitute for learning about XDC and packaging
- DOES NOT
 - Execute “xdc release” or generate release tar files
 - Automatically place sections in memory
 - Provide memory maps other than for DVSDK
 - Create Server Combos
 - But the unit test servers are a good place to start from when creating combos

Steps to Create a Codec Package & Server

- Create your codec and associated content
 - Library, documentation, header files, etc.
- Run the RTSC Codec Package Wizard
- Build the codec package via XDC release
- Run the RTSC Server Package Wizard
- Create the DSP executable and server package via XDC release
- Test the Server & distribute packages!

Agenda

- XDC Basics & Terminology
- Creating Codec Packages
- Creating and testing Codec Servers
- Introduction to the RTSC Package Wizards
- Demo: From Codec library to Digital Video Test Bench

Demo

- Objective
 - Create AAC Decode Codec Package and Server Package using RTSC Package Wizards
 - Run AAC Decode on DM6446 using DVTB
- Set-up
 - Red Hat 4 Linux
 - BIOS v5.31
 - Codec Engine v2.00
 - XDC Tools v3.00
 - RTSC Package Wizards (ceutils v1.03)
 - MPEG2 Decode Library
 - DM6446 DVSDK v1.20

Resources

- Downloads
 - XDC
 - https://www-a.ti.com/downloads/sds_support/targetcontent/rtsc/index.html
 - XDAIS
 - https://www-a.ti.com/downloads/sds_support/targetcontent/xdaais/index.html
 - Codec Engine
 - https://www-a.ti.com/downloads/sds_support/targetcontent/CE/index.html
 - RTSC Package Wizards
 - https://www-a.ti.com/downloads/sds_support/applications_packages/ceutils/index.htm
- Documentation
 - XDC
 - XDC Getting Started Guide (available in /XDC_x_xx/docs)
 - XDC User Guide
 - RTSC Package Wizards Wiki Page
 - http://wiki.davincidsp.com/index.php?title=RTSC_Codec_And_Server_Package_Wizards
 - Codec Engine
 - Codec Engine Server Integrator's Guide (SPRUED5A)
 - Codec Engine Algorithm Creator User's Guide (SPRUED6B)
 - DVTB
 - DVTB User Guide (available in /DVSDK_x_xx/dvtb_vx_yy_zz/docs)

Summary

- Distributing codecs in XDC packages with servers allows consumers to easily test and integrate codecs into their application
- TI provides tools to assist codec producers creating codec packages and servers

XDC for Multimedia Codecs

Creating Packages and Servers

Katie Roberts-Hoffman
Software Applications
Texas Instruments
katierh@ti.com

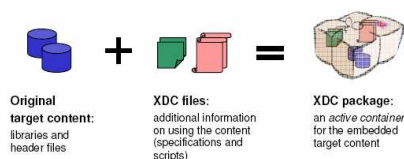
Loc Truong
DSP Catalog/EEE
Texas Instruments
l-truong@ti.com

Backup Slides

- Terminology
- Detailed server package format
- RTSC Codec Package Wizard Installation and Instructions
- RTSC Server Package Wizard Installation and Instructions

Terminology

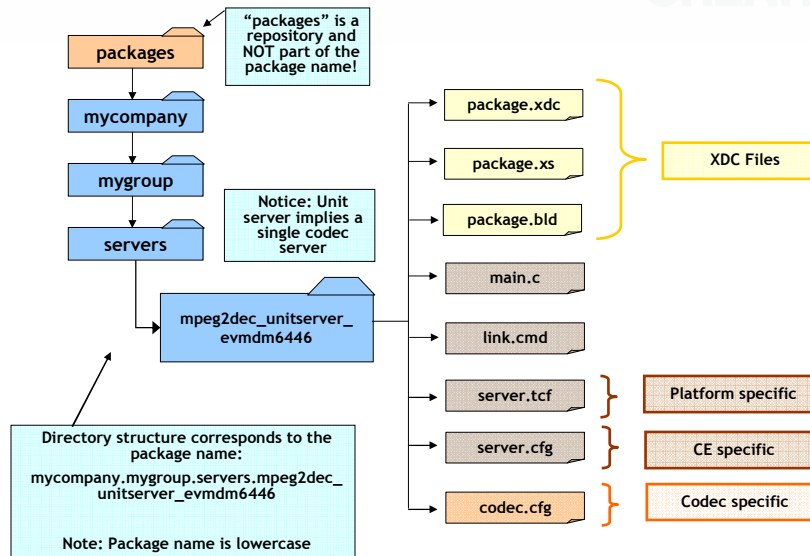
- Package
 - Software component's unit of delivery
 - Has its own global namespace
 - Well defined interfaces in JavaScript
- Package Name
 - Must be globally unique and in lower-case
 - Start your package name with your company name
 - Common forms: mycompany.mygroup.codecs.codec_name
 - E.g. ti.sdo.codecs.h264dec, ti.xdais
- Repository
 - A directory that contains packages
- XDCPATH
 - Environment variable that specifies package repositories
- XDCScript
 - XDC extends JavaScript with type-safety and object models suitable to configure and build embedded programs
- Codec Engine
 - A light-threaded Digital Media processing layer with Video, Imaging, Speech and Audio as its main classes
- Engine
 - A Codec Engine instantiation of a Digital Media processing subsystem



Terminology

- Codec Package
 - XDC package containing a codec library
 - May integrate into Codec Engine
- Server Package
 - Integrates codecs, framework components, and system code
 - Sometimes referred to (casually) as "codec combos"
 - Servers only exist in GPP+DSP configurations!
 - "Engine" invokes codec servers to instantiate **remote** codecs
 - Unlike codecs, servers are on a per platform basis
 - Memory maps are different between DM6446 and DM6467
 - GPP uses VISA APIs to transparently invoke remote codecs on the DSP
- Digital Video Test Bench (DVTB)
 - Provides an easy interface to rapidly test codec servers
 - Available in the DVSDK
 - Supports encode/decode for all VISA classes
 - Supports configuration of audio/video peripherals
 - Scriptable

Server Package Structure



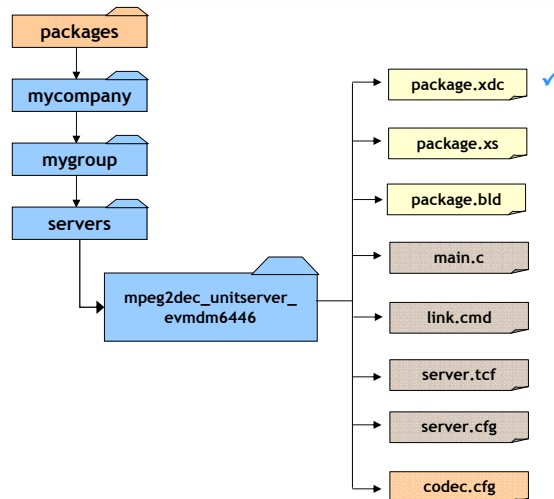
Package Specification File

```
/*!  
 * ===== 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;
```

Server Package



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" );

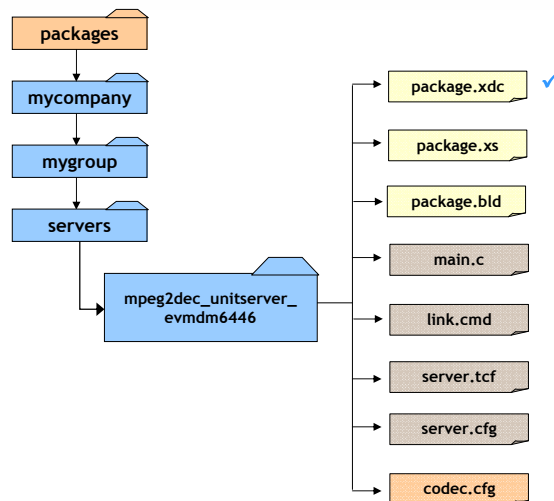
}

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

Server Package



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];

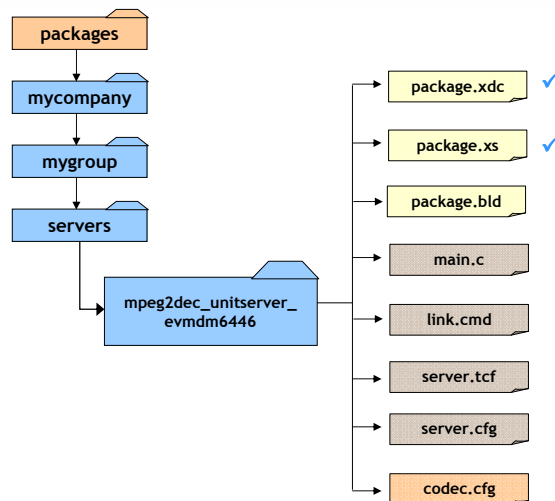
    /* only build for DSP targets */
    if (targ.os == "Linux") {
        continue;
    }
    print("building for target " + targ.name + "...");
}
    
```

```

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)

Server Package



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];

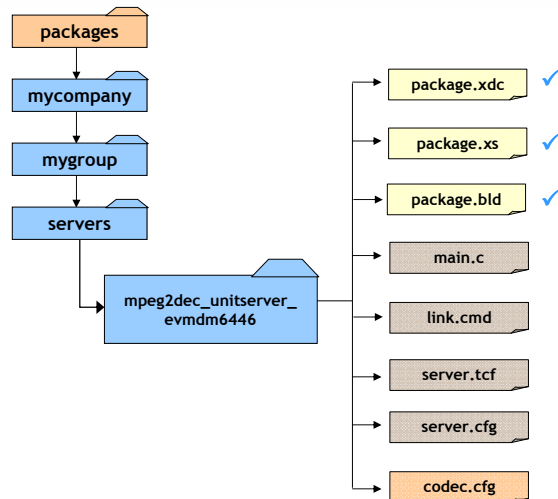
    /* only build for DSP targets */
    if (targ.os == "Linux") {
        continue;
    }
    print("building for target " + targ.name + "...");
}
    
```

```

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

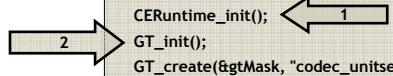
Server Package



```

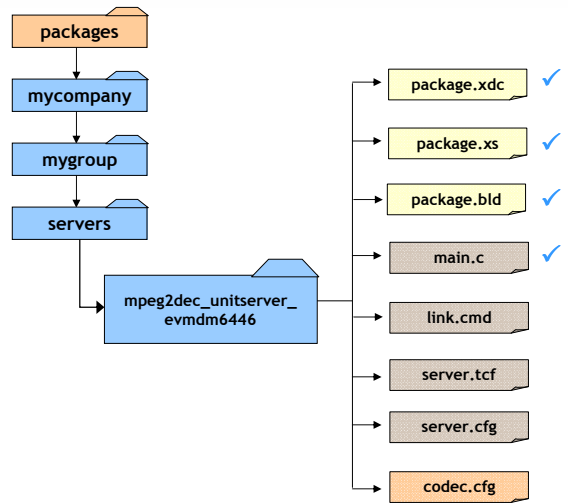
/* ===== main.c ===== */
#include <xdc/std.h>
#include <ti/sdo/ce/CERuntime.h>
#include <ti/sdo/ce/trace/gt.h>
static GT_Mask gtMask = {0,0};

Void main(Int argc, Char *argv[])
{
    CERuntime_init();
    GT_init();
    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

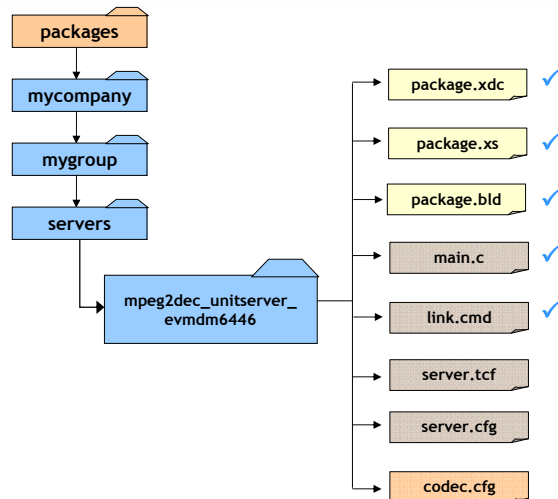
```

/*!
 * ===== 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

Server Package



DSP/BIOS Configuration File

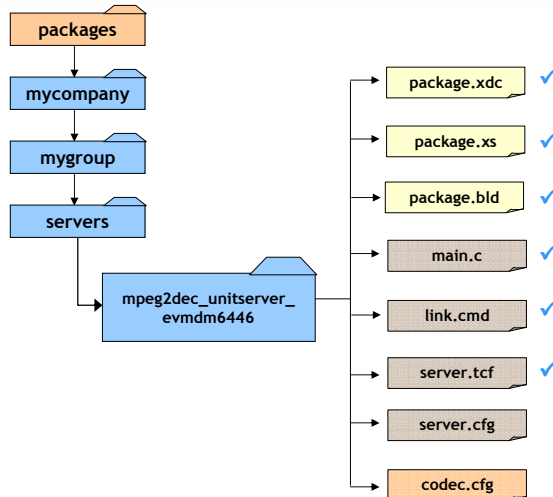
```

/*!
 * ===== server.tcf =====
 */
... CODE ...
var mem_ext = [
{
  comment: "DDRALGHEAP: off-chip memory for dynamic
  almem 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

Server Package



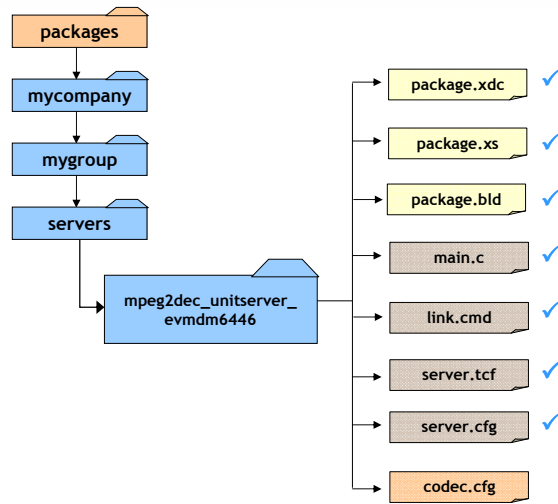
Codec Engine Configuration File

```

/*!
 * ===== 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 ===== */

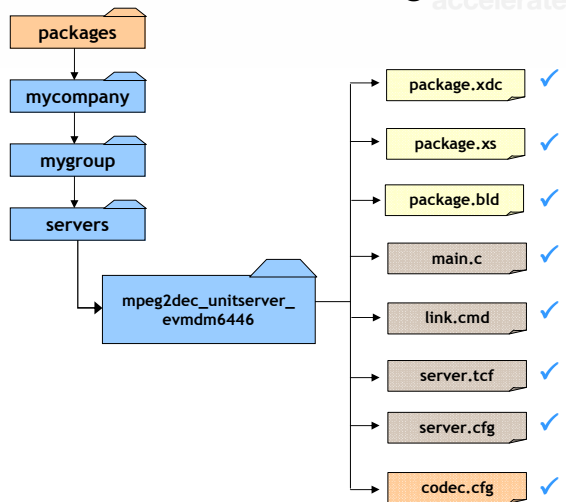
var MPEG2DEC = xdc.useModule('mycompany.mygroup.codecs.mpeg2dec.ce.MPEG2DEC');

// 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!

Server Package



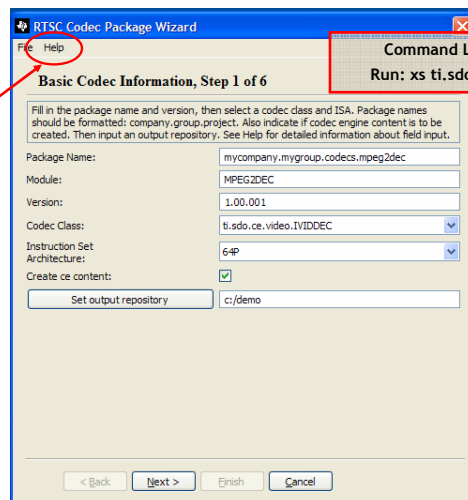
Now that I have a server package, how do I test it?

RTSC Codec Package Wizard: Installation

- Download RTSC Package Wizards
 - https://www-a.ti.com/downloads/sds_support/applications_packages/ceutils/index.htm
- Download CG_XML 1.20
 - https://www-a.ti.com/downloads/sds_support/applications_packages/cg_xml/index.htm
- Download XDC 2.95 or greater, Codec Engine 1.20 or greater
- Set your XDCPATH
 - /ceutils_1_xx/packages
 - /codec_engine_1_20/packages
 - /codec_engine_1_20/ceutils/packages OR /xdais_5_21/packages
- Start the wizard
 - /xdc_x_xx/xs ti.sdo.codecutils.genpackage
- See Wizard help
 - /xdc_x_xx/xs ti.sdo.codecutils.genpackage --help

RTSC Codec Package Wizard: Instructions

Unsure of what to enter?



RTSC Codec Package Wizard

File Help

Basic Codec Information, Step 1 of 6

Fill in the package name and version, then select a codec class and ISA. Package names should be formatted: company.group.project. Also indicate if codec engine content is to be created. Then input an output repository. See Help for detailed information about field input.

Package Name: mycompany.mygroup.codecs.mpeg2dec

Module: MPEG2DEC

Version: 1.00.001

Codec Class: ti.sdo.ce.video.IVIDDEC

Instruction Set: 64P

Architecture:

Create ce content:

Set output repository: c:/demo

< Back Next > Finish Cancel

Command Line Version is also available!
Run: `xs ti.sdo.codecutils.genpackage --help`

Don't forget to see the Wizard FAQ Or Wizard Wiki!

RTSC Codec Package Wizard: XDC Release

- After running the wizard, browse to “base” directory of the output files
 - E.g. If the package name is ti.sdo.codecs.mpeg2dec and the output repository is /my_codec_repo, then change directories to /my_codec_repo/ti/sdo/codecs/mpeg2dec
- Run “xdc release –PR .”
 - Don’t forget the dot
- XDC release generates a directory “package” in the “base” and “ce” directories as well as a tarball for each package
 - These tarballs can be distributed!
- Alternatively, create a single tarball for distribution using the instructions outlined in the wizard readme/FAQ

RTSC Server Package Wizard: Installation

- Download RTSC Package Wizards
- Set your XDCPATH
 - /ceutils_1_02/packages
 - /codec_engine_x_xx/packages
 - /codec_engine_x_x/cetools/packages OR /xdais_x_xx/packages
 - /bios_x_xx/packages
- Start the wizard
 - /xdc_x_xx/xs ti.sdo.codecutils.genserver
- See Wizard help
 - /xdc_x_xx/xs ti.sdo.codecutils.genserver --help

RTSC Server Package Wizard: Instructions

Command Line Version is also available!
Run: `xs ti.sdo.codecutils.genserver --help`

Codec Package Name

Default Output Repository is the Codec Package's Repository!
Change the Output Repository by selecting this button!

Notice this Codec Package has module MPEG2DEC

Technology for Innovators™

TEXAS INSTRUMENTS

RTSC Server Package Wizard: XDC Release

- Browse to the “base” directory
 - E.g.
`/my_server_repo/mycompany/mygroup/codecs/mpeg2dec_unittest_server`
- Edit `codec.fg`
 - Specify the placement of sections into memory
 - E.g. Change `MPEG2DEC.alg.codeSection = undefined;`
 - To: `MPEG2DEC.alg.codeSection = "DDR2";`
- Verify your `XDCPATH` includes:
 - Codec Engine/packages, Codec Engine/cetools/packages OR
XDAIS/packages, BIOS/packages, and repository containing your codec package used in the server
- Run `“xdc release -PD .”` OR `“xdc release XDCARGS=“release myservername” -PD .”` from the base directory
- Test the server!

Technology for Innovators™

TEXAS INSTRUMENTS