So hello, everyone, and welcome to this video.

In this video, we will try to understand how to do SSIs project deployment or how to deploy SSIs project

into production server.

So in order to understand deployment in SSIs, what I'll do is let me go and create a very simple SSIs

package which will copy files from one folder to other folder.

Let us try to keep this program as simple as possible because our main concentration here is to understand

deployment and not write some complex program.

Okay, so I'm going to go and create a very simple package which will go and copy files from this location

one folder.

So if you see this in my C drive, I have a location, one folder, and in this location, one folder

I have a text file called as some text file.

Right?

So I'm going to go and create a package which will copy this, some text file from location one to location

two.

Okay.

So in order to do that, so here's a very simple SSIs project I have created.

If you see the Solution Explorer, you can see that this package or this program has a very simple file

here called as file copy or DTS file.

So in this file, let us go ahead and put the components which will copy the file from location one

to location two now in SSIs to copy the file from one location to other location.

We have a very great component here called as the file system Task.

This file system task component helps us to create files, copy files, delete files.

Right?

So I'm going to go and drag and drop this file system task component on the x file here.

So let us go into this file system task component here.

Let us right click say edit.

So let us go ahead and first create a source connection.

So we want to copy from location one to location two, right?

So I'm going to say, okay, a new connection and this connection would be a file connection, right?

So I'm going to say yes, it's a existing file, which is located in the C drive in the location one

folder.

So this is this some text file.

And I would like to copy this, some text file from location one to the destination folder.

So I'm going to go and create a destination here as well.

So I'll say if the file is not existing, please create a file.

Right?

So I'll say the destination is location two and the file name we will give at this moment as destination

dot txt.

So.

So that will be the file which will get created.

Okay.

So you can see here I have created.

So if you just go to the connection manager tab here.

So this is actually my source.

Let me go and put a proper name here.

You can see it has put the names with the text file names, which does not look good.

So I'm going to say this is my source connection and this is my destination connection.

Right.

Good.

So.

We have this file system task here which is copying from the source that is from C program files.

C location one I'm sorry, C location one to the destination C location two.

Great.

Let us go and run this program to just ensure that everything is running so great.

You can see that the program has ran successfully.

Let us go and just check.

You can see that in the location.

Two he has created the destination text file as well.

And if I open it, everything is there.

Right?

Good.

So I'm going to go and delete this destination file at this moment.

So this program is working.

Good.

So now we would like to go and deploy this program into SSIs, right?

Or we would like to do deployment of this program into a production server.

Now, before we do the deployment on the server, let us first try to understand that what things gets

deployed, what things actually gets deployed onto the production server.

So in order to understand that, we have to understand, you know, how the solution of this project

is structured.

So you can see here the solution of the project is structured in this hierarchy.

At the top we have a solution file and the solution file will have a lot of project file.

For example, you can see that I have a solution file here called as file copy program.

And in that we have one solution called as file copy program again.

Right.

And then I can have one more project here must be file copy program two or must be some other program.

Right?

So you have solution and then solution has projects and then project has packages.

Again, I'll repeat, we have solution.

Solution has projects and projects has packages and every project has parameters.

You can see there is a file here called as project dot params.

So solution then projects and then project has packages that is dot exe file and parameters.

So this is how your SSIs project structure looks like, right?

So you have solution project.

Project has packages which is nothing but x file and every project has params file.

So solution can have multiple projects as well.

You can see this is one project and then there is second project.

But every project will have the same structure that is package and parameters.

Now you can deploy SSIs packages or SSIs projects in production server in two ways.

One is termed as the project deployment mode and the second one is termed as the package deployment

mode.

Now the project deployment, you know, it has been introduced from RS 2012 and it is the most preferred

way of doing deployment while package deployment was there till 2008.

And I will talk about the differences between both of them in the later part of the video.

But for now my main focus will be on project deployment.

But before I move ahead, let me explain you the difference between project deployment and package deployment.

In project deployment mode, the complete project gets deployed in one go.

Remember I said that project has packages.

So for example, you can have one project and it can have lot of X files, right?

So you would like to go and probably deploy all these files in one go.

Then project deployment is the way to go.

But let's say, you know, later on, later on, you know, when you have done the deployment, some

developers go and they modify 1 or 2 package files, then you don't want to do the full deployment again,

right?

So at that time you can go and do the package deployment.

That means you can go and take individual X files and deploy them.

So project deployment means you take the complete project and deploy it.

While package deployment means you can you can deploy individual X file.

So in this video, our main concentration would be to understand project deployment more because you

know, that is a new way of deploying and that is a proper way of deploying as well.

Now SSIs projects, you know, can be deployed in two different locations, you know, depending on

what you prefer.

The first one is inside SQL Server.

So if you are doing project deployment, then you can go and deploy your complete project into SQL Server.

You know where we have a nice database here called as the Sis.

DB So if you are doing project deployment, you can go and deploy your complete project.

One is into SQL Server, the other one is inside the Sis services.

You know, you can go and deploy your project either into Msdb or you can go and deploy it either into

your file system.

So there are three places, you know where you can go and deploy your SSIs project depending on what

your preference is.

First one is into SQL Server, that is into Sis DB.

The second one is into file system and the last one is into Msdb.

So let me quickly go and switch back to my project here.

So very quickly if you see here.

You can see that I have a SQL Server and I have SSIs here, right?

So the first thing is you can if you're doing project deployment, you can go and deploy your project

into this SSIs if you're doing project deployment or else you can go and deploy it here.

Inside either msdb or either into file system.

So again, there are three places where you can go and deploy your project into SQL Server, SDB, into

Msdb or into file system.

Also would like to make one important point here.

You can do package deployment as well as file deployment into SDB.

That means into SQL Server, but you can only do package deployment into SQL Server integration services

that is in file system and Msdb.

So in file system and Msdb, you can only do package deployment.

Again, repeat in msdb and file system, you can only do package deployment.

That means you can deploy at the file level at X level.

While in SQL Server you can do both.

You can do project deployment.

That means you can deploy all the packages in one go, or you can also go and import package by package

and do the deployment.

In this video we will be more concentrating on project deployment because from CES 2012, the most preferred

way of doing deploying is at the project level.

Okay.

So let me let us go and open our solution here so you can see that I have opened the Visual Studio here,

right?

And let us go and rebuild our full solution once.

So let us go and do a rebuild so everything build everything nice right now to do project deployment.

What SES does is it creates a set of file.

So if you go to the if you right click on the project or you can right click on the solution if you

wish.

And over here you can see there is a menu saying Open folder in Windows Explorer.

So let's do that and let us go inside this file copy program folder.

And let us go in the bin directory and let us go inside this development.

Now, you can see over here, uh, there is one file created here if you see very quickly.

So, so this file extension is integration services project deployment file.

And if you go and see the extension of this file.

So if you right click on this and if you go to properties here, the extension of this file is a spec

file.

You can see this dot spec name here, right?

So this is nothing but your setup file.

So if you take this file and if you run it, it will actually go and start installing your packages

inside SQL Server.

So let me go and double click and let me start installing the package.

So I'm going to go and double click this spec file.

So wherever you want to do deployment on whichever production server or whatever server you want to

do deployment, copy this setup, file this dot spec extension, take it over there and and run it.

Once you run it, you will see that he will actually start up a deployment wizard here.

So let us do a next.

The next thing he says is that so you want to do project deployment or you want to do individual, uh,

file deployment, right?

Means a catalog deployment.

So I'll say, no, I want to do project deployment.

And yes, the, the spec file is this.

So take whatever is there in this spec file and deploy it on my server.

I'll do a next again.

He goes and he validates the project.

Right now the next thing is we have to go and browse to our SQL Server.

Remember we are going to go and install into SQL Server first.

So when I say I want to install into SQL Server, I mean I want to install into this SSIs DB over here,

right?

So I'm going to go and give the SQL Server name here.

So that is this.

And I'll say, okay, so once you have specified the server name, the next thing what you need to specify

is the path.

Now you must be wondering what is what exactly is this path here?

Because the server name is more than enough, you know, and probably this x file should just get deployed

into a database.

Right?

So why this path?

Remember, at the end of the day, what we deploy in SSIs is a dot x file.

So in other words, if you look at the least level of deployment or the least unit of deployment, I

will say it is the X file.

So at the end of the day, if it is a file, then it needs a folder, right?

So what I'll do is let us go back to our, uh, SQL Server here and let us create a folder here.

So I'm going to go and create a folder here.

You can see I have right clicked on that SSIs DB and I'll create a folder saying my package or something.

So I'll say this is my package, right?

So you can see there my package is created.

Now inside this package we will go and deploy our projects, right?

So the folder is created.

So let us go back again here and I will say browse and I will say yes, go and install in this folder

which I've recently created.

I'll say, okay, I'll do a next.

This is all fine and deploy.

So you can see everything is done here.

It has loaded the project, connected to the server, did all the verification, and then it has deployed

the project.

So if I go and close this, let me quickly close this and let me do a refresh here.

So I'm going to go and refresh over here and I should see my project.

So my project name, remember it is file copy program, and inside that you can see my package here,

file copy dot exe.

Right.

Great.

So now the next thing is you would like to go and run the package, but before you go and run or execute

the package, you would like to go and configure the package, right?

So in other words, for example, if you remember we this project, what it does actually, it actually

copies from one location to other location, right?

So probably you would like to go and configure the location saying that, okay, it is not see location

one, It is probably see location two.

Right?

So let's first configure it.

So let's right click on it.

Let's click on configure.

So you can see this is our UI, which will help us to do configuration.

The first thing, what we would like to configure is the connection managers.

Remember we had two connection managers created, one which actually has the destination file, right

location, and the other one which has the source file location.

So in case you would like to go and configure this for now, for example, for now you can see the location

is see location one, some text, right?

The source location is see location one, some text or TXT and the destination by default is see location

two.

Right.

What I'll do is let us go and change this destination.

Right?

So let me go back here and let me create one more folder here and I'll term I'll, I'll name that folder

as location three.

Okay.

So what we'll do is, you know, rather than creating copying the file in location two, I would like

to go and change this configuration to location three.

So I'm going to go and click on this and I'll say, don't use the default value from the package, edit

the value.

So the value is now control C control V it is location three.

I'll say okay.

And I'll say, okay.

So you can see you can go and change your configuration, you know, in your production server as well.

So remember to configure you need to go and right click on your package very quickly.

You need to go and right click on your package and say configure, right?

So with that, you know, you can always go and change your connection Managers default value.

So what will happen now is it will actually take this file from C location one, some text dot txt and

copy it to this destination here c location three destination dot txt.

Right.

So let us go and run this package.

So to run this package right click execute.

And before running the package, if you wish, you can again go and change this value here as well.

Right?

So in case you want to go and make changes while running.

But remember, when you make changes while running, right, you know, these values are not stored.

So when you make changes to these connection managers during configure, the values are actually stored

in a production server.

Right?

I will talk about this parameters and advanced thing later on.

So leave that thing at this moment.

And let me just say, okay.

And I hope that the file will be copied to location three, right?

So you can see the package has executed and once your package is executed, it also provides you a report,

a report which will help you to view if your package has executed successfully or if it has errors.

In case it has errors, it will give you a display here as well.

So I'm going to say yes, I want to see the report also.

So I've said yes here.

So you can see that there are my package is executed and you can see the report as well saying succeeded.

Right.

So let us quickly go to our location three here to just check if everything has been copied here.

You can see in my C drive location three destination text file has been copied.

Great.

So this is great.

I'm going to go and delete this file for now.

This is great.

But now I would like to do the following.

I would like to go and run this package as a task.

As a task which keeps running continuously.

Must be every hour or every week and copies the file from that folder source folder to the destination

folder.

So in order to go and create a task, a background running task, what you can do is you can give this

package to your SQL Server agent.

So let me go and start my SQL Server agent.

I'm going to go right click and start this SQL Server agent.

Now in case you are new to SQL Server Agent, right.

What My suggestion is to go and see the video in SQL Server Questions and Answers Video series where

we have explained what exactly SQL Server agent does.

But for now you can think about this SQL Server agent actually, you know, runs background jobs continuously,

you know, like a task, you know, so that, you know, you don't have to do things manually, right?

So what we have to do here now is we have to go and create a job here.

So I would like to go and create a job which will actually go and run this package file copy continuously.

So let us right click on this jobs folder here, say new job.

So the first thing is we need to go and specify a name here.

So I'll say yes, the name is file copy job, right?

And it will run under administrator.

So that is fine.

It is enabled.

That is also fine steps.

So I'll say.

Basically a.

So the step means you know exactly what you want to run.

So I'll say, okay, I clicked on new Step here.

You can see I went here and I clicked on this new step here and the So this step is actually nothing

but what exactly you want to run.

So I want to say is I want to run a SSIs package.

Can you see here SQL Server Integration Service package.

So I want to run a package and I'll say this is a file copy package.

So the next thing he asks is basically, So where exactly is this package stored?

So I'll say my package is stored currently you can see I have a server name here, so let me go and

click on this dropdown here.

So I have clicked on the dropdown, so my package is stored in this win hyphen, EUC, whatever it is.

Right.

And which is that package.

The package which we have to run as a job is you can see that I've clicked on this dot dot here and

from here I will go and browse to that folder and inside that folder I will go and specify my exe file.

Right?

So you can see in the steps I've specified what has to be run as a job.

I'll say, okay, so that is done.

The next thing, the last thing.

What you have to define is the schedule.

So you want to run every week.

You want to run every day how it is, right?

So I'll say schedule one, two, three, whatever.

So the name is given.

Now the next thing is we need to go and specify the frequency by which this job will run.

Right?

So if you see my clock today, it is actually Monday, right?

So I'll say, okay, this has to run.

It is actually Monday and it is 257.

So what we'll do is let's go and run this on every Monday at 259.

Okay.

So I'm going to say Monday here occurs every minute, every one minute, right at 2:59 p.m..

Okay.

So we have to do this very quickly so that we sync up with the clock.

Right.

And I'll say I don't need Sunday.

Only on Monday.

Okay.

So.

Okay.

Right.

So there it is.

You can see my clock is still on 257 here.

So what I'll do is let us go and see the view history.

So in this view history, you can see that if this job has ran or not.

So I'm going to do a refresh here.

Right.

And let us go and start my clock here as well.

So you can see now it is still 258 here.

You can see my clock here.

It is still 258 here, Right?

So after one minute, right.

What should happen is I should get an entry here.

Right.

Or or the file should get copied.

So must be.

I can.

That's my location three.

And this is my clock here, which is running.

So let us see.

So as soon as 259 happens.

So we have more 30s to go.

Breathe in.

Breathe out.

You can till then.

So this is completely live here.

Right.

So more 30s to go.

So as soon as 259 happens and it is Monday, that job will run and it will go and copy that file into

this location.

Three folder while location three.

Remember, you know, we had configured that this packet should run on location three.

We had changed the configuration of the connection manager.

Right?

So 48, 49, 50 and more.

10s I know it is a bit boring, so but even I'm interested to watch if this happens or not.

So that it is.

That is my file.

Must be I should do a refresh here.

So I'm going to do a refresh.

Well, there it is.

You can see.

Good.

And also, if you go here.

Here.

You'll get a report saying that, yes, this file copy job actually ran at two £0.59.

Great.

Right.

So in case you want to go and run this package as a job, as a background job, you can go and create

a job into this SQL Server agent here and keep it running continuously.

Right?

So at this moment, I'll just go and delete this thing from here because it's going to probably go and

hang my computer, Right?

So.

Right.

Great.

Now, also, one of the things, you know, you would like to do with your setup is or with your deployment

is parameterize.

It means, for example, you know, when we did the configuration, right, when we actually click on

the configuration, you can see that there was something here called as parameters.

So what you can also do is you can go and create parameters by using this project dot params.

So you can go and create parameters and pass value to those parameters.

And those parameters will internally go and set values to variables or set values to connections, right?

For example, I can go to this project dot params here and I can let's say I will go and create a parameter

here.

So I'm going to go and add a parameter and this parameter would be a string.

So what I'll do is I'll add a parameter which is a string.

I'll name this parameter as destination destination path.

Right?

And I'll save this now.

So by default, what I'll do is.

I will.

Configure these parameters as it has to go to see location three.

Right.

And test.txt.

Okay, so by default, this is the value of this parameter, right?

And this parameter value.

I will set it to my connection here.

So I'll go to my exe file here.

So let us say I can go to my destination here.

So to my destination I will say that go and configure this connection string.

You can see that there we have a connection string here.

So let us go and configure this connection string by using that parameter so I can go and say here that

this connection string value will come from that parameter.

Very important step.

Again, I'll show.

I have clicked on the connection string.

I have clicked on this expression and I'm dragging and dropping this parameters over here.

Great.

So you can see I've created a parameter and I have attached that parameter to my destination path.

Okay.

Also, one more important point here.

You can keep you can create parameter at the project level.

So if you remember this parameter I have created by double clicking on this project dot params or you

can create parameter at the package level that means at your file level.

So you can have parameters, you know on different levels, right?

So I'll go and rebuild this solution again because I made some changes here.

So in case you want to go and deploy those changes, you have to go and redeploy.

Your project means rerun your setup.

So let us go to file copy program.

Let's go to development.

Let's run this spec file setup here again.

And let us do the next next.

So all these are this is common.

You know, this we already talked about in the previous part of this video and I will deploy this thing.

Same on this thing here.

So he will he will give me a give me a warning here saying that a project with this name already exist.

Do you want to continue?

Absolutely.

I want to just go and overwrite it.

I'll say deploy.

So now let us go back to our project here.

Now, remember the parameter.

What we created is at a solution level or at a project level, right?

So in other words, you know, that parameter will apply to all the X files.

Okay.

So if you see here.

So for example, if you go and click on this single X file here and if you say configure, you won't

find that parameters here.

Why?

Because it is not at the package level.

Remember, when I say package level, that means x file level, right?

But it is at this level, at the project level.

So if you go and if you say configure here, you can see that inside this parameters tab, you know

this file is here.

So for example, now let's say I will say that don't know if you wish you can again go and configure

from here as well.

So you can go and configure here and it will get applied to all your packages, right?

So in case you want to go and create parameters, you know, so you have to go to the project dot params

the file and go and add a parameter.

You can create parameter at two levels, as I have said.

One is at this project level, right?

Or you can create it at a package level.

So again, this is one more important thing, you know, which will help you to ease your deployment

as well as it will help you to ease your project.

Execution on production environment.

But now let us say that you want to go and parameterize all your projects which are inside this package.

So if you see here currently I have the file copy program and let's say I have one more project inside

this, let's say file copy program to write.

What if I want to go and parameterize, you know, some kind of a global thing to all my project inside

a package, Right?

So that means what if I want to go and set one global variable or a global parameter which is which

can be applied to this full package here.

So for that we can go and create something called as the environmental variables.

So what you can do is you can go and create an environment here.

So let us say this environment.

I'll say it belongs to my package, okay?

And inside this environment I can go and create a variable.

So for example, I can go to properties here and I can create a variable, let's say, let's say destination

path.

Okay, so destination, this is a destination variable which is of type string and the value is C location

three slash let's say global dot txt file.

Okay.

And I'll say okay, so now you can go and configure literally anything what you want.

For example, I can go here and I can say that.

Attach these parameters to my environmental variable here.

You can see here, I'll say use my environmental variable that is destination, which I've recently

created, and apply it to this parameters which I had created.

Remember the destination path here.

And in case you want, you can always go and even configure at the how do you say at the, at the package

level as well or I will say at the connection manager as well, you can go and set your environmental

variables.

So at this moment what I've done is I have set the environmental variable destination to my file copy

program, which is a parameter, right?

So I'll say, okay, here.

So what will happen is if you remember this environmental variable is actually pointing towards C location

three global dot txt file here.

So if I run this, if I run this package now in my location folder, the file name.

Will be global dot txt.

If I go to my location three here you can see now it has created global dot txt here.

Right?

So in other words it has taken the value from the environmental variables.

So in case you want to go and set parameters at the at the project level, right, you can use your

environment, your parameters and in case you want to go and apply at a global level, at a, at multiple

project level, if you want to go and set value, you can create environmental variables.

So very quickly, again, just revising what I said in case you want to go and set.

Parameters at the project level.

When I say project level means to all your X file.

Then you should use this.

Project dot params which had recently shown this one right.

And in case you want to, if you have multiple project, for example, you have file copy program,

then you have file copy program too, right?

If you want to go and set global variables at the project level, then you should use environments.

Now if you remember I said that you can do project deployment or you can do package deployment.

Now let's say that you have deployed the project and let's say that later on, you know, some changes

were made only in one of the files.

So, for example, at this moment we have only one file here, but let's say we had some two, three

file and some developer has made changes only to one exe file.

So what if you want to take that individual file and deploy it?

So what you can do is in that case you can always go to your SQL Server here and you can click on this

project node here and say import packages.

So if you say import packages here.

You will see that.

You can go here.

You can say that.

Okay, where is this package at this moment?

You can go and browse that folder and import that package.

Okay.

So I'll not show this further steps here, but this is very easy.

You can see there's a wizard here and you can go and import single files if you wish.

Right?

So this is one more important thing in case, you know, you load your 100 or 200 file in one go by

use by using project deployment.

And later on, when you want to go and send individual changes, then you can go and take individual

files and and deploy them into your production environment.

Now, when I initially started this video, I said that, you know, there are two ways of deploying.

One is you can deploy at the project level.

That means you can do project deployment or you can do package deployment.

And by default in SSIs 2012, it is project deployment.

So from RS 2012 onwards, it is only project deployment.

But in case, in case you are ever interested to go back to package deployment for for various reasons,

you can see here there is a menu here saying convert to package deployment so you can always go and

convert back to package deployment and then deploy at package level if you wish.

And also, you know, as I've said previously, you can always go and deploy not only in SQL Server,

you can go and deploy on integration services also.

So in case you want to go and deploy inside integration services, then you have two options here.

One is you can deploy it as a file system and other is you can deploy inside this msdb here.

So for example, if I want to go and deploy inside Msdb, I can right click, I can say import package,

I can go wherever that package is.

For example, let us try to give this package only.

So remember when we say package deployment, it is at the package level.

So you have to give the folder where your dot x file is.

So I'm going to say here, yes, my package path is this right?

And I'll say, okay.

Uh, there is some error here.

Okay, Let me say browse.

There it is to select the package and say, okay, so let me do a refresh and you can see that there

is this file copy package here.

I can go and execute this package, right?

So I can right click and I can run this package if I wish.

In the same way, if I want, I can go and deploy inside the file system also.

So here also I have to say import package, give the package name and it will get deployed here.

So I will not be going inside file deployment more.

In other words, package deployment at this moment because project deployment is a new way to go.

And I and I think that it is more better way to go.

Right.

So quickly, let's revise what we have learned in this video.

So first thing, the structure of SSIs project is solution project and then packages and params.

You can do deployment in two ways.

One is you can do deployment at the project level or you can do deployment at the package level.

You can deploy in three locations.

One is you can deploy inside SQL Server, which is your Sis DB, or you can go and deploy inside your

Sis integration services that is msdb or file system.

You can configure your project by using the project dot params or if you want, you can give global

configuration by using the environmental variables and if you ever want to go and run it as a job,

you can use the SQL Server agent.

So I hope that you enjoyed this video.

In this video we were trying to understand how to do SSIs deployment.

Thank you so much.